aws

User Guide

AWS Security Hub

AWS Security Hub : User Guide

Table of Contents

What are Security Hub and Security Hub CSPM?

> **Note**
>
> Security Hub is in preview release and is subject to change.

AWS Security Hub and AWS Security Hub CSPM are AWS services that protect your cloud environment. The services complement each other. When used together, they provide valuable insight into the security posture of your AWS environment.

Security Hub CSPM provides a comprehensive view of your security posture and helps you evaluate your cloud environment against security industry standards and best practices. Security Hub provides a unified experience that helps you prioritize and respond to critical security issues. Security Hub CSPM findings are routed to Security Hub automatically, where they're correlated with findings from other security services, such as Amazon Inspector, to generate exposures. This helps you identify the most critical risks in your environment. Security Hub also provides automated workflow capabilities, which help you incorporate Security Hub CSPM findings into your operational workflows.

As a best practice, we recommend enabling both services. You can enable Security Hub CSPM without enabling Security Hub if your primary focus is identifying misconfigurations and evaluating your security posture. However, if you enable Security Hub without enabling Security Hub CSPM, Security Hub cannot use Security Hub CSPM findings to provide information about risks and exposures in your AWS environment. For the optimal experience, we recommend not only enabling Security Hub and Security Hub CSPM, but also enabling these other security services: Amazon GuardDuty, Amazon Inspector, and Amazon Macie.

Introduction to AWS Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

AWS Security Hub is a unified cloud security solution that prioritizes your critical security issues and helps you respond at scale. Security Hub detects security issues by automatically correlating and enriching security signals from multiple sources, such as posture management, vulnerability management (Amazon Inspector), sensitive data (Macie), and threat detection (GuardDuty). This enables security teams to prioritize active risks in their cloud environments through automated analyses and contextual insights. Through intuitive visualizations, Security Hub transforms complex security signals into actionable insights, which enables you to make informed decisions about your security quickly. Security Hub also includes automated response workflows to help you remediate risks, improve team productivity, and minimize operational disruptions.

Features

Unified security solution

Gain broader visibility across your cloud environment through centralized management in a unified cloud security solution.

Actionable security insights

Gain actionable security insights through advanced analytics to learn about security risks associated with your environment.

Reduced response times

Streamline response times with automated workflows and an integrated ticketing system.

Exposure findings

Security Hub correlates findings from Security Hub CSPM control checks, Amazon Inspector, and other AWS services to detect exposures associated with AWS resources.

Findings are formatted in the Open Cybersecurity Schema Framework (OCSF)

Security Hub generates findings in OCSF and receives findings in OCSF from Security Hub CSPM and other AWS services:

- Amazon GuardDuty
- Amazon Macie
- Amazon Inspector

Dashboard

The Security Hub console provides a comprehensive view of your exposures, threats, security coverage, and resources as well as an interactive visualization called the attack path graph, which shows how potential attackers can access and take control of resources associated with an exposure finding.

Integrations with third-party products

You can enhance your security posture with Security Hub integrations. For example, if you use Jira Cloud or ServiceNow ITSM, you can use this feature to create tickets from findings.

Integrations

Security Hub integrates with the following AWS services.

- AWS Security Hub CSPM
- Amazon GuardDuty
- Amazon Inspector
- Amazon Macie

AWS Regions supported for public preview

Security Hub supports the following AWS Regions for this public preview release.

- Asia Pacific (Tokyo)
- Asia Pacific (Seoul)
- Asia Pacific (Osaka)

- Asia Pacific (Mumbai)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Canada (Central)
- Europe (Frankfurt)
- Europe (Stockholm)
- Europe (Ireland)
- US West (N. California)
- US West (Oregon)
- Europe (London)
- Europe (Paris)
- South America (São Paulo)
- US East (N. Virginia)
- US East (Ohio)

The following are opt-in AWS Regions, which require that you enable them before you can access them.

- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Jakarta)
- Europe (Milan)
- Middle East (Bahrain)

For information about these AWS Regions, see Opt-in status in the *AWS Regions and Availability Zones User Guide*.

Accessibility

Security Hub is available in the AWS Regions listed above. You can enable Security Hub for individual accounts or accounts in your organization. You can access Security Hub through the following:

Security Hub console

The Security Hub console is a browser-based interface you can use to create and manage AWS resources. In this console, you can access your account, data, and resources.

Security Hub API

The Security Hub API gives you programmatic access to your account, data, and resources. You can send HTTPS requests directly to Security Hub.

AWS CLI

With the AWS CLI, you can run commands in your system command line to perform tasks and build scripts that perform tasks. In some cases, the AWS CLI can be more useful than the Security Hub console.

AWS SDKs

AWS SDKs consist of libraries and sample code for various programming languages and platforms (C++, Go, Java, .NET , and Python). They provide programmatic access to Security Hub and other AWS services in your preferred language and can help you manage tasks such as managing errors, signing requests, and retrying requests.

Pricing

There is no cost to use Security Hub. Security Hub is free during this public preview.

Getting started with Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

The topics in this section describe how to get started with Security Hub.

Enabling Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

You can enable Security Hub for any AWS account. The procedures in this topic describe how to enable Security Hub from an AWS organization management account, a delegated administrator account, and a standalone account.

> **Note**
>
> After you enable Security Hub, exposures in your environment are analyzed immediately. However, you can wait up to 6 hours to receive an exposure finding for a resource.

Enable Security Hub for an organization

The procedure in this section describes how to enable Security Hub for the AWS organization management account. The procedure assumes you have set a delegated administrator for Security Hub CSPM and includes a step where you can set a delegated administrator for your organization in Security Hub. For more information about setting a delegated administrator in Security Hub, see Setting a delegated administrator account in Security Hub.

If you decide to set a delegated administrator for Security Hub during enablement, you will need to create a resource policy in the AWS Organizations console allowing the delegated administrator to perform actions on behalf of your organization. You can use the following sample resource policy for the delegated administrator account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::delegated-administrator-account-id:root"
      },
      "Action": [
        "organizations:AttachPolicy",
        "organizations:CreatePolicy",
        "organizations:DetachPolicy",
        "organizations:DeletePolicy",
        "organizations:UpdatePolicy",
        "organizations:ListPolicies",
        "organizations:ListPoliciesForTarget",
        "organizations:ListTargetsForPolicy",
```

```
        "organizations:DescribePolicy",
        "organizations:DescribeEffectivePolicy",
        "organizations:DisablePolicyType",
        "organizations:EnablePolicyType"
      ],
      "Resource": "*"
    }
  ]
}
```

If you do not set a delegated administrator, you can set a delegated administrator later. For more information, see Setting a delegated administrator account in Security Hub. The topic includes a procedure that describes how to set a delegated administrator for your organization from the **General** page in the Security Hub console.

The following procedure describes how to set a delegated administrator account for your organization in Security Hub.

To enable Security Hub for an AWS organization management account

1. Sign in to your AWS account with your AWS organization management account credentials. Open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the Security Hub homepage, select **Security Hub**. Choose **Get started**.
3. (Optional) For **Delegated administrator account**, set a delegated administrator based on the options provided. As a best practice, we recommend using the same delegated administrator across security services for consistent governance. For more information about setting a delegated administrator account, see Setting a delegated administrator account in Security Hub.
4. (Optional) For **Account enablement**, select the box to enable Security Hub for your AWS account.
5. Choose **Copy and attach** to open organization settings. In the Organizations console, select **Delegate** under **Delegated administrator for AWS Organizations**, and paste the resource policy. **Choose Create Policy**.
6. Go to the Security Hub console. Choose **Configure**.

When you enable Security Hub, a service-linked role called AWSServiceRoleForSecurityHubV2 is created in your account, and a service-linked recorder is added to your account. A service-linked recorder is a type of AWS Config recorder managed by an AWS service that can record

configuration data on service-specific resources. With a service linked recorder, Security Hub enables an event-driven approach for obtaining resource configuration items required for exposure analysis coverage. A service linked recorder is configured per AWS account and AWS Region.

> **Note**
>
> If you set a delegated administrator, the delegated administrator can create and apply a policy allowing it to enable and disable member accounts for Security Hub. For more information, see Creating a policy as the delegated administrator to manage member accounts.

Enable Security Hub for the delegated administrator

If the AWS organization management account sets a delegated administrator for their organization, the delegated administrator must enable Security Hub for their account. The following procedure must be completed by the delegated administrator, but only if the delegated administrator hasn't enabled Security Hub for their account. For information about setting a delegated administrator, see Setting a delegated administrator account in Security Hub.

To enable Security Hub for a delegated administrator account

1. Sign in to your AWS account with your delegated administrator credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the Security Hub homepage, select **Security Hub**, and choose **Get started**.
3. Choose **Enable**.
4. (Optional) For **Tags**, determine whether to add a key-value pair to the account setup.
5. Choose **Go to Security Hub**.

When you enable Security Hub, a service-linked role called AWSServiceRoleForSecurityHubV2 is created in your account, and a service-linked recorder is added to your account. A service-linked recorder is a type of AWS Config recorder managed by an AWS service that can record configuration data on service-specific resources. With a service linked recorder, Security Hub enables an event-driven approach for obtaining resource configuration items required for exposure analysis coverage. A service linked recorder is configured per AWS account and AWS Region.

> **Note**
>
> As the delegated administrator for an organization, you can create and apply a policy allowing you to enable and disable member accounts for Security Hub. For more information, see Creating a policy as the delegated administrator to manage member accounts.

Enable Security Hub for a standalone account

The following procedure describes how to enable Security Hub for a standalone account. There are two types of standalone accounts that can enable Security Hub: an AWS account not inside of an organization and an AWS account inside of an organization. An AWS account inside of an AWS organization can be an AWS account where a delegated administrator attaches AWS Organizations policy to the AWS account. For more information, see Security Hub policies in the *AWS Organizations User Guide*.

To enable Security Hub for a standalone account

1. Sign in to your AWS account with your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the Security Hub homepage, select **Security Hub**, and choose **Get started**.
3. Choose **Enable**.

When you enable Security Hub, a service-linked role called AWSServiceRoleForSecurityHubV2 is created in your account, and a service-linked recorder is added to your account. A service-linked recorder is a type of AWS Config recorder managed by an AWS service that can record configuration data on service-specific resources. With a service linked recorder, Security Hub enables an event-driven approach for obtaining resource configuration items required for exposure analysis coverage. A service linked recorder is configured per AWS account and AWS Region.

Setting a delegated administrator account in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

From the AWS organization management account, you can set a delegated administrator for your organization. As a best practice, we recommend using the same delegated administrator across security services for consistent governance. The procedures in this section describe how to set a delegated administrator for your organization in two ways. The first way is for an AWS organization management account that hasn't set a delegated administrator in Security Hub CSPM. The second way is for an AWS organization management account that enabled Security Hub but skipped setting a delegated administrator during enablement.

Considerations

You might encounter a scenario where you want to set a delegated administrator for Security Hub that's different from the delegated administrator for Security Hub CSPM. If you have a delegated administrator set up in Security Hub CSPM, consider the following:

- If the AWS organization management account is set as the delegated administrator for Security Hub CSPM, you cannot set this account as the delegated administrator for Security Hub. However, you can designate another AWS account in the organization as the delegated administrator for Security Hub. For consistent governance across security services, we recommend using the same account (other than the AWS organization management account) as the delegated administrator for Security Hub CSPM and Security Hub.
- If an account other than the AWS organization management account is set as the delegated administrator for Security Hub CSPM, this account becomes the delegated administrator in Security Hub automatically. In this scenario, Security Hub only allows this specific AWS account to serve as the delegated administrator.

Note

If you're using an account other than the organization management account as the Security Hub CSPM delegated administrator, removing it through either the Security Hub CSPM console or AWS Organizations API will also remove it from Security Hub. Similarly, if you remove the Security Hub delegated administrator through the Security Hub console or AWS Organizations API, it will be removed from Security Hub CSPM. When the delegated administrator is removed from Security Hub CSPM, Central Configuration will automatically opt out.

The following procedure assumes you have not set a delegated administrator for Security Hub CSPM and are setting a delegated administrator for Security Hub.

To set a delegated administrator in Security Hub

1. Sign in to your AWS account with your organization management account credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the Security Hub homepage, select Security Hub, and choose **Get started**.
3. In **Delegated administrator**, choose **Configure**. In the pop-up window, enter the 12-digit AWS account number for the AWS account that you want, or choose a suggested account (if you use delegated administrators in other security services) to set as the delegated administrator for your organization. Choose **Save**.
4. (Optional) For **Account enablement**, select the box to enable Security Hub for your AWS account.
5. Choose **Copy and attach** to open organization settings. In the Organizations console, select **Delegate** under **Delegated administrator for AWS Organizations**, and paste the resource policy. **Choose Create Policy**.
6. Go to the Security Hub console. Choose **Configure**.

> **Note**
>
> After you set the delegated administrator, that account must enable Security Hub and configure policies to receive findings from their member account.

The following procedure assumes you enabled Security Hub but skipped setting a delegated administrator during enablement. You can set a delegated administrator in the Security Hub console from the **General** page.

To set a delegated administrator in the Security Hub console from the General page

1. Sign in to your AWS account with your organization management account credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, choose **General**.
3. In **Delegated administrator**, choose **Configure**. In the pop-up window, enter the 12-digit AWS account number for the AWS account that you want to set as the delegated administrator for

your organization. Or choose a suggested AWS account if you set a delegated administrator in other AWS security services. Choose **Save**.

After you complete this procedure, you will need to copy the delegation policy statement for Security Hub and attach it to your delegated administrator for AWS Organizations policy, so the delegated administrator for Security Hub can perform actions in Security Hub. Without this policy statement, the delegated administrator cannot configure Security Hub for your organization. For more information, see Attaching the delegation policy statement for Security Hub.

Attaching the delegation policy statement for Security Hub

From the organization management account, you must copy the delegation policy statement for Security Hub and attach it to your delegated administrator for AWS Organizations policy, so the delegated administrator for Security Hub can perform actions in Security Hub. Without this policy statement, the delegated administrator cannot configure Security Hub for your organization. You can copy this policy from the **General** page in the Security Hub console. When you do this, you're directed to the **Settings** page in AWS Organizations console where you can edit your delegated administrator for AWS Organizations policy. This topic describes how to copy the policy in Security Hub. For information about how to update the delegated administrator for AWS Organizations policy, see Update a resource-based delegation policy with AWS Organizations in the *AWS Organizations User Guide*.

To attach the delegation policy statement for Security Hub

1. Sign in to your AWS account with your organization management account credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, choose **General**.
3. In **Delegation policy statement for Security Hub**, choose **Copy and attach**. You're directed to the **Settings** page in AWS Organizations where you can edit your delegated administrator for AWS Organizations policy to include the delegation policy statement. If you want to view the policy statement before you copy it, choose **Policy details**.

> **Note**
>
> If you set a delegated administrator, the delegated administrator can create and apply a policy that allows it to enable and disable memeber accounts. The procedure in the following topic describes how to set this policy.

Creating a policy as the delegated administrator to manage member accounts

As the delegated administrator for an organization, you can create and apply a policy that allows you to enable and disable member accounts. You can access all of your configured policies from the **Configurations** screen of the Security Hub console. The following procedure describes how to create this policy.

> **Note**
>
> Step 6. is an optional step where you can describe how this policy interacts with parent policies. For information about policy inheritance, see Understanding managment policy inheritance in the *AWS Organizations User Guide*.

To create a policy that allows you to enable and disable member accounts

1. Sign in using your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, choose **Settings**, and then choose **Configurations**.
3. Choose **Create policy**.
4. For **Details**, enter a name for the policy and determine whether to enter an optional description for the policy.
5. For **Regions**, choose **Enable all Regions**, **Disable all Regions**, or **Specify Regions**. If you choose **Enable all Regions**, you can determine whether to automatically enable new Regions. If you choose **Disable all Regions**, you can determine whether to automatically disable new Regions. If you choose, **Specify Regions**, you must choose which Regions you want to enable and disable.
6. (Optional) For **Advanced settings**, please refer to the guidance from AWS Organizations.
7. (Optional) For **Tags**, determine whether to add a key-value pair to the policy. You can add up to 50 tags.

8. Choose **Next**.
9. Review your changes, and then choose **Apply**. Your target accounts are configured based on the policy. To view the effective policy at the account level, you can review the **Organization** tab on the **Configurations** page where you can choose an account.

Removing the delegated administrator account in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

You can remove the delegated administrator account in the Security Hub console at any time. However, this action not only removes the delegated administrator from Security Hub, but also Security Hub CSPM. We recommend only performing this action when you have confirmed this operation with your security account.

> **Note**
>
> If you're using an account other than the organization management account as the Security Hub CSPM delegated administrator, removing it through either the CSPM Console or AWS Organizations API will also remove it from Security Hub.
> Similarly, if you remove the Security Hub delegated administrator through either the Security Hub Console or AWS Organizations API, it will also be removed from Security Hub CSPM. When the delegated administrator is removed from CSPM, Central Configuration will automatically opt out.

To remove the delegated administrator account

1. Sign in to your AWS account with your organization management account credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, choose **General**.
3. In **Delegated adminsitrator**, choose **Remove delegated administrator**. In the pop-up window, enter *confirm*, and choose **Remove**.

Security Hub recommendations

> **Note**
>
> Security Hub is in preview release and is subject to change.

The following security services in AWS send findings to Security Hub in the OCSF format. After you enable Security Hub, we recommend enabling these AWS services for additional security.

Security Hub CSPM

When you enable Security Hub CSPM, you get a comprehensive view of your security state in AWS. This helps you assess your environment against security industry standards and best practices. Although you can get started with Security Hub without enabling Security Hub CSPM, we recommend enabling Security Hub CSPM because Security Hub correlates security signals from Security Hub CSPM to improve your posture management.

If you enable Security Hub CSPM, we also recommend enabling the AWS Foundational Security Best Practices standard for your account. This standard consists of a set of controls that detect when your AWS accounts and resources deviate from security best practices. When you enable the AWS Foundational Security Best Practices standard for your account, AWS Security Hub CSPM automatically enables all of its controls, including controls for the following resource types:

- Account controls
- DynamoDB controls
- Amazon EC2 controls
- IAM controls
- AWS Lambda controls
- Amazon RDS controls
- Amazon S3 controls

You can disable any of the controls in this list. However, if you disable any of these controls, you cannot receive exposure findings for supported resources. For information about controls that apply to the AWS Foundational Security Best Practices standard, see AWS Foundational Security Best Practices v1.0.0 (FSBP) standard.

GuardDuty

When you enable GuardDuty, you can view all of your threats and security coverage findings in the dashboard of the Security Hub console. If you enable GuardDuty, GuardDuty automatically begins sending data to Security Hub in the OCSF format.

Amazon Inspector

When you enable Amazon Inspector, you can view all of your exposures and security coverage findings in the dashboard of the Security Hub console. If you enable Amazon Inspector, Amazon Inspector automatically begins sending data to Security Hub in the OCSF format.

We recommend activating Amazon EC2 scanning and Lambda standard scanning. When you activate Amazon EC2 scanning, Amazon Inspector scans Amazon EC2 instances in your account for package vulnerabilities and network reachability issues. When you activate Lambda standard scanning, Amazon Inspector scans Lambda functions for software vulnerabilities in package dependencies. For more information, see Activating a scan type in the *Amazon Inspector User Guide*.

Macie

When you enable Macie, you can detect additional exposures for your Amazon S3 buckets. We recommend configuring automated sensitive data discovery, so Macie can evaluate your Amazon S3 bucket inventory on a daily basis.

Security Hub concepts

> **Note**
>
> Security Hub is in preview release and is subject to change.

The following terms and concepts will help you understand how to manage exposure findings.

Exposure

A potential security scenario in your account that may be due to vulnerabilities, exploitable resources, or misconfigurations.

Exposure finding

A type of finding that describes an exposure present in your environment. An exposure finding includes traits and signals. A signal can include one or more types of exposure traits. Security Hub generates an exposure finding when signals from Security Hub CSPM control findings or other AWS services, such as Amazon Inspector, indicate the presence of an exposure. A resource can have at most one exposure finding. Security Hub generates an exposure finding when a resource is exposed. If a resource doesn't have any exposure traits or has insufficient traits, Security Hub doesn't generate an exposure finding for that resource.

Signal

A finding that contributes to an exposure finding. A signal can be referred to as a *contributing finding*. A signal can originate in Security Hub CSPM, AWS Config, or other AWS services, such as Amazon Inspector.

Trait

A security deviation that results in an exposure finding. Trait types include **Misconfiguration**, **Reachability**, **Sensitive Data**, and **Vulnerability**. A trait is associated with one signal, and a signal can contain multiple traits. For example, a Security Hub CSPM control indicates a customer managed policy allows administrative access control. This signal contains a misconfiguration trait.

OCSF findings in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

All findings in Security Hub are formatted in the Open Cybersecurity Schema Framework (OCSF). Security Hub considers findings with `activity_name != Close` as active findings. Active findings are automatically deleted if they aren't updated in 90 days. Security Hub considers findings with `Activity_name = Close` as closed findings. Closed findings are automatically deleted if they aren't updated in 14 days. Security Hub determines when a finding is updated using the most recent value of the finding `modified_time_dt`. At the end of a finding's retention period, Security Hub permanently deletes the finding. Finding providers can change the value of

the `finding.info.modified_time_dt` field when they update a finding. For information about other `Activity_name` values, see Vulnerability Finding in the OCSF schema.

Coverage findings in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

Coverage findings for Security Hub provide visibility into which AWS security features are enabled and where there might be gaps in coverage in a standalone account or across an organization's AWS environment. Enabling additional security features will enhance the detection capabilities of Security Hub. Coverage Findings evaluate what GuardDuty, Amazon Inspector, Macie, and Security Hub CSPM features are enabled for an account. These findings appear as a widget on the Security Hub dashboard with the ability to drill down into more detailed views by specific security capability. For the delegated administrator, this widget shows coverage breakdown across all Security Hub enabled accounts.

Limitations

- For member accounts, coverage information is aggregated across linked AWS Regions, but only for that member account.
- Coverage information is not shown for accounts not onboarded to Security Hub
- Coverage only indicates if an AWS service is enabled, not whether specific features in an AWS service are enabled.

Coverage findings for Security Hub CSPM

Security Hub CSPM Coverage Findings assess whether a qualified posture management security standard is enabled in an account. Enabling any Security Hub CSPM Standard will qualify, with the exception of AWS Control Tower and Resource Tagging standards.

It can take up to 24 hours to detect standards enabled by default when enabling Security Hub CSPM.

Coverage findings for GuardDuty

GuardDuty coverage findings assess whether GuardDuty is enabled and which GuardDuty features are enabled in an AWS account:

- Malware Protection for Amazon EC2 – Scans Amazon EC2 instances for potential malware
- Amazon EKS Protection – Monitors Kubernetes audit logs for threats in Amazon EKS clusters
- Lambda Protection – Analyzes Lambda function invocations for potential threats
- Amazon S3 Protection – Analyzes data events for potential threats to Amazon S3 buckets
- Amazon RDS Protection – Monitors for threats to Amazon RDS databases
- Runtime Monitoring – Provides real-time monitoring of runtime behavior in Amazon EC2 instances

It can take up to 24 hours for updates to GuardDuty coverage to reflect across all member accounts in an organization.

Coverage findings for Amazon Inspector

Amazon Inspector Coverage Findings assess whether Amazon Inspector is enabled and which features are enabled in an account:

- Amazon EC2 Scanning – Scans Amazon EC2 instances for vulnerabilities
- Amazon ECR Scanning – Scans container images in Amazon ECR for vulnerabilities
- Lambda Standard Scanning – Scans Lambda functions for vulnerabilities
- Lambda Code Scanning – Scans Lambda code functions for code vulnerabilities
- Amazon Inspector Code Security – Scans first-party application source code, third-party application dependencies, and Infrastructure as Code for vulnerabilities

Coverage findings for Macie

Macie Coverage Findings are assessments that indicate whether Macie is enabled across AWS accounts.

It can take up to 24 hours for updates to Macie automated sensitive data discovery to reflect across all member accounts in an organization.

Exposure findings in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

Security Hub correlates findings from Security Hub CSPM control checks and other AWS services, such as Amazon Inspector, to detect exposures associated with AWS resources. An exposure finding is a type of finding that describes a potential exposure in your environment due to vulnerabilities, exploitable resources, or misconfigurations. By prioritizing and resolving your most critical exposures, you can prevent potential attacks against your environment. You can access your exposure findings in the Security Hub console and programmatically with Security Hub CSPM and Security Hub API operations.

Supported resource types for exposure findings in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

Security Hub generates exposure findings for the following AWS resource types:

- `AWS::DynamoDB::Table`
- `AWS::EC2::Instance`
- `AWS::ECS::Service`
- `AWS::EKS::Cluster`
- `AWS::IAM::User`
- `AWS::Lambda::Function`
- `AWS::RDS::DBInstance`
- `AWS::S3::Bucket`

Security Hub generates one exposure finding per resource. If a resource doesn't have any exposure traits or has insufficient traits, Security Hub doesn't generate an exposure finding for that resource.

Supported trait types in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

Security Hub generates an exposure finding when AWS Security Hub CSPM control findings and findings generated by other supported AWS services, such as Amazon Inspector, contain exposure traits for a resource. The following table provides information about the supported trait types.

Trait type	Description	Source	Impacted resources
Misconfiguration	Indicates a misconfigured resource.	Security Hub CSPM control findings.	All resource types.
Reachability	Indicates open network paths to a resource.	Security Hub CSPM control findings and Amazon Inspector network reachability findings.	Amazon EC2 instances
Sensitive Data	Indicates that a resource contains sensitive data.	Macie sensitive data findings.	Amazon S3 buckets
Vulnerability	Indicates that a resource is exposed to Common Vulnerabilities and Exposure (CVEs).	Amazon Inspector package vulnerability findings.	Amazon EC2 instances , Amazon ECS services, Amazon EKS clusters,

Trait type	Description	Source	Impacted resources
			and Lambda functions

Each trait can be associated with multiple titles that provide details about the exposure affecting the resource. For example, you might see an **Exploit Available** title for the **Vulnerability** trait in the details for an EC2 exposure finding.

Generating exposure findings

> **Note**
>
> Security Hub is in preview release and is subject to change.

Security Hub generates exposure findings every 6 hours. During each 6-hour period, Security Hub considers the available exposure traits for a resource. It produces at most one exposure finding per resource ID. The uniqueness of a finding is determined by ID, AWS Region, type, and account. This means you can have two resources with the same ID, but the resources would be different resource types. This exposure finding aggregates all of the applicable exposure traits that apply to the resource.

If a resource doesn't have any exposure traits or has insufficient traits, Security Hub doesn't generate an exposure finding for that resource. Security Hub doesn't publish exposure findings for resource types that don't support exposure findings. When a resource has a significant number and combination of traits, Security Hub generates an exposure finding. The number and combination of traits also determine the severity level of the exposure finding.

Sample exposure finding

> **Note**
>
> Security Hub is in preview release and is subject to change.

Security Hub normalizes exposure findings in the Open Cybersecurity Schema Framework (OCSF).

Sample OCSF schema

In the following sample OCSF schema, the `related_events` parameter contains details unique to the exposure finding, such as contributing findings. Contributing findings are the traits and signals associated with an exposure finding. A single contributing finding can include one or more traits. The `observables` parameter identifies the resource associated with the contributing finding. This can be different from the `resources` parameter, which identifies the resource associated with the exposure finding.


```
{
    "activity_id": 1,
    "activity_name": "Create",
    "category_name": "Findings",
    "category_uid": 2,
    "class_name": "Detection Finding",
    "class_uid": 2004,
    "cloud": {
        "account": {
            "uid": "123456789012",
            "name": "production-application"
        },
        "cloud_partition": "aws",
        "provider": "AWS",
        "region": "us-east-1"
    },
    "finding_info": {
        "analytic": {
            "name": "Exposure",
            "type": "Rule",
            "type_id": 1,
            "uid": "0.0.1"
        },
        "created_time_dt": "2024-11-15T21:39:26.337224100Z",
        "desc": "Publicly invocable Lambda function executed outside of VPC has
 vulnerability with known exploit that can be exploited from remote network",
        "finding.info.modified_time_dt": "2024-11-15T21:39:26.337224100Z",
        "related_events_count": 3,
        "related_events": [
            {
                "tags": [
                    {
                        "name": "Vulnerability",
                        "values": [
```

```
                            "Attack Vector Network",
                            "EPSS Level >= High",
                            "EPSS Level >= Medium",
                            "Exploit Available",
                            "No Privileges Required",
                            "No User Interaction Required",
                            "Vulnerable"
                        ]
                    }
                ],
                "product": {
                    "uid": "arn:aws:securityhub:us-east-1::productv2/aws/inspector"
                },
                "observables": [
                    {
                        "type": "Resource UID",
                        "type_id": 10,
                        "value": "arn:aws:lambda:us-east-1:123456789012:application-
function"
                    }
                ],
                "type": "Finding",
                "title": "CVE-2023-33246 - org.apache.rocketmq:rocketmq-controller",
                "uid": "arn:aws:inspector2:us-
east-1:123456789012:finding/1234567890abcdef0"
            },
            {
                "tags": [
                    {
                        "name": "Reachability",
                        "values": [
                            "Publicly Invocable"
                        ]
                    }
                ],
                "product": {
                    "uid": "arn:aws:securityhub:us-east-1::productv2/aws/securityhub"
                },
                "observables": [
                    {
                        "type": "Resource UID",
                        "type_id": 10,
                        "value": "arn:aws:lambda:us-east-1:123456789012:application-
function"
```

```
                    }
                ],
                "type": "Finding",
                "title":  "Lambda function policies should prohibit public access",
                "uid": "arn:aws:securityhub:us-east-1:123456789012:security-control/
Lambda.1/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa"
            },
            {
                "tags": [
                    {
                        "name": "Misconfiguration",
                        "values": [
                            "Deployed outside VPC"
                        ]
                    }
                ],
                "product": {
                    "uid": "arn:aws:securityhub:us-east-1::productv2/aws/securityhub"
                },
                "observables": [
                    {
                        "type": "Resource UID",
                        "type_id": 10,
                        "value": "arn:aws:lambda:us-east-1:123456789012:application-
function"
                    }
                ],
                "type": "Finding",
                "title": "Lambda functions should be in a VPC",
                "uid": "arn:aws:securityhub:us-east-1:123456789012:security-control/
Lambda.3/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            }
        ],
        "title": "Publicly invocable Lambda function executed outside of VPC has
 vulnerability with known exploit that can be exploited from remote network",
        "types": [
            "Exposure/Potential Impact/Resource Hijacking"
        ],
        "uid": "arn:aws:securityhub:us-
east-1:123456789012:risk:1234f781c7ae7507f01e2fb460f15ca8fe7f9c95e257698a092cb74a4ea84a42"
    },
    "metadata": {
        "product": {
            "name": "Security Hub Exposure Analysis",
```

```
            "uid": "arn:aws:securityhub:us-east-1::productv2/aws/securityhub-risk",
            "vendor_name": "Amazon"
        },
        "processed_time_dt": "2024-11-15T21:39:58.819Z",
        "profiles": [
            "cloud",
            "datetime"
        ],
        "version": "1.4.0-dev"
    },
    "resources": [
        {
            "cloud_partition": "aws",
            "region": "us-east-1",
            "tags": [
                {
                    "name": "aws:cloudformation:stack-name",
                    "value": "VeepLambdaRule3"
                },
                {
                    "name": "aws:cloudformation:stack-id",
                    "value": "arn:aws:cloudformation:us-east-1:123456789012:stack/
VeepLambdaRule3/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
                },
                {
                    "name": "aws:cloudformation:logical-id",
                    "value": "lambdar3function94D10D40"
                }
            ],
            "type": "AwsLambdaFunction",
            "uid": "arn:aws:lambda:us-east-1:123456789012:application-function"
        }
    ],
    "severity": "Critical",
    "severity_id": 5,
    "status": "New",
    "status_id": 1,
    "time": 1731706766337,
    "time_dt": "2024-11-15T21:39:26.337224100Z",
    "type_name": "Detection Finding: Create",
    "type_uid": 200401,
    "vendor_attributes": {
        "severity_id": 5,
        "severity": "Critical"
```

```
    }
}
```

Determining the severity level of an exposure finding

> **Note**
>
> Security Hub is in preview release and is subject to change.

Security Hub assigns each exposure finding a default severity of `CRITICAL`, `HIGH`, `MEDIUM`, or `LOW`. Exposure findings with a severity of `INFORMATIONAL` aren't published. Security Hub uses several factors to determine the default severity level of an exposure finding:

- **Awareness** – The extent at which the exposure is not theoretical, but has publicly available or automated exploits. This applies to exposure findings for EC2 instances and Lambda functions.
- **Ease of discovery** – Whether automated tools, such as a port scan or internet search, are available to discover the resource at risk.
- **Ease of exploit** – The ease at which a threat actor can exploit the exposure. For example, if open network paths or misconfigured metadata exists, a threat actor can more easily exploit the exposure.
- **Likelihood of exploit** – The likelihood the exposure will be exploited in the next 30 days. This factor corresponds to the Exploit Protection Scoring System (EPSS) and applies to exposure findings for Amazon EC2 instances and Lambda functions.
- **Impact** – The harm if the exploit is carried out. For example, an exposure could lead to loss of accountability, loss of availability, loss of confidentiality from data exposure, or loss of integrity from data corruption.

Reviewing exposure findings

> **Note**
>
> Security Hub is in preview release and is subject to change.

You can review all of your exposure findings in the AWS Security Hub console and with the API. The **Exposures** page in the Security Hub console shows all active exposure findings. Exposure findings

are listed by decreasing severity. You can filter your exposure findings by adding and removing filters with the **Add filter** search bar. You can group your exposure findings with the **Group by** dropdown. You can also filter your exposure findings with the **Quick filters** menu.

Details for exposure findings

You can view many details for an exposure finding. In the Security Hub console, these details are divided among tabs. The **Overview** tab provides a snapshot of the exposure finding. The **Traits** tab lists the traits and signals associated with an exposure finding. The **Resources** tab provides details about the resource and resource tags associated with an exposure finding. The following list provides descriptions for exposure finding details.

- **Finding title** – The title of the exposure finding.
- **Severity level** – The severity level of the exposure finding. Security Hub uses the number and combination of traits for a resource to determine the severity level of an exposure finding. The severity level can be `CRITICAL`, `HIGH`, `MEDIUM`, or `LOW`. Security Hub doesn't publish exposure findings with a severity of `INFORMATIONAL`. You can update the `Severity` through the Security Hub console or with the BatchUpdateFindingsV2 API operation.
- **Description** – The description of the exposure finding.
- **Type** – The name of the exposure finding type. For example, the name might resemble something like `Exposure/Potential Impact/Resource Hijacking`.
- **Account** – The ID of the AWS account where the exposure finding was generated.
- **Age** – Indicates how long the exposure finding has been active.
- **Created time** – A timestamp that indicates when the exposure finding was created.
- **Modified time** – A timestamp that indicates when the exposure finding was last updated.
- **Region** – The AWS Region where the exposure finding was generated.
- **Product name** – The name of the product that generated the exposure finding.
- **Company name** – The name of the company that generated the exposure finding.
- **Activity name** – The name of the activity.
- **Status** – The status of this exposure finding.
- **Finding ID** – A unique identifier associated with the exposure finding.
- **Potential attack path (console only)** – An interactive visualization shows how potential attackers can access and take control of resources associated with an exposure finding. For more information, Viewing exposures in Security Hub with the potential attack path graph.

- **Traits** – Identifies trait types and trait titles associated with the exposure finding. In the Security Hub console, you can view traits by trait type or signal. This helps you analyze contributing findings in the context of the related exposure.
- **Resources** – Identifies the resource associated with the exposure finding.

Reviewing details for exposure findings

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to review details about exposure findings in the Security Hub console and with the API.

Reviewing details for an exposure finding in the Security Hub console

To view details for an exposure finding in the Security Hub console

1. Sign in using your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, choose **Exposures**.
3. Choose an exposure finding that you want to view details.

Reviewing details for an exposure finding with the API

You can review exposure findings with the `GetFindingsV2` API or with the AWS CLI. You can filter the results with the `FindingProviderFieldsTypes` parameter and by providing a filter value of `Exposure/EC2` if you only want to return exposure findings for EC2 instances. You can filter by other fields to narrow down results.

Example command

The following is a AWS CLI example that retrieves the 10 most recently generated exposure findings in your account. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
aws securityhub get-findings-v2 \
```

```
                    --max-results '10' \
                    --filter '{"CompositeFilters": [{"StringFilters":
[{"FieldName":"finding_info.title","Filter":
{"Value":"GuardDuty","Comparison":"PREFIX"}} ]}]}'
```

Remediating exposure findings

> **Note**
>
> Security Hub is in preview release and is subject to change.

The topics in this section describe remediation risks for different AWS services.

The `Remediation` field of the OCSF format contains two fields: `remediation` and `references`.

```
"Remediation": {
     "Recommendation": {
      "remediation":{"desc":"String",
      "references":["string array"]}
     }
},
```

> **Note**
>
> The remediation guidance provided in the following sections might require additional consultation in other AWS resources.

Remediating exposures for DynamoDB tables

AWS Security Hub can generate exposure findings for DynamoDB tables.

On the Security Hub console, the DynamoDB table involved in an exposure finding and its identifying information are listed in the **Resources** section of the finding details. Programmatically, you can retrieve resource details with the GetFindingsV2 operation of the Security Hub API.

After identifying the resource involved in an exposure finding, you can delete the resource if you don't need it. Deleting a nonessential resource can reduce your exposure profile and AWS costs. If

the resource is essential, follow these recommended remediation steps to help mitigate the risk. The remediation topics are divided based on the type of trait.

A single exposure finding contains issues identified in multiple remediation topics. Conversely, you can address an exposure finding and bring down its severity level by addressing just one remediation topic. Your approach to risk remediation depends on your organizational requirements and workloads.

> **Note**
>
> The remediation guidance provided in this topic might require additional consultation in other AWS resources.

Contents

Misconfiguration traits in DynamoDB

The following describes the misconfiguration traits and remediation steps for DynamoDB tables.

The DynamoDB table has point-in-time recovery disabled

Enable DynamoDB point-in-time recovery

DynamoDB point-in-time recovery provides continuous automated backups for your DynamoDB table data. For information about how to restore a DynamoDB table to a point in time, see Restoring a DynamoDB table to a point in time in the *Amazon DynamoDB User Guide*.

The DynamoDB table is not covered by a backup plan

AWS Backup provides a centralized service to configure, manage, and automate backups across AWS services, including DynamoDB. Without a backup plan, your table lacks scheduled, automated backups with customizable retention periods, creating significant security risks. An attacker could maliciously corrupt or delete your table data. Without proper backups, you may have no recovery option beyond the Point-in-Time Recovery window (if enabled), potentially resulting in permanent

data loss. Following data protection best practices, we recommend covering your DynamoDB tables with a backup plan.

Create a backup plan

Before creating a backup plan, determine an appropriate backup frequency and retention periods for your data. For information about how to create a backup plan, see Assign resources to a backup plan in the *Amazon DynamoDB User Guide*.

The DynamoDB table has deletion protection disabled

Deletion protection prevents the accidental deletion of DynamoDB tables. When deletion protection is disabled, DynamoDB tables are vulnerable to unintended deletion through console actions, API calls, CLI commands, or automated processes. This can expose your AWS environment to data loss, as an unauthorized entity with access to your AWS environment could intentionally delete tables, resulting in service disruption and permanent data loss. Following data protection best practices, we recommend enabling data protection for DynamoDB tables.

Enable deletion protection

If you manage multiple tables, consider using AWS CloudFormation to update table properties in bulk. You can modify your AWS CloudFormation templates to include `DeletionProtectionEnabled` property and update your stacks. After completing remediation, verify deletion protection is enabled in the **Additional** info dropdown in the table **Settings** tab.

Remediating exposures for EC2 instances

AWS Security Hub can generate exposure findings for Amazon Elastic Compute Cloud (EC2) instances.

On the Security Hub console, the EC2 instance involved in an exposure finding and its identifying information are listed in the **Resources** section of the finding details. Programmatically, you can retrieve resource details with the GetFindingsV2 operation of the Security Hub API.

After identifying the resource involved in an exposure finding, you can delete the resource if you don't need it. Deleting a nonessential resource can reduce your exposure profile and AWS costs. If the resource is essential, follow these recommended remediation steps to help mitigate the risk. The remediation topics are divided based on the type of trait.

A single exposure finding contains issues identified in multiple remediation topics. Conversely, you can address an exposure finding and bring down its severity level by addressing just one

remediation topic. Your approach to risk remediation depends on your organizational requirements and workloads.

> **Note**
>
> The remediation guidance provided in this topic might require additional consultation in other AWS resources.

Contents

Misconfiguration traits for EC2 instances

Here are misconfiguration traits for EC2 instances and suggested remediation steps.

The EC2 instance allows access to IMDS using version 1

Instance metadata is data about your Amazon EC2 instance that applications can use to configure or manage the running instance. The instance metadata service (IMDS) is an on-instance component that code on the instance uses to securely access instance metadata. If IMDS is not properly secured, it can become a potential attack vector, as it provides access to temporary

credentials and other sensitive configuration data. IMDSv2 provides stronger protection against exploitation through session-oriented authentication, requiring a session token for metadata requests and limiting session duration. Following standard security principles, AWS recommends that you configure Amazon EC2 instances to use IMDSv2 and disable IMDSv1.

Test application compatibility

Before implementing IMDSv2, test your instance to ensure its compatibility with IMDSv2. Some applications or scripts may require IMDSv1 for core functionality and require additional configuration. For more information about tools and recommended paths for testing application compatibility, Transition to using Instance Metadata Service Version 2 in the *Amazon Elastic Compute Cloud User Guide*.

Update instance to use IMDSv2

Modify existing instances to use IMDSv2. For more information, see Modify instance metadata options for existing instances in the *Amazon Elastic Compute Cloud User Guide*.

Apply updates to instances in an Auto Scaling group

If your instance is part of an Auto Scaling group, update your launch template or launch configuration with a new configuration, and perform an instance refresh.

The IAM role associated with the Amazon EC2 instance has an administrative access policy

Administrative access policies provide Amazon EC2 instances with broad permissions to AWS services and resources. These policies typically include permissions not required for instance functionality. Providing an IAM identity with an administrative access policy on an Amazon EC2 instance (instead of the minimum set of permissions the role attached to your instance profile needs) can increase the scope of an attack if the Amazon EC2 instance is compromised. If an instance is compromised, attackers could utilize these excessive permissions to move laterally across your environment, access data, or manipulate resources. Following standard security principles, we recommend you grant least privileges, which means you only grant the permissions required to perform a task.

Review and identify administrative policies

In the IAM dashboard, find the role with the role name. Review the permissions policy attached to the IAM role. If the policy is an AWS managed policy, look for `AdministratorAccess` or `IAMFullAccess`. Otherwise, in the policy document, look for statements with `"Effect": "Allow", "Action": "*"`, and `"Resource": "*"`.

Implement least privilege access

Replace administrative policies with policies that grant only the specific permissions required for the instance to function. For more information about security best practices for IAM roles, see Apply least-privilege permissions in Security best practices in the *AWS Identity and Access Management User Guide*. To identify unnecessary permissions, you can use the IAM Access Analyzer to understand how to modify your policy based on access history. For more information, see Findings for external and unused access in the *AWS Identity and Access Management User Guide*. Alternatively, you can create a new IAM role to avoid impacting other applications using the existing role. In this scenario, create a new IAM role, then associate the new IAM role with the instance. For instructions on replacing an IAM role for an instance, see Attach an IAM role to an instance in the *Amazon Elastic Compute Cloud User Guide*.

Secure configuration considerations

If service-level administrative permissions are necessary for the instance, consider implementing these additional security controls to mitigate risk:

- **Secure configuration considerations**

  - **Multi-factor authentication (MFA)** – MFA adds an additional security layer by requiring an additional form of authentication. This helps prevent unauthorized access even if credentials are compromised. For more information, see Require multi-factor authentication (MFA) in the *AWS Identity and Access Management User Guide*.
  - **IAM conditions** – Setting up condition elements allow you to restrict when and how administrative permissions can be used based on factors like source IP or MFA age. For more information, see Use conditions in IAM policies to further restrict access in the *AWS Identity and Access Management User Guide*.
  - **Permissions boundaries** – Permission boundaries establish the maximum permissions a role can have, providing guardrails for roles with administrative access. For more information, see Use permissions boundaries to delegate permissions management within an account in the *AWS Identity and Access Management User Guide*.

Apply updates to instances in an auto scaling group

For Amazon EC2 instances in an AWS auto scaling group, update the launch template or launch configuration with the new instance profile, and perform an instance refresh. For information about updating a launch template, see Modify a launch template (manage launch template

versions) in the *Amazon Elastic Compute Cloud User Guide*. For more information, see Use an instance refresh to update instances in an Auto Scaling group. For more information about using IAM roles with Auto Scaling groups, see IAM role for applications that run on Amazon EC2 instances in the *Amazon EC2 Auto Scaling User Guide*.

The IAM role associated with the Amazon EC2 instance has a service admin policy

Service access policies provide Amazon EC2 instances with broad permissions to AWS services and resources. These policies typically include permissions that are not required for instance functionality. Providing an IAM identity with an administrative access policy on an Amazon EC2 instance instead of the minimum set of permissions the role attached to your instance profile needs can increase the scope of an attack if an instance is compromised. Following standard security principles, we recommend that you grant least privileges, which means that you grant only the permissions required to perform a task.

Review and identify administrative policies

In the IAM dashboard, find the role with the role name. Review the permissions policy attached to the IAM role. If the policy is an AWS managed policy, look for `AdministratorAccess` or `IAMFullAccess`. Otherwise, in the policy document, look for statements with `"Effect": "Allow", "Action": "*"`, and `"Resource": "*"`.

Implement least privilege access

Replace service admin policies with those that grant only the specific permissions required for the instance to function. For more information on security best practices for IAM roles, see Apply least-privilege permissions in Security best practices in the *AWS Identity and Access Management User Guide*. To identify unnecessary permissions, you can use the IAM Access Analyzer to understand how to modify your policy based on access history. For more information, see Findings for external and unused access in the *AWS Identity and Access Management User Guide*. Alternatively, you can create a new IAM role to avoid impacting other applications that are using the existing role. In this scenario, create a new IAM role, then associate the new IAM role with the instance. For information about replacing an IAM role for an instance, see Attach an IAM role to an instance in the *Amazon Elastic Compute Cloud User Guide*

Secure configuration considerations

If service-level administrative permissions are necessary for the instance, consider implementing these additional security controls to mitigate risk:

Secure configuration considerations

If service-level administrative permissions are necessary for the instance, consider implementing these additional security controls to mitigate risk:

- **Multi-factor authentication (MFA)** – MFA adds an additional security layer by requiring an additional form of authentication. This helps prevent unauthorized access even if credentials are compromised. For more information, see Require multi-factor authentication (MFA) in the *AWS Identity and Access Management User Guide*.
- **IAM conditions** – Setting up condition elements allows you to restrict when and how administrative permissions can be used based on factors like source IP or MFA age. For more information, see Use conditions in IAM policies to further restrict access in the *AWS Identity and Access Management User Guide*.
- **Permissions boundaries** – Permission boundaries establish the maximum permissions a role can have, providing guardrails for roles with administrative access. For more information, see Use permissions boundaries to delegate permissions management within an account in the *AWS Identity and Access Management User Guide*.

Apply updates to instances in Auto Scaling group

For Amazon EC2 instances in an AWS auto scaling group, update the launch template or launch configuration with the new instance profile, and perform an instance refresh. For information about updating a launch template, see Modify a launch template (manage launch template versions) in the *Amazon Elastic Compute Cloud User Guide*. For more information, see Use an instance refresh to update instances in an Auto Scaling group. For more information about using IAM roles with Auto Scaling groups, see IAM role for applications that run on Amazon EC2 instances in the *Amazon EC2 Auto Scaling User Guide*.

The Amazon EC2 instance has a security group or network ACL that allows SSH or RDP access

Remote access protocols like SSH and RDP allow users to connect to and manage Amazon EC2 instances from external locations. When security groups permit unrestricted access to these protocols from the internet, they increase the attack surface of your Amazon EC2 instances by allowing internet access to your instance. Following standard security principles, AWS recommends you limit remote access to specific, trusted IP addresses or ranges.

1. **Modify security group rules**

   Restrict access to your Amazon EC2 instances to specific trusted IP addresses. Limit SSH and RDP access to specific trusted IP addresses, or use CIDR notation to specify IP ranges (e.g.,

198.168.1.0/24). To modify security group rules, see Configure security group rules in the *Amazon Elastic Compute Cloud User Guide*.

The Amazon EC2 instance has an open security group

Security groups act as virtual firewalls for your Amazon EC2 instances to control inbound and outbound traffic. Open security groups, which allow unrestricted access from any IP address, may expose your instances to unauthorized access. Following standard security principles, AWS recommends restricting security group access to specific IP addresses and ports.

Review security group rules and assess current configuration

Evaluate which ports are open and accessible from broad IP ranges, such as `(0.0.0.0/0 or ::/0)`. For instructions on viewing security group details, see DescribeSecurityGroups in the *Porting Assistant for .NET API Reference*.

Modify security group rules

Modify your security group rules to restrict access to specific trusted IP addresses or ranges. When updating your security group rules, consider separating access requirements for different network segments by creating rules for each required source IP range or restricting access to specific ports. To modify security group rules, see Configure security group rules in the *Amazon EC2 User Guide*.

The Amazon EC2 instance has a public IP address

Amazon EC2 instances with public IP addresses are publicly accessible from the internet. While public IP addresses are sometimes necessary for instances that provide services to external customers, this can be used as a potential by attack unauthorized principals. Following standard security principles, AWS recommends that you limit public exposure of resources when possible.

Move the instance to a private subnet

If the instance does not require direct internet access, consider moving it to a private subnet within your VPC. This will remove its public IP address while still allowing it to communicate with other resources within your VPC. For more information, see How do I move my Amazon EC2 instance to another subnet, Availability Zone, or VPC? in the AWS Knowledge Center.

Configure instances to launch without public IP addresses

If the instance was launched in a public subnet that doesn't require public IP addresses, the launch configuration can be modified to prevent the automatic assignment of public IP addresses. This can

be disabled at the subnet level or when launching individual instances. For more information, see Modify the IP addressing attributes of your subnet in the *Amazon Virtual Private Cloud User Guide* and Amazon Amazon EC2instance IP addressing in the *Amazon Elastic Compute Cloud User Guide*.

Alternative access methods

Consider the following options for alternative access methods:

- **Use a NAT Gateway for outbound internet connectivity** –

  For instances in private subnets that require access to the internet (e.g., to download updates), consider using a NAT Gateway instead of assigning a public IP address. A NAT Gateway allows instances in private subnets to initiate outbound connections to the internet while preventing inbound connections from the internet. For more information, see NAT gateways in the *Amazon Virtual Private Cloud User Guide*.
- **Use Elastic Load Balancing** – For instances that are running web applications, consider using an Elastic Load Balancer (LB). LBs can be configured to allow your instances to run in private subnets while the LB runs in a public subnet and handles internet traffic. For more information, see What is Elastic Load Balancing? in the AWSELB User Guide. See Load balancer subnets in *AWS Prescriptive Guidance* for guidance on how to choose a stickiness strategy for your LB.

Reachability traits for EC2 instances

Here are reachability traits for EC2 instances and suggested remediation steps.

The EC2 instance is reachable over the internet

Amazon EC2 instances with ports that are reachable from the internet through an internet gateway (including instances behind Application Load Balancers or Classic Load Balancers), a VPC peering connection, or a VPN virtual gateway may expose your instance to the internet. Following standard security principles, we recommend implementing least-privilege network access controls by restricting inbound traffic to only necessary sources and ports.

Modify or remove security group rules

In the **Resources** tab, open the resource for the Amazon EC2 Security Group. Review whether internet access is required for the instance to function. Modify or remove inbound security group rules that allow unrestricted access (`0.0.0.0/0` or `::/0`). Implement more restrictive rules based on specific IP ranges or security groups. If limited public access is necessary, restrict access to

specific ports and protocols required for the instance's function. For instructions on managing security group rules, see Configure security group rules in the *Amazon EC2 User Guide*.

Update network ACLs

Review and modify network access control lists (ACLs) associated with the instance's subnet. Verify that the ACL settings align with the security group changes and don't unintentionally allow public access. For instructions on modifying network ACLs, see Work with network ACLs in the *Amazon VPC User Guide*.

Alternative access methods

Consider the following options for alternative access methods:

- **Use NAT Gateway for outbound internet connectivity** – For instances in private subnets that require access to the internet (e.g., to download updates), consider using a NAT Gateway instead of assigning a public IP address. A NAT Gateway allows instances in private subnets to initiate outbound connections to the internet while preventing inbound connections from the internet. s
- **Use Systems Manager Session Manager** – Session Manager provides secure shell access to your Amazon EC2 instances without the need for inbound ports, managing SSH keys, or maintaining bastion hosts.
- **Use WAF and Elastic Load Balancing or Application Load Balancer** – For instances that are running web applications, consider using an LB combined with AWS Web Application Firewall (WAF). LBs can be configured to allow your instances to run in private subnets while the LB runs in a public subnet and handles internet traffic. Adding a WAF to your load balancer provides additional protection against web exploits and bots.

The Amazon EC2 instance is reachable within the Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) allows you to launch AWS resources in a defined virtual network. Amazon VPC network configurations that allow unrestricted access between instances can increase the scope of an attack if an instance is compromised. Following security best practices, AWS recommends implementing network segmentation and least-privilege access controls at the subnet and security group levels.

Review Amazon VPC network connectivity patterns

In the the exposure finding, identify the security group ID in the ARN. Identify which instances need to communicate with each other and on which ports. You can use Amazon VPC Flow Logs to analyze existing traffic patterns in your Amazon VPC to help identify which ports are being used.

Modify security group rules

Modify your security group rules to restrict access to specific trusted IP addresses or ranges. For example, instead of allowing all traffic from the entire VPC CIDR range (e.g., 10.0.0.0/16), restrict access to specific security groups or IP ranges. When updating your security group rules, consider separating access requirements for different network segments by creating rules for each required source IP range or restricting access to specific ports. To modify security group rules, see Configure security group rules in the Amazon EC2User Guide.

Consider organizing your Amazon VPC resources into subnets based on security requirements or function. For example, place web servers and database servers in separate subnets. For more information, see Subnets for your VPC in the *Amazon Virtual Private Cloud User Guide*.

Configure network ACLs for subnet level protection

Network Access Control Lists (NACLs) provide an additional layer of security at the subnet level. Unlike security groups, NACLs are stateless and require both inbound and outbound rules to be explicitly defined. For more information, see Control subnet traffic with network access control lists in the *Amazon Virtual Private Cloud User Guide*.

Additional considerations

Consider the following when restricting access to your Amazon VPC

- **Transit Gateway or Amazon VPC Peering with restrictive routing** – If your architecture uses multiple VPCs that need to communicate, consider using AWS Transit Gateway and Amazon VPC peering to provide connectivity between Amazon VPCs while allowing you to control which subnets can communicate with each other. For more information, see Get started with using Amazon VPC Transit Gateways and VPC peering connections.
- **Service endpoints and private links** – Amazon VPC endpoints can be used to keep traffic within the AWS network to communicate with AWS resources rather than over the internet. This reduces the need for direct connectivity between instances accessing the same services. For information on VPC endpoints, see What are Amazon VPC endpoints? in the *Amazon Virtual Private Cloud User Guide*. For connectivity to services hosted in other Amazon VPCs, consider using AWS PrivateLink.

Vulnerability traits for EC2 instances

Here are vulnerability traits for EC2 instances and suggested remediation steps.

EC2 instance has network-exploitable software vulnerabilities with a high likelihood of exploitation

Software packages that are installed on EC2 instances can be exposed to Common Vulnerabilities and Exposures (CVEs). Critical CVEs pose significant security risks to your AWS environment. Unauthorized principals can exploit these unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of data, or to access other systems. Critical vulnerabilities with high exploitation likelihood represent immediate security threats, as exploit code may already be publicly available and actively used by attackers or automated scanning tools. We recommend patching these vulnerabilities to protect your instance.

Update affected instances

Review the **References** section in the **Vulnerability** tab of the trait. Vendor documentation may include specific remediation guidance. Follow the appropriate remediation using these general guidelines:

Use Systems Manager Patch Manager to apply patches for both operating systems and applications. Patch Manager helps you select and deploy operating system and software patches automatically on large groups of instances. If you don't have Patch Manager configured, manually update the operating system on each affected instance.

Update the affected applications to their latest secure versions following the vendor's recommended procedures. To manage application updates across multiple instances, consider using Systems Manager State Manager to keep your software in a consistent state. If updates aren't available, consider removing or disabling the vulnerable application until a patch is released or other mitigations, such as restricting network access to the application or disabling vulnerable features.

Follow the specific remediation advice provided in the Amazon Inspector finding. This could involve changing security group rules, modifying instance configurations, or adjusting application settings.

Check if the instance is part of Auto Scaling Group. AMI-replacement patching is done on immutable infrastructures by updating the AMI ID that is configured to deploy new Amazon EC2 instances in an Auto Scaling group. If you are using a custom/golden AMI, create an instance with

the new AMI, and then customize the instance and create a new golden AMI For more information, see AMI updates patching (using patched AMIs for Auto Scaling groups).

Future considerations

To prevent future occurrences, consider implementing a vulnerability management program. Amazon Inspector can be configured to automatically scan for CVEs on your instances. Amazon Inspector can also be integrated with Security Hub for automatic remediations. Consider implementing a regular patching schedule using Systems Manager Maintenance Windows to minimize disruption to your instances.

The Amazon EC2 instance has software vulnerabilities

Software packages that are installed on Amazon EC2instances can be exposed to Common Vulnerabilities and Exposures (CVEs). Noncritical CVEs represent security weaknesses with lower severity or exploitability compared to critical CVEs. While these vulnerabilities pose less immediate risk, attackers can still exploit these unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of data, or to access other systems. Following security best practices, AWSrecommends patching these vulnerabilities to protect your instance from attack.

Update affected instances

Use AWS Systems Manager Patch Manager to apply patches for operating systems. Patch Manager helps you select and deploy operating system and software patches automatically on large groups of instances. If you don't have Patch Manager configured, manually update the operating system on each affected instance.

Update the affected applications to their latest secure versions following the vendor’s recommended procedures. To manage application updates across multiple instances, consider using AWS Systems Manager State Manager to keep your software in a consistent state. If updates aren't available, consider removing or disabling the vulnerable application until a patch is released or other mitigations, such as restricting network access to the application or disabling vulnerable features.

Follow the specific remediation advice provided in the Amazon Inspector finding. This could involve changing security group rules, modifying instance configurations, or adjusting application setting.

Check if the instance is part of Auto Scaling Group. AMI-replacement patching is done on immutable infrastructures by updating the AMI ID that is configured to deploy new Amazon EC2 instances in an Auto Scaling group. If you are using a custom/golden AMI, create an instance with

the new AMI, and then customize the instance and create a new golden AMI For more information, see AMI updates patching (using patched AMIs for Auto Scaling groups).

Future considerations

To prevent future occurrences, consider implementing a vulnerability management program. Amazon Inspector can be configured to automatically scan for CVEs on your instances. Amazon Inspector can also be integrated with Security Hub for automatic remediations. Consider implementing a regular patching schedule using Systems Manager Maintenance Windows to minimize disruption to your instances.

Remediating exposures for Amazon ECS services

AWS Security Hub can generate exposure findings for Amazon Elastic Container Service (Amazon ECS) services.

The Amazon ECS service involved in an exposure finding and its identifying information are listed in the **Resource** section of the finding details. You can retrieve these resource details on the Security Hub console or programmatically with the GetFindingsV2 operation of the Security Hub API.

After identifying the resource involved in an exposure finding, you can delete the resource if you don't need it. Deleting a nonessential resource can reduce your exposure profile and AWS costs. If the resource is essential, follow these recommended remediation steps to help mitigate the risk. The remediation topics are divided based on the type of trait.

A single exposure finding contains issues identified in multiple remediation topics. Conversely, you can address an exposure finding and bring down its severity level by addressing just one remediation topic. Your approach to risk remediation depends on your organizational requirements and workloads.

> **Note**
>
> The remediation guidance provided in this topic might require additional consultation in other AWS resources.

Contents

- The Amazon ECS service use a task definition configured with elevated privileges
- The Amazon ECS service has a container that can assume an IAM role
- The Amazon ECS service uses a task definition that allows containers to access the root file systems
- The Amazon ECS service uses a task definition configured to share a host's process namespace
- The Amazon ECS service uses a task definition configured with cleartext credentials in the environment variables
- The Amazon ECS service has an open security group
- The Amazon ECS service has a public IP addresses
- The Amazon ECS service uses a task definition that is configured with host networking mode enabled
- The IAM role associated with the Amazon ECS service has an administrative access policy
- Vulnerability traits for Amazon ECS services
  - The Amazon ECS service has a container with network-exploitable software vulnerabilities with a high likelihood of exploitation
  - The Amazon ECS service has a container with software vulnerabilities

Misconfiguration traits for Amazon ECS services

Here are misconfiguration traits for Amazon ECS services and suggested remediation steps.

The Amazon ECS service use a task definition configured with elevated privileges

Amazon ECS containers running with elevated privileges have similar capabilities to the host system, potentially allowing access to host resources and other containers. This configuration increases the risk that a compromised container could be used to access or modify resources outside its intended scope, potentially leading to container escape, unauthorized access to the underlying host, and breaches affecting other containers on the same host. Following standard security principles, AWS recommends that you grant least privileges, which means that you grant only the permissions required to perform a task.

Review and modify task definition

In the exposure, identify the task definition ARN. Open the task definition in the Amazon ECS console. In the task definition, look for the privileged flag set to true in the container definitions. If privileged mode is not required, create a new task definition revision without the privileged flag.

If privileged mode is required, consider configuring the container to use a read-only file system to prevent unauthorized modifications.

The Amazon ECS service has a container that can assume an IAM role

IAM Roles enable Amazon ECS tasks to securely access other AWS services using temporary credentials. Task execution roles may be required for Amazon ECS tasks where the container needs to interact with other AWS resources. While this is sometimes necessary for container functionality, improperly configured roles can grant excessive privileges that can be exploited by attackers if a container is compromised, potentially allowing unauthorized access to AWS resources, data theft, or unauthorized modification of your infrastructure. Following standard security principles, AWS recommends implementing least privilege access and reviewing IAM roles attached to your Amazon ECS tasks.

Review attached roles

Go to the IAM dashboard, and select the identified role. Review the permissions policy attached to the IAM role. If the task requires interaction with other AWS services, keep the task execution role and consider applying least-privilege permissions. Otherwise, create a new task definition revision without the execution role.

The Amazon ECS service uses a task definition that allows containers to access the root file systems

Amazon ECS containers with access to the host root filesystem can potentially read, modify, or execute critical files on the host system. This configuration increases the risk that a compromised container could be used to access or modify resources outside its intended scope, potentially exposing sensitive data on the host filesystem. Following standard security principles, AWS recommends that you grant least privileges, which means that you grant only the permissions required to perform a task.

Review and modify containers with host filesystem access

In the the exposure finding, identify the task definition ARN. Open the task definition in the Amazon ECS console. Look for the volumes section in the task definition that defines host path mappings. Review the task definition to determine if the host filesystem access is required for container functionality. s If host filesystem access is not required, create a new task definition revision and remove any volume definitions that use host paths. If host filesystem access is required, consider configuring the container to use a read-only file system to prevent unauthorized modifications.

The Amazon ECS service uses a task definition configured to share a host's process namespace

Amazon ECS containers running with exposed namespaces can potentially access host system resources and other container namespaces. This configuration could allow a compromised container to escape its isolation boundary, which could lead to accessing processes, network interfaces, or other resources outside of its intended scope. A process ID (PID) namespace provides separation between processes. It prevents system processes from being visible, and allows PIDs to be reused, including PID 1. If the host's PID namespace is shared with containers, it would allow containers to see all of the processes on the host system. This reduces the benefit of process level isolation between the host and the containers. These factors could lead to unauthorized access to processes on the host itself, including the ability to manipulate and terminate them. Following standard security principles, AWS recommends maintaining proper namespace isolation for containers.

Update task definitions with exposed namespaces

Open the **Resources** tab of the exposure, identify the task definition with the exposed namespace. Open the task definition in the Amazon ECS console. Look for the pidMode settings with a value of host, which would share the process ID namespaces with the host. Remove the pidMode: host settings from your task definitions to ensure containers run with proper namespace isolation.

The Amazon ECS service uses a task definition configured with cleartext credentials in the environment variables

Amazon ECS containers with cleartext credentials in environment variables expose sensitive authentication information that could be compromised if an attacker gains access to the task definition, container environment, or container logs. This creates a significant security risk, as leaked credentials could be used to access other AWS services or resources.

Replace cleartext credentials

In the exposure finding, identify the task definition with cleartext credentials. Open the task definition in the Amazon ECS console. Look for environment variables in the container definition that contain sensitive values such as AWS access keys, database passwords, or API tokens.

Consider the following alternatives to pass credentials:

- Instead of using AWS access keys, use IAM task execution roles and task roles to grant permissions to your containers.

- Store credentials as secrets in AWS Secrets Manager and reference them in your task definition.

Update task definitions

Create a new revision of your task definition that securely handles credentials. Then update your Amazon ECS service to use the new task definition revision.

The Amazon ECS service has an open security group

Security groups act as virtual firewalls for your Amazon ECS tasks to control inbound and outbound traffic. Open security groups, which allow unrestricted access from any IP address, may expose your containers to unauthorized access, increasing the risk of exposure to automated scanning tools and targeted attacks. Following standard security principles, AWS recommends restricting security group access to specific IP addresses and ports.

Review security group rules and assess current configuration

Open the resource for the Amazon ECS Security Group. Evaluate which ports are open and accessible from broad IP ranges, such as (`0.0.0.0/0 or ::/0`).

Modify security group rules

Modify your security group rules to restrict access to specific trusted IP addresses or ranges. When updating your security group rules, consider separating access requirements for different network segments by creating rules for each required source IP range or restricting access to specific ports.

Modify security group rules

Consider the following options for alternative access methods:

- Session Manager provides secure shell access to your Amazon EC2 instances without the need for inbound ports, managing SSH keys, or maintaining bastion hosts.
- NACLs provide an additional layer of security at the subnet level. Unlike security groups, NACLs are stateless and require both inbound and outbound rules to be explicitly defined.

The Amazon ECS service has a public IP addresses

Amazon ECS services with public IP addresses assigned to their tasks are directly accessible from the internet. While this may be necessary for services that need to be publicly available, it increases the attack surface and potential for unauthorized access.

Identify services with public IP addresses

In the exposure finding, identify the Amazon ECS service that has public IP addresses assigned to its tasks. Look for the `assignPublicIp` setting with a value of `ENABLED` in the service configuration.

Update task definitions

Create a new revision of your task definition that disables public IP addresses. Then update your Amazon ECS service to use the new task definition revision.

Implement private network access patterns

For instances that are running web applications, consider using a Load Balancer (LB). LBs can be configured to allow your instances to run in private subnets while the LB runs in a public subnet and handles internet traffic.

The Amazon ECS service uses a task definition that is configured with host networking mode enabled

Amazon ECS containers running with host networking mode share the network namespace with the host, allowing direct access to the host's network interfaces, ports, and routing tables. This configuration bypasses the network isolation provided by containers, potentially exposing services running on the container directly to external networks and allowing containers to modify host network settings. Following standard security principles, AWS recommends maintaining proper network isolation for containers.

Disable host networking mode

In the exposure finding, identify the task definition with host networking mode. Open the task definition in the Amazon ECS console. Look for the networkMode setting with a value of host in the task definition.

Consider the following options to disable host networking mode:

- The `awsvpc` network mode provides the strongest level of network isolation by giving each task its own elastic network interface.
- The `bridge` network mode provides isolation while allowing port mappings to expose specific container ports to the host.

Update task definitions

Create a new revision of your task definition with the updated network mode configuration. Then update your Amazon ECS service to use the new task definition revision.

The IAM role associated with the Amazon ECS service has an administrative access policy

IAM roles with administrative access policies attached to Amazon ECS tasks provide broad permissions that exceed what is typically required for container operation. This configuration increases the risk that a compromised container could be used to access or modify resources throughout your AWS environment. Following standard security principles, AWS recommends implementing least privilege access by granting only the permissions required for a task to function.

Review and identify administrative policies

n the **Resource ID**, identify the IAM role name. Go to the IAM dashboard and select the identified role. Review the permissions policy attached to the IAM role. If the policy is an AWS managed policy, look for `AdministratorAccess`. Otherwise, in the policy document, look for statements that have the statements `"Effect": "Allow", "Action": "*", and "Resource": "*"` together.

Implement least privilege access

Replace administrative policies with those that grant only the specific permissions required for the instance to function. To identify unnecessary permissions, you can use the IAM Access Analyzer to understand how to modify your policy based on access history. Alternatively, you can create a new IAM role to avoid impacting other applications that are using the existing role. In this scenario, create a new IAM role, then associate the new IAM role with the instance.

Secure configuration considerations

If service-level administrative permissions are necessary for the instance, consider implementing these additional security controls to mitigate risk:

- MFA adds an additional security layer by requiring an additional form of authentication. This helps prevent unauthorized access even if credentials are compromised.
- Setting up condition elements allow you to restrict when and how administrative permissions can be used based on factors like source IP or MFA age.

Update task definitions

Create a new revision of your task definition that references the new or updated IAM roles. Then update your Amazon ECS service to use the new task definition revision.

Vulnerability traits for Amazon ECS services

Here are reachability traits for Amazon ECS and suggested remediation steps.

The Amazon ECS service has a container with network-exploitable software vulnerabilities with a high likelihood of exploitation

1. **Understand the exposure**

   Package vulnerability findings identify software packages in your AWS environment that are exposed to Common Vulnerabilities and Exposures (CVEs). Attackers can exploit these unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of data, or to access other systems. ECR container images can have package vulnerability findings.
2. **Remediate the exposure**
   a. **Update package version**

      Review the package vulnerability finding for your Lambda function. Update the package version as suggested by Amazon Inspector. For information, see Viewing details for your Amazon Inspector findings in the *Amazon Inspector User Guide*. The **Remediation** section of the finding details in the Amazon Inspector console tells you which commands you can run to update the package.

   b. **Update base container images**

      Rebuilding and update the base container images regularly to keep your containers up to date. When rebuilding the image, don't include unnecessary components to reduce the attack surface. For instructions on rebuilding a container image, see Rebuild you images often.

The Amazon ECS service has a container with software vulnerabilities

Software packages that are installed on Amazon ECS containers can be exposed to Common Vulnerabilities and Exposures (CVEs). Low priority vulnerabilities represent security weaknesses with lower severity or exploitability compared to high priority vulnerabilities. While these vulnerabilities pose less immediate risk, attackers can still exploit these unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of data, or to access other systems.

Update affected container images

Review the **References** section in the **Vulnerability** tab of the trait. Vendor documentation may include specific remediation guidance.

Apply the appropriate remediation by following these general guidelines:

- Update your container images to use patched versions of the affected packages.
- Update the affected dependencies in your application to their latest secure versions.

After updating your container image, push it to your container registry and update your Amazon ECS task definition to use the new image.

Future considerations

To further strengthen the security posture of your container images, consider following Amazon ECS task and container security best practices. Amazon Inspector can be configured to automatically scan for CVEs on your containers. Amazon Inspector can also be integrated with Security Hub for automatic remediations. Consider implementing a regular patching schedule using Systems Manager Maintenance Windows to minimize disruption to your containers.

Remediating exposures for Amazon EKS clusters

AWS Security Hub can generate exposure findings for Amazon Elastic Kubernetes Service (Amazon EKS) clusters.

The Amazon EKS cluster involved in an exposure finding and its identifying information are listed in the **Resource** section of the finding details. You can retrieve these resource details on the Security Hub console or programmatically with the GetFindingsV2 operation of the Security Hub API.

After identifying the resource involved in an exposure finding, you can delete the resource if you don't need it. Deleting a nonessential resource can reduce your exposure profile and AWS costs. If the resource is essential, follow these recommended remediation steps to help mitigate the risk. The remediation topics are divided based on the type of trait.

A single exposure finding contains issues identified in multiple remediation topics. Conversely, you can address an exposure finding and bring down its severity level by addressing just one remediation topic. Your approach to risk remediation depends on your organizational requirements and workloads.

> **Note**
>
> The remediation guidance provided in this topic might require additional consultation in other AWS resources.

Contents

Misconfiguration traits for Amazon EKS clusters

Here are misconfiguration traits for Amazon EKS clusters and suggested remediation steps.

The Amazon EKS cluster allows public access

The Amazon EKS cluster endpoint is the endpoint that you use to communicate with your cluster's Kubernetes API server. By default, this endpoint is public to the internet. Public endpoints increase your attack surface area and the risk of unauthorized access to your Kubernetes API server, potentially allowing attackers to access or modify cluster resources or access sensitive data. Following security best practices, AWS recommends restricting access to your EKS cluster endpoint to only necessary IP ranges.

Modify endpoint access

In the exposure finding, open the resource. This will open the affected Amazon EKS cluster. You can configure your cluster to use private access, public access, or both. With private access, Kubernetes API requests that originate within your cluster's VPC use the private VPC endpoint. With public access, Kubernetes API requests that originate from outside your cluster's VPC use the public endpoint.

Modify or remove public access to the cluster

To modify endpoint access for an existing cluster, see Modifying cluster endpoint access in the *Amazon Elastic Kubernetes Service User Guide*. Implement more restrictive rules based on specific IP ranges or security groups. If limited public access is necessary, restrict access to specific CIDR block ranges or use prefix lists.

The Amazon EKS cluster uses an unsupported Kubernetes version

Amazon EKS supports each Kubernetes version for a limited period of time. Running clusters with unsupported Kubernetes versions can expose your environment to security vulnerabilities, as CVE patches will stop being released for outdated versions. Unsupported versions may contain known security vulnerabilities that can be exploited by attackers and lack security features that may be available in newer versions. Following security best practices, AWS recommends keeping your Kubernetes version updated.

Update Kubernetes version

In the exposure finding, open the resource. This will open the affected Amazon EKS cluster. Before updating your cluster, review Available versions on standard support in the *Amazon Elastic Kubernetes Service User Guide* for a list of currently supported Kubernetes versions.

The Amazon EKS cluster uses unencrypted Kubernetes secrets

Kubernetes secrets are, by default, stored unencrypted in the API server's underlying data store (etcd). Anyone with API access or with access to etcd can retrieve or modify a secret. To prevent this, you should encrypt Kubernetes secrets at rest. If Kubernetes Secrets are unencrypted, they are vulnerable to unauthorized access if etcd is compromised. Since secrets often contain sensitive information like passwords and API tokens, their exposure could lead to unauthorized access to other applications and data. Following security best practices, AWS recommends encrypting all sensitive information stored in Kubernetes secrets.

Encrypt Kubernetes secrets

Amazon EKS supports the encryption of Kubernetes secrets using KMS keys through envelope encryption. To enable encryption of Kubernetes secrets for your EKS cluster, see Encrypt Kubernetes secrets with KMS on existing clusters in the *Amazon EKS User Guide*.

Vulnerabilitiy traits for Amazon EKS clusters

Here are the vulnerability traits for Amazon EKS clusters.

The Amazon EKS cluster has a container with network-exploitable software vulnerabilities with a high likelihood of exploitation

Software packages that are installed on EKS clusters can be exposed to Common Vulnerabilities and Exposures (CVEs). Critical CVEs pose significant security risks to your AWS environment. Unauthorized users can exploit these unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of data, or to access other systems. Critical vulnerabilities with high exploitation likelihood represent immediate security threats, as exploit code may already be publicly available and actively used by attackers or automated scanning tools. Following security best practices, AWS recommends patching these vulnerabilities to protect your instance from attack.

Update affected instances

Update your container images to newer versions that include security fixes for the identified vulnerabilities. This typically involves rebuilding your container images with updated base images or dependencies, then deploying the new images to your Amazon EKS cluster.

The Amazon EKS cluster has a container with software vulnerabilities

Software packages that are installed on Amazon EKS clusters can be exposed to Common Vulnerabilities and Exposures (CVEs). Noncritical CVEs represent security weaknesses with lower severity or exploitability compared to critical CVEs. While these vulnerabilities pose less immediate risk, attackers can still exploit these unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of data, or to access other systems. Following security best practices, AWS recommends patching these vulnerabilities to protect your instance from attack.

Update affected instances

Update your container images to newer versions that include security fixes for the identified vulnerabilities. This typically involves rebuilding your container images with updated base images or dependencies, then deploying the new images to your Amazon EKS cluster.

Remediating exposures for IAM users

AWS Security Hub can generate exposure findings for AWS Identity and Access Management (IAM) users.

On the Security Hub console, the IAM user involved in an exposure finding and its identifying information are listed in the **Resources** section of the finding details. Programmatically, you can retrieve resource details with the GetFindingsV2 operation of the Security Hub API.

After identifying the resource involved in an exposure finding, you can delete the resource if you don't need it. Deleting a nonessential resource can reduce your exposure profile and AWS costs. If the resource is essential, follow these recommended remediation steps to help mitigate the risk. The remediation topics are divided based on the type of trait.

A single exposure finding contains issues identified in multiple remediation topics. Conversely, you can address an exposure finding and bring down its severity level by addressing just one remediation topic. Your approach to risk remediation depends on your organizational requirements and workloads.

> **Note**
>
> The remediation guidance provided in this topic might require additional consultation in other AWS resources.

IAM best practices recommend that you create IAM roles or use federation with an identity provider to access AWS using temporary credentials instead of creating individual IAM users. If that's an option for your organization and use case, we recommend switching to roles or federation instead of using IAM users. For more information, see IAM users in the *IAM User Guide*.

Contents

Misconfiguration traits for IAM users

Here are misconfiguration traits for IAM users and suggested remediation steps.

The IAM user has a policy with administrative access

IAM policies grant a set of privileges to IAM users when accessing resources. Administrative policies provide IAM users with broad permissions to AWS services and resources. Providing full administrative privileges, instead of the minimum set of permissions that the user needs, can increase the scope of an attack if credentials are compromised. Following standard security principles, AWS recommends that you grant least privileges, which means that you grant only the permissions required to perform a task.

1. **Review and identify administrative policies** – In the **Resource ID**, identify the IAM role name. Go to the IAM dashboard and select the identified role. Review the permissions policy attached to the IAM user. If the policy is an AWS managed policy, look for `AdministratorAccess` or `IAMFullAccess`. Otherwise, in the policy document, look for statements that have the statements `"Effect": "Allow"` with `"Action": "*"` over `"Resource": "*"`.
2. **Implement least privilege access** – Replace service administrative policies with those that grant only the specific permissions required for the user to function. For more information on security best practices for IAM policies, see Apply least-privilege permissions in the *AWS Identity and Access Management User Guide*. To identify unnecessary permissions, you can use the IAM Access Analyzer to understand how to modify your policy based on access history. For more information, see Findings for external and unused access in the *AWS Identity and Access Management User Guide*.
3. **Secure configuration considerations** – If service administrative permissions are necessary for the instance, consider implementing these additional security controls to mitigate risk:
   - **Multi-factor authentication (MFA)** – MFA adds an additional security layer by requiring an additional form of authentication. This helps prevent unauthorized access even if credentials are compromised. For more information, see Require multi-factor authentication (MFA) in the *AWS Identity and Access Management User Guide*.
   - **IAM conditions** – Setting up condition elements allow you to restrict when and how administrative permissions can be used based on factors like source IP or MFA age. For more information, see Use conditions in IAM policies to further restrict access in the *AWS Identity and Access Management User Guide*.
   - **Permission boundaries** – Permission boundaries establish the maximum permissions a role can have, providing guardrails for roles with administrative access. For more information, see Use permissions boundaries to delegate permissions management within an account in the *AWS Identity and Access Management User Guide*.

The IAM user does not have MFA enabled

Multi-factor authentication (MFA) adds an extra layer of protection on top of a user name and password. When MFA is enabled and an IAM user signs in to an AWS website, they are prompted for their user name, password, and an authentication code from their AWS MFA device. The authenticating principal must possess a device that emits a time-sensitive key and must have knowledge of a credential. Without MFA, if a user's password is compromised, an attacker gains full access to the user's AWS permissions. Following standard security principles, AWS recommends enabling MFA for all accounts and users that have AWS Management Console access.

Review MFA types

AWS supports the following MFA types:

- Passkeys and security keys
- Virtual authenticator applications
- Hardware TOTP tokens

Although authentication with a physical device typically provides more stringent security protection, using any type of MFA is more secure than having MFA disabled.

Enable MFA

To enable the MFA type that suits your requirements, see AWS multi-factor authentication in IAM in the *IAM User Guide*. Follow the steps for the specific MFA type you want to implement. For organizations managing many users, you may want to enforce MFA usage by requiring MFA to access sensitive resources.

The IAM user has a policy with administrative access to an AWS service

Service admin policies provide IAM users with permissions to perform all actions within a specific AWS service. These policies typically include permissions that are not required for users to perform their job functions. Providing an IAM user with service administrator privileges, instead of the minimum set of permissions needed, increases the scope of an attack if credentials are compromised. Following standard security principles, AWS recommends that you grant least privileges, which means that you grant only the permissions required to perform a task.

Review and identify service admin policies

In the **Resource ID**, identify the IAM role name. Go to the IAM dashboard and select the identified role. Review the permissions policy attached to the IAM user. If the policy is an AWS managed policy, look for `AdministratorAccess` or `IAMFullAccess`. Otherwise, in the policy document, look for statements that have the statements `"Effect": "Allow" with "Action": "*" over "Resource": "*"`.

Implement least privilege access

Replace service administrative policies with those that grant only the specific permissions required for the user to function. To identify unnecessary permissions, you can use the IAM Access Analyzer to understand how to modify your policy based on access history.

Secure configuration considerations

If service administrative permissions are necessary for the instance, consider implementing these additional security controls to mitigate exposure:

- MFA adds an additional security layer by requiring an additional form of authentication. This helps prevent unauthorized access even if credentials are compromised.
- Use condition elements to restrict when and how administrative permissions can be used based on factors like source IP or MFA age.
- Use permission boundaries to establish the maximum permissions a role can have, providing guardrails for roles with administrative access.

The AWS account for the IAM user has weak password policies

Password policies help protect against unauthorized access by enforcing minimum complexity requirements for IAM user passwords. Without strong password policies, there's an increased risk that user accounts could be compromised through password guessing or brute force attacks. Following standard security principles, AWS recommends implementing a strong password policy to ensure users create complex passwords that are difficult to guess.

Configure a strong password policy

Go to the IAM dashboard and navigate to Account settings. Review the current password policy settings for your account, including minimum length, character types required, and password expiration settings.

At a minimum, AWS recommends following these best practices when setting your password policy:

- Require at least one uppercase character.
- Require at least one lowercase character.
- Require at least one symbol.
- Require at least one number.
- Require at least eight characters.

Additional security considerations

Consider these additional security measures in addition to a strong password policy:

- MFA adds an additional security layer by requiring an additional form of authentication. This helps prevent unauthorized access even if credentials are compromised.
- Setting up condition elements to restrict when and how administrative permissions can be used based on factors like source IP or MFA age.

The IAM user has unused credentials

Unused credentials, including passwords and access keys that have remained inactive for 90 days or more pose a security risk to your AWS environment. These unused credentials create potential attack vectors for attackers and increase your organization's overall attack surface. Following security best practices, AWS recommends deactivating or removing credentials that haven't been used in 90 days or more to reduce your attack surface.

Deactivate or remove unused credentials

In the exposure finding, open the resource. This will open the user details window. Before taking action on unused credentials, assess the potential impact on your environment. Removing credentials without proper assessment could disrupt background processes, scheduled jobs, and more. Consider a brief deactivation period before permanent removal to verify the impact of removing the unused credentials.

Take the appropriate action based on the credential type:

- For unused console passwords, consider first changing the password and temporarily deactivating it. If no issues arise, proceed with permanent deactivation or deletion.
- For unused access keys, consider first deactivating the key. After confirming no systems are affected, proceed with permanent deactivation or deletion.

- For unused users, consider temporarily deactivating the user by attaching a restrictive policy before full deletion.

The IAM user has unrotated access keys

Access keys consist of an access key ID and a secret access key that enable programmatic access to AWS resources. When access keys remain unchanged for extended periods of time, they increase the risk of unauthorized access if they are compromised. Following security best practices, AWS recommends rotating access keys every 90 days to minimize the window of opportunity for attackers to use compromised credentials.

Rotate access keys

In the exposure finding, open the resource. This will open the user details window. To rotate access keys, see Manage access keys for IAM users in the *IAM User Guide*.

The IAM user has a policy that allows unrestricted access to KMS key decryption

AWS KMS enables you to create and manage cryptographic keys that are used to protect your data. IAM policies that allow unrestricted AWS KMS decryption permissions (e.g., `kms:Decrypt` or `kms:ReEncryptFrom`) on all KMS keys can lead to unauthorized data access if an IAM user's credentials are compromised. If an attacker gains access to these credentials, they could potentially decrypt any encrypted data in your environment, which could include sensitive data. Following security best practices, AWS recommends implementing least privilege by limiting AWS KMS decryption permissions to only specific keys that users need for their job functions.

Implement least-privilege access

In exposure finding, open the resource. This will open the IAM Policy window. Look for permissions in KMS that allow kms:Decrypt or `kms:ReEncryptFrom` or `KMS:*` with a resource specification of `"*"`. Update the policy to restrict AWS KMS decryption permissions to only the specific keys needed. Modify the policy to replace the `"*"` resource with the specific ARNs of required AWS KMS keys.

Secure configuration considerations

Consider adding conditions to further restrict when these permissions can be used. For example, you can limit decryption operations to specific VPC endpoints or source IP ranges. You can also configure key policies to further restrict who can use specific KMS keys.

Remediating exposures for Lambda functions

> **Note**
>
> Security Hub is in preview release and is subject to change.

AWS Security Hub can generate exposure findings for AWS Lambda (Lambda) functions.

On the Security Hub console, the Lambda function involved in an exposure finding and its identifying information are listed in the **Resources** section of the finding details. Programmatically, you can retrieve resource details with the GetFindingsV2 operation of the Security Hub API.

After identifying the resource involved in an exposure finding, you can delete the resource if you don't need it. Deleting a nonessential resource can reduce your exposure profile and AWS costs. If the resource is essential, follow these recommended remediation steps to help mitigate the risk. The remediation topics are divided based on the type of trait.

A single exposure finding contains issues identified in multiple remediation topics. Conversely, you can address an exposure finding and bring down its severity level by addressing just one remediation topic. Your approach to risk remediation depends on your organizational requirements and workloads.

> **Note**
>
> The remediation guidance provided in this topic might require additional consultation in other AWS resources.

> **Note**
>
> The remediation guidance provided in this topic might require additional consultation in other AWS resources.

Contents

- Lambda function is deployed outside of an Amazon VPC
- The Lambda function is able to assume an IAM role
- The IAM Role associated with the Lambda function has an Administrative access policy
- The IAM Role associated with the Lambda function has a policy with administrative access to an AWS Service

- Reachability traits for Lambda functions
  - The Lambda function can be publicly invoked
- Vulnerability traits for Lambda functions
  - The Lambda function has network-exploitable software vulnerabilities
  - The Lambda function has software vulnerabilites

Misconfiguration traits for Lambda functions

Here are misconfiguration traits for Lambda functions and suggested remediation steps.

Lambda function is running an unsupported runtime

Lambda allows developers to run code without provisioning or managing servers through runtimes that execute your code in a managed environment. Lambda automatically applies patches and security updates to managed runtimes and their corresponding container base images. When a runtime version is no longer supported, it no longer receives security updates, bug fixes, or technical support. Functions running on deprecated runtimes can have security vulnerabilities and may eventually stop working due to issues like certificate expiration. Additionally, unsupported runtimes may be vulnerable to newly discovered security exploits without available patches. Following security best practices, we recommend using patched, supported runtimes for Lambda functions.

Upgrade function runtime

In the **Resources** tab of the exposure, open the resource with the hyperlink. This will open the Lambda function window. To upgrade your function to a supported runtime, configure the runtime management configuration. You can choose to have your function automatically update to the latest runtime version, but before selecting this option, assess if automatic upgrades could impact your running applications. For more information, see Understanding how Lambda manages runtime version updates.

Lambda function is deployed outside of an Amazon VPC

Lambda functions by default are deployed with access to the public internet. This default configuration gives Lambda functions the ability to reach AWS service endpoints and external APIs, but it also exposes them to potential security risks. Functions with internet access could be used to exfiltrate data, communicate with unauthorized servers, or become entry points for external actors if compromised. Amazon VPC provides network isolation by restricting your Lambda functions to communicate only with resources within your defined private network. Following standard security principles, we recommend deploying Lambda functions within a VPC to improve security through network isolation.

Attach function to VPC

In the exposure finding, open the resource with the hyperlink. This will open the Lambda function window. To secure your Lambda function by restricting its network access, attach it to a VPC that has the appropriate network controls in place. Before attaching your function to a VPC, plan for any AWS service access it may need, as functions in private subnets without NAT gateways or VPC endpoints won't be able to reach AWS service APIs. For information about how to attach a Lambda function to an Amazon VPC in your account, see Attaching Lambda functions to an Amazon VPC in your AWS account. Consider using VPC endpoints for service connectivity without internet access if your function requires to access AWS services from within a private subnet. Configure a NAT Gateway if you require outbound internet connectivity from private subnets.

The Lambda function is able to assume an IAM role

Lambda functions use IAM roles to interact with AWS services. These roles grant permissions for the Lambda function to access AWS resources during execution. While these roles are sometimes necessary for Lambda functions to perform their tasks, these roles should follow the principle of least privilege. Following standard security principles, AWS recommends that you review whether the permissions attached to the role are appropriate based on the function's intended functionality.

1. **Determine if attached IAM Role is required**

   Determine whether the Lambda function requires an IAM execution role to be configured. Most Lambda functions need basic permissions to operate, such as writing logs to CloudWatch. Review the permissions attached to the function's execution role and determine whether the IAM Role is required for the function. For information on Lambda execution roles, see Defining Lambda function permissions with an execution role in the *AWS Lambda Developer Guide*.

2. **Implement least privilege access**

   Replace overly permissive policies with those that grant only the specific permissions required for the function to operate. For information about security best practices for IAM roles, see Apply least-privilege permissions in the *AWS Identity and Access Management User Guide*. To identify unnecessary permissions, you can use the IAM Access Analyzer to understand how to modify your policy based on access history. For more information, see Findings for external and unused access in the *AWS Identity and Access Management User Guide*. Alternatively, you can create a new IAM role to avoid impacting other Lambda functions that are using the existing role. In this scenario, create a new IAM role, then associate the new IAM role with the instance. For instructions on replacing an IAM role for a function, see Update a function's execution role in the *AWS Lambda Developer Guide*.

The IAM Role associated with the Lambda function has an Administrative access policy

Administrative access policies provide Lambda functions with broad permissions to AWS services and resources. These policies typically include permissions that are not required for functionality. Providing an IAM identity with an administrative access policy on a Lambda function, instead of the minimum set of permissions that the execution role needs, can increase the scope of an attack if the function is compromised. Following standard security principles, AWS recommends that you grant least privileges, which means that you grant only the permissions required to perform a task.

1. **Review and identify administrative policies**

   In the exposure finding, identify the role name. Go to the IAM dashboard and find the role with the role name identified previously. Review the permissions policy attached to the IAM role. If the policy is an AWS managed policy, look for `AdministratorAccess` or `IAMFullAccess`. Otherwise, in the policy document, look for statements that have the statements `"Effect": "Allow", "Action": "*"` and `"Resource": "*"` together.

2. **Implement least privilege access**

   Replace administrative policies with those that grant only the specific permissions required for the function to operate. For more information on security best practices for IAM roles, see Apply least-privilege permissions in the *AWS Identity and Access Management User Guide*. To identify unnecessary permissions, you can use the IAM Access Analyzer to understand how to modify your policy based on access history. For more information, see Findings for external and unused access in the *AWS Identity and Access Management User Guide*. Alternatively, you can create a new IAM role to avoid impacting other Lambda functions using the existing role. In this scenario,

create a new IAM role. Then associate the new role with the instance. For information about replacing an IAM role for a function, see Update a function's execution role in the *AWS Lambda Developer Guide*.

3. **Secure configuration considerations**

   If administrative access permissions are necessary for the instance, consider implementing these additional security controls to mitigate risk:

   - **Multi-factor authentication (MFA)** – MFA adds an additional security layer by requiring an additional form of authentication. This helps prevent unauthorized access even if credentials are compromised. For more information, see Require multi-factor authentication (MFA) in the *AWS Identity and Access Management User Guide*.
   - **IAM conditions** – Setting up condition elements allows you to restrict when and how administrative permissions can be used based on factors like source IP or MFA age. For more information, see Use conditions in IAM policies to further restrict access in the *IAM User Guide*.
   - **Permission boundaries** – Permission boundaries establish the maximum permissions a role can have, providing guardrails for roles with administrative access. For more information, see Use permissions boundaries to delegate permissions management within an account in the *AWS Identity and Access Management User Guide*.

The IAM Role associated with the Lambda function has a policy with administrative access to an AWS Service

Service admin policies allow Lambda functions to perform all actions within a specific AWS service. These policies typically grant more permissions than necessary for a function's operation. If a Lambda function with a service admin policy is compromised, an attacker could use those permissions to potentially access or modify sensitive data or services within your AWS environment. Following standard security principles, we recommend that you grant least privileges, which means that you grant only the permissions required to perform a task.

1. **Review and identify administrative policies**

   In the exposure finding, identify the role name in the ARN. Go to the IAM dashboard, and find the role name. Review the permissions policy attached to the role. If the policy is an AWS managed policy, look for `AdministratorAccess` or `IAMFullAccess`. Otherwise, in the policy document, look for statements that have the statements `"Effect": "Allow", "Action": "*"` and `"Resource": "*"`.

2. **Implement least privilege access**

Replace administrative policies with those that grant only the specific permissions required for the function to operate. For more information, see Apply least-privilege permissions in the *AWS Identity and Access Management User Guide*. To identify unnecessary permissions, you can use the IAM Access Analyzer to understand how to modify your policy based on access history. For more information, see Findings for external and unused access in the *AWS Identity and Access Management User Guide*. Alternatively, you can create a new IAM role to avoid impacting other Lambda functions that are using the existing role. In this scenario, create a new IAM role, then associate the new IAM role with the instance. For instructions on replacing an IAM role for a function, see Update a function's execution role in the *AWS Lambda Developer Guide*.

3. **Secure configuration considerations**

   If service-level administrative permissions are necessary for the instance, consider implementing these additional security controls to mitigate risk:

   - **Multi-factor authentication (MFA)** – MFA adds an additional security layer by requiring an additional form of authentication. This helps prevent unauthorized access even if credentials are compromised. For more information, see Require multi-factor authentication (MFA) in the *AWS Identity and Access Management User Guide*.
   - **IAM conditions** – Setting up condition elements allows you to restrict when and how administrative permissions can be used based on factors like source IP or MFA age. For more information, see Use conditions in IAM policies to further restrict access in the *AWS Identity and Access Management User Guide*.
   - **Permissions boundaries** – Permission boundaries establish the maximum permissions a role can have, providing guardrails for roles with administrative access. For more information, see Use permissions boundaries to delegate permissions management in the *AWS Identity and Access Management User Guide*.

Reachability traits for Lambda functions

Here are reachability traits for Lambda functions and suggested remediation steps.

The Lambda function can be publicly invoked

Lambda resource-based policies determine who can invoke your functions. A function with a resource policy that includes "*" as the principal (or no principal at all) allows any authenticated AWS users to invoke it. This creates significant risk, especially for functions that process sensitive data, modify resources, or have elevated permissions. Unauthorized users could exploit this

configuration to perform unwanted operations, potentially exposing data, manipulating resources, or abusing function capabilities. Following security best practices, we recommend restricting Lambda function access to only authorized principals.

Modify the function's resource-based policy

In the **Resources** tab of the exposure, open the resource with the hyperlink. This will open the Lambda function window. Restrict access to your Lambda function by specifying only authorized AWS account IDs or specific IAM principals (users, roles, or services) in the resource-based policy. You can only modify resource-based policies programmatically.

Vulnerability traits for Lambda functions

Here are vulnerability traits for Lambda functions and suggested remediation steps.

The Lambda function has network-exploitable software vulnerabilities

Software packages used in Lambda function code can contain Common Vulnerabilities and Exposures (CVEs) that have a high chance of being exploited. Critical CVEs pose significant security risks to your AWS environment. Attackers can exploit these unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of data, or to access other systems. Critical vulnerabilities with high exploitation likelihood represent immediate security threats, as exploit code may already be publicly available and actively used by attackers or automated scanning tools. Following security best practices, we recommend patching these vulnerabilities to protect your function from attack.

Update affected functions

Review the **References** section in the **Vulnerability** tab for the trait. Vendor documentation may include specific remediation guidance. Update the vulnerable libraries to their latest secure versions following the vendor recommended procedures. Typically, the remediation workflow depends on whether you deployed the Lambda package by uploading a zip file or by creating a Lambda function with a container image. After updating the libraries, update the Lambda function code to use the fixed version. Afterwards, deploy the updated version.

The Lambda function has software vulnerabilites

Lambda functions often use third-party libraries and dependencies that may contain security vulnerabilities with lower severity or exploitability compared to critical CVEs. While these non-critical vulnerabilities might not be as immediately exploitable, they still represent security

weaknesses that could be chained together with other vulnerabilities to compromise your function. Over time, new exploit techniques might also emerge that elevate the risk of these vulnerabilities. Following standard security principles, we recommend patching these vulnerabilities to maintain a secure environment.

Review the **References** section in the **Vulnerability** tab for the trait. Vendor documentation may include specific remediation guidance. Update the vulnerable libraries to their latest secure versions following the vendor recommended procedures. Typically, the remediation workflow depends on whether you deployed the Lambda package by uploading a zip file or by creating a Lambda function with a container image. After updating the libraries, update the Lambda function code to use the fixed version. Afterwards, deploy the updated version.

Remediating exposures for Amazon RDS functions

> **Note**
>
> Security Hub is in preview release and is subject to change.

AWS Security Hub can generate exposure findings for Amazon RDS functions.

On the Security Hub console, the Amazon RDS function involved in an exposure finding and its identifying information are listed in the **Resources** section of the finding details. Programmatically, you can retrieve resource details with the GetFindingsV2 operation of the Security Hub API.

After identifying the resource involved in an exposure finding, you can delete the resource if you don't need it. Deleting a nonessential resource can reduce your exposure profile and AWS costs. If the resource is essential, follow these recommended remediation steps to help mitigate the risk. The remediation topics are divided based on the type of trait.

A single exposure finding contains issues identified in multiple remediation topics. Conversely, you can address an exposure finding and bring down its severity level by addressing just one remediation topic. Your approach to risk remediation depends on your organizational requirements and workloads.

> **Note**
>
> The remediation guidance provided in this topic might require additional consultation in other AWS resources.

Contents

Misconfiguration traits for Amazon RDS functions

The following describes the misconfiguration traits and remediation steps for Amazon RDS functions.

The Amazon RDS DB instance is configured with public access

Amazon RDS instances with public access are potentially accessible over the internet through their endpoints. While public access is sometimes necessary for instance functionality, this configuration can be used as a potential attack vector for unauthorized users to attempt to access your database. Publicly accessible databases can be exposed to port scanning, brute force attacks, and exploitation attempts. Following standard security principles, we recommend that you limit public exposure of your database resources.

1. **Modify public access settings**

   In the exposure finding, open the resource with the hyperlink. This will open the affected DB instance. Evaluate whether the DB instance requires public accessibility based on your

application architecture. For more information, see Setting up public or private access in Amazon RDS.

The Amazon RDS DB cluster has a snapshot that's shared publicly

Public snapshots can be accessed by any AWS account, potentially exposing sensitive data to unauthorized users. Any AWS account has permission to copy these public snapshots and create DB instances from them, which could lead to data breaches or unauthorized data access. Following security best practices, we recommend restricting access to your Amazon RDS snapshots to only trusted AWS accounts and organizations.

1. Configure an Amazon RDS snapshot for private access

In the exposure finding, open the resource through the hyperlink. For information how about how to modify snapshot sharing settings, see Sharing a snapshot in the *Amazon Aurora User Guide.* For information about how to stop sharing snapshots, see Stopping snapshot sharing in the *Amazon Aurora User Guide.*.

The Amazon RDS DB instance has a snapshot that is not encrypted at rest

Unencrypted Amazon RDS DB instance snapshots may expose sensitive data if unauthorized access to the storage layer is obtained. Without encryption, data in snapshots could potentially be exposed through unauthorized access. This creates a risk of data breaches and compliance violations. Following security best practices, we recommend encrypting all database resources and their backups to maintain data confidentiality.

In the exposure finding, open the resource with the hyperlink. This will open the affected snapshot. You cannot directly encrypt an existing unencrypted snapshot. Instead, create an encrypted copy of the unencrypted snapshot. For detailed instructions, see DB cluster snapshot copying and Encrypting Amazon RDS resources in the *Amazon Aurora User Guide...*

The Amazon RDS DB cluster has a snapshot that is not encrypted at rest

Unencrypted Amazon RDS DB cluster snapshots may expose sensitive data if unauthorized access to the storage layer is obtained. Without encryption, data in snapshots could potentially be exposed through unauthorized access. This creates a risk of data breaches and compliance violations. Following security best practices, we recommend encrypting all database resources and their backups to maintain data confidentiality.

1. Create an encrypted copy of the snapshot

In the exposure finding, open the resource with the hyperlink. This will open the affected snapshot. You cannot directly encrypt an existing unencrypted snapshot. Instead, create an encrypted copy of the unencrypted snapshot. For detailed instructions, see DB cluster snapshot copying and Encrypting Amazon RDS resources in the *Amazon Aurora User Guide*..

The Amazon RDS DB instance has an open security group

Security groups act as virtual firewalls for your Amazon RDS instances to control inbound and outbound traffic. Open security groups, which allow unrestricted access from any IP address, may expose your database instances to unauthorized access and potential attacks. Following standard security principles, we recommend restricting security group access to specific IP addresses and ports to maintain the principle of least privilege.

Review security group rules and assess current configuration

In the exposure finding, open the resource for the DB instance Security Group. Evaluate which ports are open and accessible from broad IP ranges, such as (`0.0.0.0/0 or ::/0`). For information about viewing security group details, see DescribeSecurityGroups in the *Amazon Elastic Compute Cloud API Reference*.

Modify security group rules

Modify your security group rules to restrict access to specific trusted IP addresses or ranges. When updating your security group rules, consider separating access requirements for different network segments by creating rules for each required source IP range or restricting access to specific ports. To modify security group rules, see Configure security group rules in the *Amazon EC2 User Guide*. To modify the default port of an existing Amazon RDS database instance, see Modifying the DB cluster by using the console, CLI, and API in the *Amazon Aurora User Guide*.

The Amazon RDS DB instance has IAM database authentication disabled

IAM database authentication allows you to authenticate to your Amazon RDS database using IAM credentials instead of database passwords. This provides several security benefits, such as centralized access management, temporary credentials, and elimination of storing database passwords in application code. IAM database authentication allows authentication to database instances with an authentication token instead of a password. As a result, network traffic to and from the database instance is encrypted using SSL. Without IAM authentication, databases typically rely on password-based authentication, which can lead to password reuse and weak passwords. Following security best practices, we recommend enabling IAM database authentication.

Enable IAM database authentication

In the exposure finding, open the resource with the hyperlink. This will open the affected DB instance. You can enable IAM database authentication in the Database options. For more information, see Enabling and disabling IAM database authentication in the *Amazon RDS User Guide*. After enabling IAM authentication, update your DB instances to use IAM authentication instead of password based authentication.

The Amazon RDS DB instance uses the default admin username

Using default usernames (e.g., “admin”, “root”) for DB instances increases security risk as these are widely known and commonly targeted in brute force attacks. Default usernames are predictable and make it easier for unauthorized users to attempt to gain access to your database. With default usernames, attackers only need to obtain passwords rather than needing both to gain access to your database. Following security best practices, we recommend using unique administrator usernames for your database instance to enhance security through obscurity and reduce the risk of unauthorized access attempts.

Configure a unique administrator username

In the exposure finding, open the resource with the hyperlink. This will open the affected DB instance. Consider what backup frequency, retention period, and lifecycle rules are best for your applications.

The Amazon RDS DB cluster uses the default admin username

Using default usernames (e.g., “admin”, “root”) for DB instances increases security risk as these are widely known and commonly targeted in brute force attacks. Default usernames are predictable and make it easier for unauthorized users to attempt to gain access to your database. With default usernames, attackers only need to obtain passwords rather than needing both to gain access to your database. Following security best practices, we recommend using unique administrator usernames for your database instance to enhance security through obscurity and reduce the risk of unauthorized access attempts.

Configure a unique administrator username

In the exposure finding, open the resource with the hyperlink. This will open the affected DB instance. You cannot change the administrator username of an existing Amazon RDS DB instance. To create a unique administrator name, you need to create a new DB instance with a custom username and migrate your data.

The Amazon RDS DB instance has automatic minor version upgrades disabled

Automatic minor version upgrades ensure that your Amazon RDS instances automatically receive minor engine version upgrades when they become available. These upgrades often include important security patches and bug fixes that help maintain the security and stability of your database. Your database is at risk of running with known security vulnerabilities that have been fixed in newer minor versions. Without automatic updates, database instances can accumulate security vulnerabilities as new CVEs are discovered. Following security best practices, we recommend enabling automatic minor version upgrades for all Amazon RDS instances.

Enable automatic minor version upgrades

In the exposure finding, open the resource with the hyperlink. This will open the affected DB instance. You can view automatic minor upgrade settings in the **Maintenance & backups** tab. For more information, see Automatic minor version upgrades for Amazon RDS for MySQL You can also configure your maintenance window to occur during periods of low database activity.

The Amazon RDS DB instance has automated backups disabled

Automated backups provide point-in-time recovery for your Amazon RDS instances, allowing you to restore your database to any point within your retention period. When automated backups are disabled, you risk losing data in case of malicious deletion, data corruption, or other data loss scenarios. In the event of malicious activity like ransomware attacks, database table deletion, or corruption, the ability to restore to a point in time before the incident reduces the time required to recover from an incident. Following security best practices, we recommend enabling automated backups with an appropriate retention period for all production databases.

The Amazon RDS DB instance has deletion protection disabled

Database deletion protection is a feature that helps prevent the deletion of your database instances. When deletion protection is disabled, your database can be deleted by any user with sufficient permissions, potentially resulting in data loss or application downtime. Attackers can delete your database, leading to service disruption, data loss, and increased recovery time. Following security best practices, we recommend enabling deletion protection for your RDS DB instances to prevent malicious deletion.

Enable delete protection for your Amazon RDS DB cluster

In the exposure finding, open the resource with the hyperlink. This will open the affected DB cluster.

The Amazon RDS DB cluster has deletion protection disabled

Database deletion protection is a feature that helps prevent the deletion of your database instances. When deletion protection is disabled, your database can be deleted by any user with sufficient permissions, potentially resulting in data loss or application downtime. Attackers can delete your database, leading to service disruption, data loss, and increased recovery time. Following security best practices, we recommend enabling deletion protection for your RDS DB clusters to prevent malicious deletion.

Enable delete protection for your Amazon RDS DB cluster

In the exposure finding, open the resource with the hyperlink. This will open the affected DB cluster.

The Amazon RDS DB instance uses the default port for the database engine

Amazon RDS instances that use default ports for database engines may face increased security risks, as these default ports are widely known and are often targeted by automated scanning tools. Modifying your DB instance to use non-default ports adds an additional layer of security through obscurity, making it more difficult for unauthorized users to perform automated or targeted attacks on your database. Default ports are commonly scanned for by unauthorized persons, and may cause your DB instance to be targeted. Following security best practices, we recommend changing the default port to a custom port to reduce the risk of automated or targeted attacks.

In the exposure finding, open the resource with the hyperlink. This will open the affected DB instance.

Update application connection strings

After changing the port, update all applications and services that connect to your Amazon RDS instance to use the new port number.

The Amazon RDS DB instance is not covered by a backup plan

AWS Backup is a fully managed backup service that centralizes and automates the backup of data across AWS services. If your DB instance is not covered by a backup plan, you risk losing data in case of malicious deletion, data corruption, or other data loss scenarios. In the event of malicious activity like ransomware attacks, database table deletion, or corruption, the ability to restore to a point in time before the incident reduces the time required to recover from an incident. Following security best practices, we recommend including your Amazon RDS instances in a backup plan to ensure data protection.

Create and assign a backup plan for your DB instance

In the exposure finding, open the resource with the hyperlink. This will open the affected DB instance. Consider what backup frequency, retention period, and lifecycle rules are best for your applications.

Remediating exposures for Amazon S3 buckets

> **Note**
>
> Security Hub is in preview release and is subject to change.

AWS Security Hub can generate exposure findings for Amazon Simple Storage Service (S3) buckets.

On the Security Hub console, the Amazon S3 bucket involved in an exposure finding and its identifying information are listed in the **Resources** section of the finding details. Programmatically, you can retrieve resource details with the GetFindingsV2 operation of the Security Hub API.

After identifying the resource involved in an exposure finding, you can delete the resource if you don't need it. Deleting a nonessential resource can reduce your exposure profile and AWS costs. If the resource is essential, follow these recommended remediation steps to help mitigate the risk. The remediation topics are divided based on the type of trait.

A single exposure finding contains issues identified in multiple remediation topics. Conversely, you can address an exposure finding and bring down its severity level by addressing just one remediation topic. Your approach to risk remediation depends on your organizational requirements and workloads.

> **Note**
>
> The remediation guidance provided in this topic might require additional consultation in other AWS resources.

Contents

  - Amazon S3 bucket is not encrypted at rest with AWS KMS keys
  - Multi-factor authentication (MFA) delete is disabled on a versioned Amazon S3 bucket
  - The Amazon S3 bucket allows principals from other AWS accounts to modify bucket permissions
- Reachability traits for Amazon S3 buckets
  - Amazon S3 bucket has public access
  - The Amazon S3 bucket has public read access
  - The Amazon S3 bucket has write access
  - The Amazon S3 access point has public access settings enabled
- Sensitive data traits for Amazon S3 buckets
  - Sensitive data traits for Amazon S3 buckets

Misconfiguration traits for Amazon S3 buckets

Here are misconfiguration traits for Amazon S3 buckets and suggested remediation steps.

The Amazon S3 bucket has versioning disabled

Amazon S3 Versioning helps you keep multiple variants of an object in the same bucket. When versioning is disabled, Amazon S3 stores only the most recent version of each object, meaning that if objects are accidentally or maliciously deleted or overwritten, they cannot be recovered. Versioning-enabled buckets provide protection against accidental deletion, application failures, and security incidents like ransomware attacks, where unauthorized modification or deletion of data could occur. Following security best practices, we recommend enabling versioning for buckets containing important data that would be difficult or impossible to recreate if lost.

1. **Enable versioning** – To enable Amazon S3 Versioning on a bucket, see Enabling versioning on buckets in the *Amazon Simple Storage Service User Guide*. When enabling versioning, consider implementing lifecycle rules to manage storage, as versioning will maintain multiple copies of objects.

The Amazon S3 bucket has Object Lock disabled

Amazon S3 Object Lock provides a write-once-read-many (WORM) model for Amazon S3 objects, preventing them from being deleted or overwritten for a fixed period or indefinitely. When Object Lock is disabled, your objects could be vulnerable to accidental or malicious deletion, modification,

or encryption by ransomware. Object Lock is especially important for compliance with regulatory requirements that demand immutable data storage and for protection against sophisticated threats like ransomware that may attempt to encrypt your data. By enabling Object Lock, you can enforce retention policies as an added layer of data protection and create an immutable backup strategy for your critical data. Following security best practices, we recommend you enable Object Lock to prevent malicious modification of your objects.

1. Note that Object Lock can only be enabled when creating a new bucket, so you will need to create a new bucket with Object Lock enabled. For large migrations, consider using Batch Operations to copy objects to the new bucket. Before you lock any objects, you must also enable Amazon S3 Versioning and Object Lock on a bucket. Since Object Lock can only be enabled on new buckets, you'll need to migrate existing data to a new bucket with Object Lock enabled. **Configure Amazon S3 Object Lock** – For information about how to configure Object Lock on a bucket, see ConfiguringAmazon S3Object Lock in the *Amazon Simple Storage Service User Guide*. After setting up Object Lock, choose an appropriate retention mode according to your environment.

Amazon S3 bucket is not encrypted at rest with AWS KMS keys

Amazon S3 applies server-side encryption with Amazon S3 managed keys as the default level of encryption for all new buckets. While Amazon S3 managed keys provides strong encryption protection, it doesn't offer the same level of access control and audit capabilities as AWS Key Management Service keys. When using KMS keys, access to objects requires permissions to both the Amazon S3 bucket and the KMS key that encrypted the object. This is particularly important for sensitive data where you need granular control over who can access the encrypted objects and comprehensive audit logging of encryption key usage. Following security best practices, we recommend using KMS keys to encrypt buckets containing sensitive data or for environments with strict compliance requirements.

1. **Configure Amazon S3 bucket key**

   To configure a bucket to use an Amazon S3 bucket key for new objects, see Configuring your bucket to use an Amazon S3 Bucket Key with SSE-KMS for new objects in the *Amazon Simple Storage Service User Guide*. For information about how to encrypt an existing object, see Encrypting objects with Amazon S3 Batch Operations in the AWS Storage Blog.

When implementing AWS KMS encryption, consider the following:

- **Key management** – Decide on whether to use an AWS managed key or a customer managed key (CMK). CMKs offer customers full control over the lifecycle and usage of their keys. For more information on the difference between these two types of keys, see AWS KMS keys in the *AWS Key Management Service Developer Guide*.
- **Key rotation** – For additional security measures, enable automatic key rotation for your KMS keys. For more information, see Enable automatic key rotation in the *AWS Key Management Service Developer Guide*.

Multi-factor authentication (MFA) delete is disabled on a versioned Amazon S3 bucket

Multi-factor authentication (MFA) delete provides an additional layer of security for your Amazon S3 bucket. It requires multi-factor authentication for destructive Amazon S3 operations. When MFA delete is disabled, users with appropriate permissions can permanently delete object versions or suspend versioning on your bucket without additional authentication challenges. Enabling MFA delete helps protect against unauthorized or accidental deletion of your data, providing enhanced protection against ransomware attacks, insider threats, and operational errors. MFA delete is particularly valuable for buckets containing critical or compliance-sensitive data that must be protected from unauthorized deletion. Following security best practices, we recommend enabling MFA for your Amazon S3 buckets.

1. **Review MFA types**

   AWS supports the following MFA types. Although authentication with a physical device typically provides more stringent security protection, using any type of MFA is more secure than having MFA disabled.

2. **Enforce MFA at the resource policy level**

   Use the `aws:MultiFactorAuthAge` condition key in a bucket policy to require MFA for sensitive operations. For more information, see see Requiring MFA in the *Amazon Simple Storage Service User Guide*.

3. **Enable MFA**

   To enable MFA delete, first, ensure that versioning is enabled on your Amazon S3 bucket. MFA delete is only supported on buckets that have versioning enabled. For information about how to enable Amazon S3 versioning, see Enabling versioning on buckets in the *Amazon Simple Storage Service User Guide*. MFA delete cannot be enabled through the Amazon S3 console. You must

use the Amazon S3 API or the AWS CLI. For more information, see Configuring MFA delete in the *Amazon Simple Storage Service User Guide*.

The Amazon S3 bucket allows principals from other AWS accounts to modify bucket permissions

Amazon S3 bucket policies control access to buckets and objects. When bucket policies allow principals from other AWS accounts to modify bucket permissions, unauthorized users can reconfigure your bucket. If external principal credentials are compromised, unathorized users can gain control over your bucket, leading to data breaches or service disruptions. Following standard security principles, AWS recommends that you restrict permission management actions to trusted principals only.

1. **Review and identify bucket policies**

   In the the exposure, identify the Amazon S3 bucket in the ARN field. In the Amazon S3 console, select the bucket, and navigate to the **Permissions** tab to review the bucket policy. Review the permissions policy attached to the bucket. Look for policy statements that grant actions like `s3:PutBucketPolicy, s3:PutBucketAcl, s3:DeleteBucketPolicy, s3:*` or policy statements that allow access to principals outside your account, as denoted in the principal statement.

2. **Modify the bucket policy**

   Modify the bucket policy to remove or restrict actions granted to other AWS accounts:

   - Remove policy statements that grant external accounts permission management actions.
   - If cross-account access is required, replace broad permissions (`s3:*`) with specific actions that don't include bucket permission management.

   For information about modifying a bucket policy, see Adding a bucket policy by using the Amazon S3 console in the *Amazon S3 User Guide*.

Reachability traits for Amazon S3 buckets

Here are reachability traits for Amazon S3 buckets and suggested remediation steps.

Amazon S3 bucket has public access

By default, Amazon S3 buckets and objects are private, but they can be made public through various configurations. If you modify bucket policies, access point policies, or object permissions to allow public access, you risk exposing sensitive data.

1. **Assess bucket**

   Assess whether your bucket can be made private based on your organizational policy, compliance requirements, or data classification. If you didn't intend to grant bucket access to the public or other AWS accounts, follow the remaining remediation instructions.

2. **Configure the bucket to be private**

   Choose one of the following options to configure private access for your Amazon S3 bucket:

   - **Account level** – To block public access for all buckets in your account using account-level settings, see Configuring block public access settings for your account in the *Amazon Simple Storage Service User Guide*.
   - **Bucket level** –To block public access for a specific bucket, see Configuring block public access settings for yourAmazon S3buckets in the *Amazon Simple Storage Service User Guide*.
   - **Bucket ACL or policies** –To modify the bucket access control list (ACL), bucket policy, Multi-Region Access Point (MRAP) policy, or access point policy to remove public access to the bucket, see Reviewing and changing bucket access in the *Amazon Simple Storage Service User Guide*. If you block public access at the account level or bucket level, those blocks take precedence over a policy that permits public access.

The Amazon S3 bucket has public read access

Amazon S3 buckets with public read access allow anyone on the internet to view the contents of your bucket. While this may be necessary for publicly accessible websites or shared resources, it can create security risks if the bucket contains sensitive data. Public read access can lead to unauthorized viewing and downloading, which can lead to data breaches if sensitive data is stored in those buckets. Following standard security principles, AWS recommends restricting access to Amazon S3 buckets to necessary users and systems.

1. **Block public access at the bucket level**

   Amazon S3 provides Block Public Access settings that can be configured at both the bucket and account levels to prevent public access regardless of bucket policies or ACLs. For more

information, see Blocking public access to your Amazon S3 storage in the *Amazon Simple Storage Service User Guide*. After blocking public access, review your bucket access control configuration to make sure it aligns with your access requirements. Then review your Amazon S3 bucket policy to explicitly define who can access your bucket. For examples of bucket policies see Examples of Amazon S3 bucket policies in the *Amazon Simple Storage Service User Guide*.

2. **Alternative access methods**

   If public read access is required, consider these more secure alternatives:

   - **CloudFront** – Use CloudFront with an Origin Access Identity (OAI) or Origin Access Control (OAC) to allow read access from a private Amazon S3 bucket. This alternative restricts direct access to your Amazon S3 bucket while allowing content to be publicly accessible through CloudFront. For more information, see Restricting access to an Amazon Amazon S3 origin in the *Amazon CloudFront Developer Guide*.
   - **Presigned URLs** – Use presigned URLs for temporary access to specific objects. For more information, see Sharing objects with presigned URLs in the *AWSAmazon S3User Guide*.

The Amazon S3 bucket has write access

Amazon S3 buckets with public write access allow anyone on the internet to upload, modify, or delete objects in your bucket. This creates significant security risks, including the potential for someone to upload malicious files, modify existing files, and delete data. Public write access creates security vulnerabilities that can be exploited by attackers. Following standard security principles, AWS recommends restricting write access to your Amazon S3 buckets to only necessary users and systems.

1. **Block public access at the account and bucket level**

   Amazon S3 provides block public access settings that can be configured at both the bucket and account levels to prevent public access regardless of bucket policies or ACLs. For more information, see Blocking public access to your Amazon S3 storage in the *Amazon Simple Storage Service User Guide*.

2. **Modify bucket policies**

   For a more granular approach to remove public write access, review the bucket policy. You can look for `s3:PutObject`, `s3:DeleteObject`, or `s3:*`. For more information on managing bucket policies, see Bucket policies for Amazon S3 in the *Amazon Simple Storage Service User Guide*.

3. **Alternative access methods** If public read access is required, consider these more secure alternatives:
   - **CloudFront** – Use CloudFront with an Origin Access Identity (OAI) or Origin Access Control (OAC) to allow read access from a private Amazon S3 bucket. This alternative restricts direct access to your Amazon S3 bucket while allowing content to be publicly accessible through CloudFront. For more information, see Restricting access to an Amazon S3 origin in the *Amazon CloudFront Developer Guide*.
   - **Presigned URLs** – Use presigned URLs for temporary access to specific objects. For more information, see Sharing objects with presigned URLs in the *Amazon Simple Storage Service User Guide*.

The Amazon S3 access point has public access settings enabled

Amazon S3 access points provide customized access to shared datasets in Amazon S3 buckets. When you enable public access for an access point, anyone on the internet to access to your data. Following standard security principles, AWS recommends restricting public access to Amazon S3 access points.

1. **Create a new access point with block public access enabled**

   Amazon S3 doesn't support changing an access point's public access settings after an access point has been created. For information about creating an access point, see Managing public access to access points for general purpose buckets in the *Amazon S3 User Guide*. For more information about managing public access to access points, see Creating access points for general purpose buckets in the *Amazon S3 User Guide*.

Sensitive data traits for Amazon S3 buckets

Here are the sensitive data traits for Amazon S3 buckets and suggested remediation steps.

Sensitive data traits for Amazon S3 buckets

When Macie identifies sensitive data in your Amazon S3 buckets, it indicates potential security and compliance exposures that require immediate attention.

Sensitive data can include:

- Credentials

- Personally identifiable information
- Financial information
- Confidential content requiring protection

If sensitive data is exposed through misconfiguration or unauthorized access, it could lead to compliance violations, data breaches, identity theft, or financial loss. Following security best practices, AWS recommends proper classification of data and continuous monitoring of sensitive data in your Amazon S3 buckets.

Implement controls for sensitive data

In the exposure finding, choose the **Open resource** . Review the type of sensitive data detected and its location in the bucket. For help interpreting Macie findings, see Types of Macie findings in the *Amazon Macie User Guide*.

Based on the type of sensitive data discovered, implement the appropriate security controls:

- **Restrict access to the bucket** – Review bucket permissions to make sure they follow the principle of least privilege. Use IAM policies, bucket policies, and ACLs to restrict access. For more information, see Identity and Access Management for Amazon S3 in the *Amazon Simple Storage Service User Guide*.
- **Enable server-side encryption** – Enable server-side encryption with KMS keys keys for additional protection. For more information, see Using server-side encryption with AWS KMS keys (SSE-KMS) in the *Amazon Simple Storage Service User Guide*.
- **Use AWS Glue DataBrew** – Use Glue DataBrew for data preparation and cleaning. For more information, seeWhat is AWS Glue DataBrew in the *AWS Glue DataBrew Developer Guide*.

Attack sequence findings in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

An attack sequence is a type of security threat. GuardDuty generates findings for attack sequences when events match a pattern of suspicious activity. If you enable GuardDuty, you can access this type of finding in the Security Hub console. This allows you to investigate and remediate all of your

security threats without switching consoles. You can review your attack sequence findings from the **Threats** screen of the Security Hub console.

Reviewing attack sequence findings

> **Note**
>
> Security Hub is in preview release and is subject to change.

You can review your attack sequence findings from the **Threats** screen of the Security Hub console. The following topic describes how you review details for an attack sequence finding.

Reviewing details for attack sequence findings

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to review details about attack sequence findings in the Security Hub console and with the API.

Reviewing details for attack sequences in the Security Hub console

The following describes how to review details for attack sequences in the Security Hub console:

To review attack sequence findings in the console

1. Sign in using your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, choose **Threats**.
3. From the list of attack sequence findings, choose the attack sequence finding you want to view details for.

Reviewing details for attack sequence findings with the API

You can review attack sequence findings with the `GetFindingsV2` API or the AWS CLI. You can filter results with the `FindingProviderFields` parameter and by providing a filter value of

`TTPs/AttackSequence` if you only want to return attack sequence findings. You can filter by other fields to narrow down results.

Example command

The following is a AWS CLI example that retrieves the 10 most recently generated attack sequence findings in your account. The example is formatted for Linux, macOS, and Unix, and the backslash character (\) is used to improve readability.

```
$ aws securityhub get-findings-v2 \
--filters '{"FindingProviderFieldsTypes":[{"Value": "TTPs/
AttackSequence","Comparison":"PREFIX"}]}' \
--sort-criteria '{ "Field": "LastObservedAt", "SortOrder": "desc"}' \
--max-items 10
```

Remediating attack sequence findings

> **Note**
>
> Security Hub is in preview release and is subject to change.

For information about remediating attack sequence findings, see Remediating detected GuardDuty security findings in the *Amazon GuardDuty User Guide*.

Automations in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

Security Hub includes features that automatically modify and take action on findings based on your specifications.

Security Hub currently supports the following types of automations:

- **Automation rules** – Automatically update and suppress findings, as well as send findings to ticketing tools, in near real time based on defined criteria.

- **Automated response and remediation** – Create custom Amazon EventBridge rules that define automatic actions to take against specific findings and insights.

Automation rules are helpful when you want to automatically update finding fields in the Open Cybersecurity Schema Framework (OCSF). For example, you can use an automation rule to update the severity level of findings for resources with a specific tag. Using the automation rule eliminates the need to manually update the severity level of each finding related to the specific tag. You can configure automation rules to create tickets in tools like Jira Cloud and ServiceNow when findings match specific attributes. This allows findings to be created into tickets as soon as they are sent to Security Hub or created in Security Hub.

EventBridge rules are helpful when you want to take actions outside of Security Hub CSPM with regards to specific findings or send specific findings to third-party tools for remediation or additional investigation. The rules can be used to trigger supported actions, such as invoking an AWS Lambda function or notifying an Amazon Simple Notification Service (Amazon SNS) topic about a specific finding.

Automation rules take effect before EventBridge rules are applied. That is, automation rules are triggered and update a finding before EventBridge receives the finding. EventBridge rules then apply to the updated finding.

Automation rules in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

With Security Hub, you can automate tasks like updating finding details and creating tickets for third-party integrations.

Automation rules and AWS Regions

Automation rules can be created in one AWS Region and then applied in all configured AWS Regions. When using region aggregation, you can only create rules in the home region. When creating rules in the home region, any rule you define is applied to all linked regions, unless your rule criteria excludes a specific linked region. You must create an automation rule for any region that's not a linked region.

Rule actions and criteria

Automation rules in Security Hub use criteria to reference OCSF attributes in Security Hub findings. For example, the filters supported for the `Criteria` parameter in `CreateAutomationRuleV2` match the filters supported for the `Criteria` parameter in `GetFindingsV2`. This means filters used in automation rules can be used to get findings. Security Hub supports the following OCSF fields for automation rule criteria.

OCSF field	Console filter value	Filter operators	Field type
activity_name	Activity name	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
class_name	Finding class name	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
cloud.account.uid	Account ID	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
cloud.provider	Cloud provider	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
cloud.region	Region	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUAL	String

OCSF field	Console filter value	Filter operators	Field type
		S, PREFIX_NO T_EQUALS	
comment	Comment	EQUALS, PREFIX, NOT_CONTA INS, NOT_EQUAL S, PREFIX_NO T_EQUALS	String
complianc e.assessm ents.category	Assessment category	EQUALS, PREFIX, NOT_CONTA INS, NOT_EQUAL S, PREFIX_NO T_EQUALS	String
complianc e.assessm ents.name	Assessment name	EQUALS, PREFIX, NOT_CONTA INS, NOT_EQUAL S, PREFIX_NO T_EQUALS	String
complianc e.control	Security control ID	EQUALS, PREFIX, NOT_CONTA INS, NOT_EQUAL S, PREFIX_NO T_EQUALS	String
complianc e.standards	Applicable standards	EQUALS, PREFIX, NOT_CONTA INS, NOT_EQUAL S, PREFIX_NO T_EQUALS	String

OCSF field	Console filter value	Filter operators	Field type
compliance.status	Compliance status	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
finding_info.desc	Finding description	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
finding_info.related_events.product.uid	Related findings product ID	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
finding_info.related_events.title	Related findings title	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
finding_info.related_events.uid	Related findings ID	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
finding_info.src_url	Source URL	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String

OCSF field	Console filter value	Filter operators	Field type
finding_info.types	Finding type	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
finding_info.uid	Provider ID	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
metadata.product.feature.uid	Generator ID	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
metadata.product.name	Product name	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
metadata.product.uid	Product ARN	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
metadata.product.vendor_name	Company name	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String

OCSF field	Console filter value	Filter operators	Field type
metadata.uid	Finding ID	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
remediation.desc	Recommendation text	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
remediation.references	Recommendation URL	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
resources.cloud_partition	Resource partition	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
resources.name	Resource name	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
resources.region	Resource region	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String

OCSF field	Console filter value	Filter operators	Field type
resources.type	Resource type	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
resources.uid	Resource ID	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
severity	Severity	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
status	Status	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
vulnerabilities.fix_coverage	Software vulnerabilities coverage	EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
finding_info.first_seen_time_dt	First observed at	Start, End, DateRange	Date (formatted as 2022-12-01T21:47:39.269Z)

OCSF field	Console filter value	Filter operators	Field type
finding_info.last_seen_time_dt	Last observed at	Start, End, DateRange	Date (formatted as 2022-12-01T21:47:39.269Z)
finding_info.modified_time_dt	Updated at	Start, End, DateRange	Date (formatted as 2022-12-01T21:47:39.269Z)
compliance.assessments.meets_criteria	Compliance assessment meets criteria	True, False	Boolean
vulnerabilities.is_exploit_available	Software vulnerabilities with exploit available	True, False	Boolean
vulnerabilities.is_fix_available	Software vulnerabilities with fix available	True, False	Boolean
activity_id	Activity ID	Eq (equal-to), Gte (greater-than-equal), Lte (less-than-equal)	Number
compliance.status_id	Compliance status ID	Eq (equal-to), Gte (greater-than-equal), Lte (less-than-equal)	Number

OCSF field	Console filter value	Filter operators	Field type
confidence_score	Confidence	Eq (equal-to), Gte (greater-than-equal), Lte (less-than-equal)	Number
severity_id	Severity ID	Eq (equal-to), Gte (greater-than-equal), Lte (less-than-equal)	Number
status_id	Status ID	Eq (equal-to), Gte (greater-than-equal), Lte (less-than-equal)	Number
finding_info.related_events_count	Related findings count	Eq (equal-to), Gte (greater-than-equal), Lte (less-than-equal)	Number
resources.tags	Resource tags	EQUALS	Map

For criteria labeled as string fields, using different filter operators on the same field affects the evaluation logic. For more information, see StringFilter in the *Security Hub API Reference*.

Each criterion supports a maximum number of values that can be used to filter matching findings. For the limits of each criterion, see OcsfFindingFilters in the *Security Hub API Reference*

OCSF fields that can be updated

The following are the OCSF fields that can be updated using automation rules.

- `Comment`
- `SeverityId`
- `StatusId`

How automation rules evaluate findings

An automation rule evaluates new and updated findings that Security Hub generates or ingests after you create the rule.

Automation rules evaluate original, provider-supplied findings. Providers can supply new findings and update existing findings through their integration with Security Hub. Rules aren't triggered when you update finding fields after rule creation through the `BatchUpdateFindingsV2` operation. If you create an automation rule and make a `BatchUpdateFindingsV2` update that both affect the same finding field, the last update sets the value for that field. Take the following example:

You use `BatchUpdateFindingsV2` to update the `Status` field of a finding from `New` to `In Process`. If you call `GetFindingsV2`, the `Status` field now has a value of `In Process`. You create an automation rule that changes the `Status` field of the finding from `New` to `Suppressed` (recall that rules ignore updates made with `BatchUpdateFindingsV2`). The finding provider updates the finding and changes the `Status` field to `New`. If you call `GetFindingsV2`, the `Status` field now has a value of `Suppressed` because the automation rule was applied, and the rule was the last action taken on the finding.

When you create or edit a rule on the Security Hub console, the console displays a preview of findings that match the rule criteria. Whereas automation rules evaluate original findings sent by the finding provider, the console preview reflects findings in their final state as they would be shown in a response to the `GetFindingsV2` API operation (that is, after rule actions or other updates are applied to the finding).

How automation rules are ordered

Each automation rule is assigned a rule order. This determines the order in which Security Hub applies your automation rules, and becomes important when multiple rules relate to the same finding or finding field.

When multiple rule actions relate to the same finding or finding field, the rule with the highest numerical value for rule order applies last and has the ultimate effect.

When you create a rule in the Security Hub console, Security Hub automatically assigns rule order based on the order of rule creation. The first rule you create will have a rule order of 1. When more than one rule exists each subsequently created rule will have the next highest available numerical value for rule order.

When you create a rule through `CreateAutomationRuleV2` API or AWS CLI, Security Hub applies the rule with the lowest numerical value for `RuleOrder` first. It then applies subsequent rules in ascending order. If multiple findings have the same `RuleOrder`, Security Hub applies a rule with an earlier value for the `UpdatedAt` field first (that is, the rule which was most recently edited applies last).

You can modify rule order at any time.

Example of rule order:

Rule A (rule order is 1):

- Rule A criteria
  - `ProductName` = `Security Hub CSPM`
  - `Resources.Type` is `S3 Bucket`
  - `Compliance.Status` = `FAILED`
  - `RecordState` is `NEW`
  - `Workflow.Status` = `ACTIVE`
- Rule A actions
  - Update `Confidence` to 95
  - Update `Severity` to `CRITICAL`
  - Update `Comment` to `This needs attention`

Rule B (rule order is 2):

- Rule B criteria
  - `AwsAccountId` = `123456789012`
- Rule B actions
  - Update `Severity` to `INFORMATIONAL`

First, Rule A actions apply to Security Hub findings that match Rule A criteria. Then, Rule B actions apply to Security Hub findings with the specified account ID. In this example, since Rule B applies last, the end value of `Severity` in findings from the specified account ID is `INFORMATIONAL`. Based on the Rule A action, the end value of `Confidence` in matched findings is 95.

Third-party integrations

You can use automation rules to create tickets for integrations with Jira Cloud and ServiceNow ITSM. For more information, see Creating a rule for a third-party integration.

Scenarios where automation rules do not work

The following are scenarios where automation rules do not work.

- The standalone account becomes a member of an organization with a delegated admin
- The organization management account removes the delegated admin and sets a new delegated admin
- The aggregator configuration for the delegated admin or standalone account changes when an unlinked region is made a linked region

During these scenarios, a member of an organization can manage automation rules with list, get, and delete operations in the AWS CLI or APIs.

When an unlinked region is made a linked region, the delegated admin or standalone account can manage resources in a linked region with list, get, and delete operations.

Creating automation rules in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to create automation rules. You can use automation rules to update details for a finding or create a ticket for a third-party integration. You must create automation rules individually and in the AWS Region where you want them applied. However, if you create an automation rule in an aggregation region, it will be applied in all regions. Otherwise, if you create an automation rule in a non-linked region, it will be applied just in that region.

Creating a rule that updates finding details

The following procedure describes how to create a rule that updates finding details.

1. Sign in to your AWS account. Open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, under **Management**, choose **Automations**.
3. Choose **Create rule**.
4. Under **Details**, enter a name for your automation rule.
   - (Optional) Enter a description for your automation rule.
5. Under **Actions**, choose **Update findings details**. You can search for criteria and add criteria in the search bar. To check if any findings match your criteria, choose **Preview matching findings**.
6. Under **Update finding details**, choose at least one finding detail to update when a finding matches your criteria. You can choose **Severity**, **Status**, or **Comment**.
7. Under **Rule settings**, select **Enabled** or **Disabled**. If you select **Enabled**, the automation rule is enabled and will process new findings. If you select **Disabled**, the automation rule is disabled and will not process any findings.
8. (Optional) Under **Tags**, choose **Add new tag** to enter a key-value pair to be applied to your automation rule.
9. Choose **Create rule**.

Creating a rule for a third-party integration

The following procedure describes how to create a rule that creates a ticket for a third-party integration. For information about the integrations Security Hub CSPM supports, see Third-party integrations for Security Hub CSPM.

1. Sign in to your AWS account. Open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, under **Management**, choose **Automations**.
3. Choose **Create rule**.
4. Under **Details**, enter a name for your automation rule.
   - (Optional) Enter a description for your automation rule.

5. Under **Actions**, choose **Create ticket**. You can search for criteria and add criteria in the search bar. To check if any findings match your criteria, choose **Preview matching findings**.
6. Under **Create a ticket**, choose an IT ticketing integration from the dropdown, and then choose **Add integration**.
7. Under **Rule settings**, select **Enabled** or **Disabled**. If you select **Enabled**, the automation rule is enabled and will process new findings. If you select **Disabled**, the automation rule is disabled and will not process any findings.
8. (Optional) Under **Tags**, choose **Add new tag** to enter a key-value pair to be applied to your automation rule.
9. Choose **Create rule**.

Viewing details for automation rules in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to view details for automation rules. You can view the following details for an automation rule:

- Name and description
- Actions and criteria
- Rule status and existing findings
- Tags

To view details for an automation rule

1. Sign in to your AWS account. Open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, under **Management**, choose **Automations**. This screen shows all of your created automation rules.
3. Select the automation rule you want to view. Choose **View details**. Alternatively, you can choose the name of the automation rule you want to view.

Updating the rule order in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to update the rule order for automation rules in the console. If you want to edit the criteria for an automation rule, see Editing automation rules in Security Hub

You cannot update the rule order for one automation rule without updating the rule order for every automation rule. For example, you have four automation rules: Rule A (1), Rule B (2), Rule C (3), and Rule D (4). You want Rule D to be applied first. To do this, you change its number from 4 to 1. As a result, Rule A gets 2, Rule B gets 3, and Rule C gets 4.

To update the rule order for your automation rules

1. Sign in to your AWS account. Open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, under **Management**, choose **Automations**.
3. Select the automation rule you want to edit. Under **Order**, choose the pencil icon next to the order number. Use the arrows to determine the new order number. Choose the ✓ icon to confirm. Choose the **X** icon to cancel. Alternatively, you can choose **Change order** to move the automation rule down, up, or to the top of the list.

Disabling automation rules in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to disable automation rules. When you disable automation rules, Security Hub stops applying them. You can disable automation rules any time and in the AWS Region where you created them.

To disable a rule

1. Sign in to your AWS account. Open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, under **Management**, choose **Automations**.
3. Select the automation rule you want to disable. Under **Status**, choose the pencil icon next to **Enabled**. Choose the dropdown, and then choose **Disabled**. Choose the ✓ icon to confirm. Choose the **X** icon to cancel. You can also choose **Disable** from the **Actions** dropdown.

Enabling an automation rule in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to enable automation rules. When you enable automation rules, Security Hub resumes applying them. You can enable automation rules any time and in the AWS Region where you created them.

To enable a rule

1. Sign in to your AWS account. Open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, under **Management**, choose **Automations**.
3. Select the automation rule you want to enable. Under **Status**, choose the pencil icon next to **Disabled**. Choose the dropdown, and then choose **Enabled**. Choose the ✓ icon to confirm. Choose the **X** icon to cancel. You can also choose **Enable** from the **Actions** dropdown.

Duplicating automation rules in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to duplicate automation rules. Duplicating automation rules can help you save time if you want to avoid creating them from scratch. When you duplicate automation rules, you can update details, actions, rule settings, and tags. You can edit automation rules in the AWS Region where you created them.

To duplicate a rule

1. Sign in to your AWS account, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, choose **Management**, and choose **Automations**.
3. Select the automation rule that you want to duplicate. Then choose **Actions**, and then choose **Duplicate**.
4. Make and review your changes, and then choose **Create rule**.

Editing automation rules in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to edit automation rules. You can edit automation rules in the AWS Region where you created them.

You can edit the following for automation rules:

- Name and description
- Actions
- Rule settings

You cannot edit tags for automation rules.

To edit a rule

1. Sign in to your AWS account. Open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, under **Management**, choose **Automations**.

3. Select the automation rule that you want to edit. Choose **Actions**, and then choose **Edit**.
4. Make and review your edits.
5. Choose **Save changes**.

Deleting automation rules in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to delete automation rules. You can delete automation rules in the AWS Region where you created them.

To delete a rule

1. Sign in to your AWS account, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, choose **Management**, and then choose **Automations**.
3. Select the automation rule that you want to delete. Choose **Actions**, and then choose **Delete**.
4. Choose **Delete**.

Automation rules in EventBridge

> **Note**
>
> Security Hub is in preview release and is subject to change.

You can use automation rules in Amazon EventBridge, to respond to Security Hub findings. Security Hub sends findings to EventBridge as events in near real time. You can write basic rules that indicate what automated actions to take when an events match the rules. Actions that can be automatically triggered include the following:

- Configuring an API destination in EventBridge.
- Invoking Amazon EC2 run commands

- Invoking Lambda functions
- Invoking Step Functions state machines
- Notifying an Amazon SNS topic or an Amazon SQS queue
- Relaying events to Kinesis Data Streams
- Sending a finding to a third-party ticketing, chat, SIEM, or incident response and management tool
- Sending an event to an EventBridge bus in another AWS account

Security Hub sends new findings and updated findings to EventBridge as events. Then you configure EventBridge rules to respond to each Security Hub event. For more information, see What is EventBridge? in the *EventBridge User Guide*.

> **Note**
>
> As a best practice, make sure users with permission to access EventBridge use AWS Identity and Access Management policies that grant the minimum required permissions. For more information, see EventBridge and AWS Identity and Access Management in the *EventBridge User Guide*.

EventBridge event types

> **Note**
>
> Security Hub is in preview release and is subject to change.

Security Hub uses the following Amazon EventBridge event types to integrate with EventBridge.

On the EventBridge dashboard for Security Hub, **All Events** includes all of these event types.

Findings Imported V2

Security Hub automatically sends all new findings and all updates to existing findings to EventBridge as **Findings Imported V2** events. Each **Findings Imported V2** event contains a single finding.

Every finding that's imported and every finding updated through a `BatchUpdateFindingsV2` request triggers a **Findings Imported V2** event.

For administrator accounts, the event feed in EventBridge includes events for findings from both their account and from their member accounts.

In an aggregation Region, the event feed includes events for findings from the aggregation Region and the linked Regions. Cross-Region findings are included in the event feed in near real time.

You can define rules in EventBridge that automatically route findings to a remediation workflow, third-party tool, or other supported EventBridge target. The rules can include filters that only apply the rule if the finding has specific attribute values.

You use this method to automatically send all findings, or all findings that have specific characteristics, to a response or remediation workflow.

EventBridge event formats

> **Note**
>
> Security Hub is in preview release and is subject to change.

The **Findings Imported V2** event type uses the following event format.

Example

This format is used when Security Hub sends an event to EventBridge.

```
{
   "version":"0",
   "id":"CWE-event-id",
   "detail-type":"Findings Imported V2",
   "source":"aws.securityhub",
   "account":"111122223333",
   "time":"2019-04-11T21:52:17Z",
   "region":"us-west-2",
   "resources":[
      "e51603d1054aad9d9f498d82d6e81acf4cf6bc88140e8ad2273123c73b81084"
   ],
   "detail":{
      "findings": [{
```

```
          <finding content>
        }]
    }
}
```

Each event sends a single finding. *<finding content>* is the content in JSON of the finding sent by the event.

For a complete list of finding attributes, see OCSF findings in Security Hub CSPM.

Configuring rules for EventBridge

> **Note**
>
> Security Hub is in preview release and is subject to change.

You can create a rule in Amazon EventBridge that defines an action to take when a **Findings Imported V2** event is received. **Findings Imported V2** events are triggered by updates through `BatchUpdateFindingsV2`.

Each rule contains an event pattern, which identifies the events that trigger the rule. The event pattern always contains the event source (`aws.securityhub`) and the event type (**Findings Imported V2**). The event pattern can also specify filters to identify the findings that the rule applies to.

The event rule then identifies the rule targets. The targets are the actions to take when EventBridge receives a **Findings Imported V2** event and the finding matches the filters.

The instructions provided here use the EventBridge console. When you use the console, EventBridge automatically creates the required resource-based policy that enables EventBridge to write to Amazon CloudWatch Logs.

You can also use the `PutRule` operation of the EventBridge API. However, if you use the EventBridge API, then you must create the resource-based policy. For information about the required policy, see CloudWatch Logs permissions in the *Amazon EventBridge User Guide*.

Format of the event pattern

The format of the event pattern for **Findings Imported V2** events is as follows:

```
{
  "source": [
    "aws.securityhub"
  ],
  "detail-type": [
    "Findings Imported V2"
  ],
  "detail": {
    "findings": {
      <attribute filter values>
    }
  }
}
```

- `source` identifies Security Hub as the service that generates the event.
- `detail-type` identifies the type of event.
- `detail` is optional and provides the filter values for the event pattern. If the event pattern does not contain a `detail` field, then all findings trigger the rule.

You can filter the findings based on any finding attribute. For each attribute, you provide a comma-separated array of one or more values.

```
"<attribute name>": [ "<value1>", "<value2>"]
```

If you provide more than one value for an attribute, then those values are joined by `OR`. A finding matches the filter for an individual attribute if the finding has any of the listed values. For example, if you provide both `INFORMATIONAL` and `LOW` as values for `Severity.Label`, then the finding matches if it has a severity label of either `INFORMATIONAL` or `LOW`.

The attributes are joined by `AND`. A finding matches if it matches the filter criteria for all of the provided attributes.

When you provide an attribute value, it must reflect the location of that attribute within the AWS Open Cybersecurity Schema Framework (OCSF) structure.

In the following example, the event pattern provides filter values for `ProductArn` and `Severity.Label`, so a finding matches if it is generated by Amazon Inspector and it has a severity label of either `INFORMATIONAL` or `LOW`.

```
{
    "source": [
        "aws.securityhub"
     ],
    "detail-type": [
        "Findings Imported V2"
    ],
    "detail": {
        "findings": {
            "ProductArn": ["arn:aws:securityhub:us-east-1::product/aws/inspector"],
            "Severity": {
                "Label": ["INFORMATIONAL", "LOW"]
            }
        }
    }
}
```

Creating an event rule

You can use a predefined event pattern or a custom event pattern to create a rule in EventBridge. If you select a predefined pattern, EventBridge automatically fills in `source` and `detail-type`. EventBridge also provides fields to specify filter values for the following finding attributes:

- `cloud.account.uid`
- `compliance.status`
- `metadata.product.name`
- `resources.uid`
- `severity`
- `status`

To create an EventBridge rule (console)

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. Using the following values, create an EventBridge rule that monitors finding events:
    - For **Rule type**, choose **Rule with an event pattern**.
    - Choose how to build the event pattern.

To build the event pattern with...	Do this...	
A template	In the **Event pattern** section, choose the following options: • For **Event source**, choose **AWS services**. • For **AWS service**, choose **Security Hub**. • For **Event type**, choose **Findings Imported V2**. • (Optional) To make the rule more specific, add filter values. For example, to limit the rule to findings with active record states, for **Specific Record state(s)**, choose **Active**.	

To build the event pattern with...	Do this...
A custom event pattern (Use a custom pattern if you want to filter findings based on attributes that do not appear in the EventBridge console.)	• In the **Event pattern** section, choose **Custom patterns (JSON editor)**, and then paste the following event pattern into the text area: `{ "source": [ "aws.securityhub" ], "detail-type": [ "Findings Imported V2" ], "detail": { "findings": { "<attribute name>": [ "<value1>", "<value2>" ] } } }` • Update the event pattern to include the attribute and attribute values that you want to use as a filter. For example, to apply the rule to findings that have a severity of `Critical`, use the following pattern example:

To build the event pattern with...	Do this...	

	<pre>{
	  "source":
	["aws.securityhub"
	],
	  "detail-type":
	["Findings Imported
	 V2"],
	  "detail":{
	  "findings":
	{
	  "Severity
	": ["Critical"]
	  }
	  }
	}</pre>	

- For **Target types**, choose **AWS service**, and for **Select a target**, choose a target such as an Amazon SNS topic or AWS Lambda function. The target is triggered when an event is received that matches the event pattern defined in the rule.

For details about creating rules, see Creating Amazon EventBridge rules that react to events in the *Amazon EventBridge User Guide*.

Third-party integrations for Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

You can enhance your security posture with third-party integrations for AWS Security Hub. With this feature you you can enable integrations that consume findings from Security Hub, allowing you to incorporate your operational, investigation, and response tools with Security Hub. Currently Security Hub supports integration with Jira Cloud and ServiceNow.

Create an KMS key to encrypt credentials

The integration procedures in this section provide you with an option to encrypt your credentials with an AWS owned key or customer managed key. An AWS owned key is a KMS key not in your AWS account because the AWS service that encrypts your credentials owns and manages the KMS key. If you want total control over the KMS key used to encrypt your credentials, create a customer managed key. A customer managed key is a KMS key that you own and manage.

Security Hub encryption operations access

This policy statement allows Security Hub to use the AWS KMS key for encryption operations. It permits Security Hub to protect your client secrets using this key. The permissions are restricted to operations related to specific Security Hub connectors through the condition block that checks the source ARN and encryption context.

```
{
    "Sid": "Allow Security Hub access to the customer managed key",
    "Effect": "Allow",
    "Principal": {
        "Service": "connector.securityhub.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt",
        "kms:ReEncrypt*"
    ],
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "aws:SourceArn": "arn:aws:securityhub:${Region}:${AccountId}:connectorv2/*"
        },
        "StringLike": {
            "kms:EncryptionContext:aws:securityhub:connectorV2Arn":
 "arn:aws:securityhub:${Region}:${AccountId}:connectorv2/*",
            "kms:EncryptionContext:aws:securityhub:providerName":
 "${CloudProviderName}"
        }
    }
}
```

Note

For *CloudProviderName*, enter JIRA_CLOUD or SERVICENOW. For *Region* and *AccountId*, enter your AWS Region and AWS account ID.

Security Hub key read access

This policy statement enables Security Hub to read metadata about the KMS key by allowing the DescribeKey operation. This permission is necessary for Security Hub to verify the key's status and configuration. The access is limited to specific Security Hub connectors through the source ARN condition.

```
{
    "Sid": "Allow Security Hub read access to the customer managed key",
    "Effect": "Allow",
    "Principal": {
        "Service": "connector.securityhub.amazonaws.com"
    },
    "Action": [
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "aws:SourceArn": "arn:aws:securityhub:${Region}:${AccountId}:connectorv2/*"
        }
    }
}
```

Note

For *Region* and *AccountId*, enter your AWS Region and AWS account ID.

IAM principal access for Security Hub operations

This policy statement grants the specified IAM role permissions to perform key operations (describe, generate, decrypt, re-encrypt, and list aliases) when interacting with Security Hub using

the CreateConnectorV2 and CreateTicketV2 APIs. The condition ensures these operations can only be performed through the Security Hub service in the specified region.

```
{
    "Sid": "Allow permissions to access key through Security Hub",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::${AccountId}:role/${RoleName}"
    },
    "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt",
        "kms:ReEncrypt*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": [
                "securityhub.${Region}.amazonaws.com"
            ]
        },
        StringLike": {
            "kms:EncryptionContext:aws:securityhub:providerName": "SERVICENOW"
        }
    }
}

{
    "Sid": "Allow read permissions to access key through Security Hub",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::${AccountId}:role/${RoleName}"
    },
    "Action": [
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": [
                "securityhub.${Region}.amazonaws.com"
            ]
        }
```

```
    }
}
```

> **Note**
>
> For *RoleName*, enter the name of the IAM role that's making calls to Security Hub. For *Region* and *AccountId*, enter your AWS Region and AWS account ID.

Integrations for AWS Security Hub Jira Cloud

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to access the Security Hub console to configure an integration for Jira Cloud. Before completing any of the procedures in this topic, you must purchase a Jira Cloud subscription plan. For information about subscription plans, see Pricing on the Atlassian website.

For accounts in an organization, only the delegated administrator can configure an integration. The delegated administrator can manually use the create ticket feature for any member account findings. Additionally, the delegated administrator can use automation rules to automatically create tickets for any findings associated with member accounts. When defining an automation rule, the delegated administrator can set criteria, which can include all member accounts or specific member accounts. For information about setting a delegated administrator, see Setting a delegated administrator account in Security Hub.

For accounts not in an organization, all aspects of this feature are available.

Prerequisites

You must complete the following prerequisites before configuring an integration for Jira Cloud. Otherwise, your integration between Jira Cloud and Security Hub will not work.

1. Install the AWS Security Hub for Jira Cloud app

The following procedure describes how to install the app.

1. Sign in to your Atlassian site as the administrator.
2. Choose **Settings**, and choose **Apps**.
3. If directed to the marketplace page, choose **Find new apps**. If directed to the apps page, choose **Explore apps**, and then search for *AWS Security Hub for Jira Cloud*. Then choose **Get it now**.

2. Create a project

This step is required if you haven't created a project. For information about how to create a project, see Create a new project in the Jira Cloud Support documentation.

Requirements for creating a project

Make sure to do the following when creating a new project.

- Choose **Software development** for the project template.
- Choose **Company-managed** for the project type.
- Make a note of the project key.

3. Add your software development projects to the AWS Security Hub for Jira Cloud app

The following procedure describes how to add your software development projects to the Security Hub for Jira Cloud app.

1. Sign in to your Atlassian site as the administrator.
2. Choose **Settings**, and choose **Apps**.
3. From the list of apps, choose **AWS Security Hub for Jira Cloud**.
4. Choose the **Connector settings** tab.
5. Under **Projects enabled**, choose **Add Jira Project**.
    a. From the dropdown, choose **Add all**, or select a project. Repeat this part of the step if you want to add more than one project, but not all projects.
    b. Choose **Save**.

You can verify which projects have been successfully installed from the **Installation Manager** tab. You can also verify configurations for fields, screens, statuses, and workflows from the **Installation Manager** tab.

> **Note**
>
> You can choose the **Installation Manager** tab to verify all of the projects you selected were installed successfully.

For additional information regarding Jira Cloud, see Jira Cloud resources on the Altlassian website.

Recommendations

The following are recommendations to consider before configuring an integration for Jira Cloud.

- Create a dedicated system account in Jira Cloud.
- Use one system account per Jira Cloud instance.

Configure an integration for Jira Cloud

Security Hub automatically creates issues in Jira Cloud. This integration allows you to send Security Hub findings to Jira Cloud, so you can manage them as part of your operational workflows. For example, you can assign ownership to issues that need investigation and remediation. You must complete the following procedure for each of your Jira Cloud projects that you want to send Security Hub findings to.

> **Note**
>
> When you create a Jira Cloud connector, you are redirected from the current AWS Region to `"https://3rdp.oauth.console.api.aws"`, so you can complete the connector registration. Afterwards, you are returned to the AWS Region where the connector is being created.

To configure an integration for Jira Cloud

1. Sign in to your AWS account with your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, choose **Management**, and then choose **Integrations**.
3. Choose **Add Jira Cloud**.

4. For **Details**, enter a unique and descriptive name for your integration, and determine whether to enter an optional description for your integration.
5. For **Security settings**, decide how to encrypt your Jira Cloud credentials in Security Hub. If you choose **Service owned key**, an AWS owned key is used to encrypt your data. If you choose **Customized key**, you must enter the ARN for an existing customized key, or create a new key by choosing **Create an AWS KMS key**. For information about how to create an KMS key, see Create a symmetric encryption KMS key .

   > **Note**
   >
   > You cannot change these settings once you complete this configuration. However, If you choose **Customized key**, you can edit your customized key policy at any time.

6. (Optional) For **Tags**, create and add a tag to your integration. You can add up to 50 tags.
7. For **Authorizations**, choose **Create connector and authorize**. A pop-up appears where you choose **Allow** to complete the authorization. After you complete the authorization, a check box appears letting you know the authorization was successful.
8. For **Configurations**, enter the Jira Cloud project ID.
9. Choose **Complete configuration**. After you complete the configuration, you can view your configured integrations in the **Configured integrations** tab.

Creating a ticket for a Jira Cloud integration

> **Note**
>
> Security Hub is in preview release and is subject to change.

After you create an integration with Jira Cloud, you can create a ticket for a finding.

> **Note**
>
> A finding will always be associated with a single ticket through its entire lifecycle. All subsequent updates to a finding after initial creation will be sent to the same ticket. If a connector associated with an automation rule is changed, the updated connector will only be used for new and incoming findings that match the rule criteria.

To create a ticket for a finding

1. Sign in to your AWS account with your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, under **Inventory**, choose **Findings**.
3. Choose a finding. In the finding, choose **Create ticket**.
4. For **Integration**, open the dropdown menu, and choose an integration. This integration is the integration you previously created when you configured the Jira Cloud project. Choose the integration where you want findings sent.
5. Choose **Create**.

Viewing a ticket for a Jira Cloud integration

> **Note**
>
> Security Hub is in preview release and is subject to change.

After you create a ticket for a finding, you can open the ticket on your Jira Cloud instance.

To view a finding on your Jira Cloud instance

1. Sign in to your AWS account with your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, under **Inventory**, choose **Findings**.
3. Choose the finding where you created the ticket.
4. In the finding, choose the ticket ID to view the ticket on your Jira Cloud instance or **View JSON**.

Integrations for ServiceNow

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes how to access the Security Hub console to configure an integration for ServiceNow ITSM. Before completing any of the procedures in this topic, you must have a subscription to ServiceNow ITSM before you can add this integration. For more information, see the pricing page on the ServiceNow website.

For accounts in an organization, only the delegated administrator can configure an integration. The delegated administrator can manually use the create ticket feature for any member account findings. Additionally, the delegated administrator can use automation rules to automatically create tickets for any findings associated with member accounts. When defining an automation rule, the delegated administrator can set criteria, which can include all member accounts or specific member accounts. For information about setting a delegated administrator, see Setting a delegated administrator account in Security Hub.

For accounts not in an organization, all aspects of this feature are available.

Prerequisites

You must complete the following prerequisites before configuring an integration for ServiceNow ITSM. Otherwise, your integration between ServiceNow ITSM and Security Hub will not work.

1. Install Security Hubfindings integration for IT Service Management (ITSM)

The following procedure describes how to install Security Hub plugin.

1. Sign into your ServiceNow ITSM instance, and then open the application navigator.
2. Navigate to the ServiceNow Store.
3. Search for *Security Hub findings integration for IT Service Management (ITSM)*, and then choose **Get** to install the application.

> **Note**
>
> In the settings for the Security Hub application, choose which action to take when new Security Hub findings are sent to your ServiceNow ITSM environment. You can choose **Do nothing**, **Create incident**, **Create problem**, or **Create both (incident/problem**

2. Configure the Client Credentials grant type for inbound OAuth requests

You must configure this grant type for inbound OAuth requests. For more information, see Client Credentials grant type for Inbound OAuth is supported in the ServiceNow Support webpage.

3. Create an OAuth application

If you already created an OAuth application, you can skip this prerequisite. For information about creating an OAuth application, see Setting up OAuth.

Configure an integration for ServiceNow ITSM

Security Hub can create incidents or problems automatically in ServiceNow ITSM.

To configure an integration for ServiceNow ITSM

1. Sign in to your AWS account with your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, choose **Management**, and then choose **Integrations**.
3. Under **ServiceNow ITSM**, choose **Add integration**.
4. For **Details**, enter a name for your integration, and determine whether to enter an optional description for your integration.
5. For **Security settings**, decide how to encrypt your Jira Cloud credentials in Security Hub. If you choose **Service owned key**, an AWS owned key is used to encrypt your data. If you choose **Customized key**, you must enter the ARN for an existing customized key, or create a new key by choosing **Create an AWS KMS key**. For information about how to create a KMS key, see Create a symmetric encryption KMS key.

   > **Note**
   >
   > You cannot change these settings once you complete this configuration. However, if you choose **Customized key**, you can edit your customized key policy at any time.

6. For **Authorizations**, enter ServiceNow ITSM URL, Client ID, and Client Secret.
7. For **Tags**, determine whether to create and add an optional tag to your integration.
8. Choose **Complete configuration**. After you complete the configuration, you can view your configured integrations in the **Configured integrations** tab.

Creating a ticket for a ServiceNow ITSM integration

> **Note**
>
> Security Hub is in preview release and is subject to change.

After you create an integration with ServiceNow ITSM, you can create a ticket for a finding.

> **Note**
>
> A finding will always be associated with a single ticket through its entire lifecycle. All subsequent updates to a finding after initial creation will be sent to the same ticket. If a connector associated with an automation rule is changed, the updated connector will only be used for new and incoming findings that match the rule criteria.

To create a ticket for a finding

1. Sign in to your AWS account with your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, under **Inventory**, choose **Findings**.
3. Choose a finding. In the finding, choose **Create ticket**.
4. For **Integration**, open the dropdown menu, and choose an integration.
5. Choose **Create**.

Viewing a ticket for a ServiceNow ITSM integration

> **Note**
>
> Security Hub is in preview release and is subject to change.

After you create a ticket for a finding, you can open the ticket on your ServiceNow ITSM instance.

To view a finding on your ServiceNow ITSM instance

1. Sign in to your AWS account with your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home?region=us-east-1.
2. From the navigation pane, under **Inventory**, choose **Findings**.
3. Choose the finding where you created the ticket.
4. In the finding, choose the ticket ID to view the ticket on your ServiceNow ITSM instance or **View JSON**.

Working in the Summary dashboard in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

This topic describes the **Summary** dashboard in the Security Hub console. This page shows an overview of your exposures, threats, top resources, and security service coverage across multiple security widgets. These widgets help you visualize exposures and threats by severity and account by security capability. Every time you open this page, data automatically refreshes.

You can customize this page by adding and removing different security widgets and setting filter criteria to retrieve specific data in each widget. Customizations to this page are saved for future use. If your account is the delegated administrator account for an organization, customizations are saved independently from member account customizations.

> **Note**
>
> We recommend that you do not include confidential, sensitive, or personally identifiable information (PII) in saved filters, custom widgets, or other related free-form text fields.

If your account is the delegated administrator account for an organization, the data includes findings for your account and member accounts. If your account is a member account or a standalone account, the data only includes findings for your account. If you configure cross-Region aggregation in Security Hub, this page shows findings from your aggregation.

The exposure summary widget

This widget shows all of your exposures by severity. You can see the frequency of each exposure in your environment. Exposures with greater severity appear first. Exposures are based on an analysis of findings and traits from Security Hub and other AWS services, such as Amazon Inspector. The list of exposures in this widget is limited to the eight highest exposures with the greatest number of critical findings. If two or more exposures have an equal number of critical findings, the list automatically groups those findings behind more recent critical findings.

The threat summary widget

This widget shows all of your threats by severity. Threats with greater severity appear first. Threats are related to a series of events and identify potential threats in your environment. They also originate in GuardDuty. The list of threats in this widget is limited to the eight threats with the highest severity. If two or more threats are of equal severity, the list automatically groups those findings behind more recent findings. You must enable GuardDuty to receive data in this widget.

The security coverage widget

This widget shows an overview of your security coverage and is based on coverage findings for supported services. It displays which coverage checks passed, failed, or are not available. Not available indicates the coverage check is unable to be completed. This can be caused by a deleted resource or a failing server.

Percentages for coverage checks point to the number of checks that passed and failed. For instance, one coverage check passes, and one coverage check fails. This indicates 50% of your checks passed, and 50% of your checks failed. In some cases, percentages are rounded to the nearest whole number.

Unlike security services such as GuardDuty, Amazon Inspector, and Macie, Security Hub CSPM publishes one coverage finding per account, which is PASS/FAIL depending on the enabled standards, such as PASS if at least 1 standard is enabled. Coverage percentages for Security Hub CSPM are the number of Security Hub CSPM coverage findings that passed to the total number of Security Hub CSPM coverage findings.

> **Note**
>
> We recommend that you do not include confidential, sensitive, or personally identifiable information (PII) in saved filters, custom widgets, or other related free-form text fields.

Viewing details about resources in Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

The **Resources** page tracks common resources across your account. You can access the **Resources** page in the Security Hub console by choosing **Resources** in the navigation pane. When you choose a resource type, you can review all of the resources associated with the resource type. You can review any findings associated with a resource.

> **Note**
>
> The delegated administrator can view all resources associated with member accounts. If you configured a home AWS Region, you can view all of your resources in your home AWS Region from linked AWS Regions.

If you choose a resource, you can review details for that resource. These details include the resource name, ID, ARN, type, and category. You can review the account ID associated with the resource, when the resource was created (timestamp), and where the resource was created (AWS Region). You also can review additional details in a JSON snippet that you can copy.

If you switch from the **Overview** tab to the **Findings** tab, you can review any findings associated with the resource. The **Findings** tab shows the name of each finding, type of each finding, and severity of each finding. You can group findings by different fields and search for findings using filters. If you choose a finding, you can review an overview of the finding, which includes information about compliance and how to remediate issues associated with the finding. If you go back to the resource, you can choose **Open resource** to review the resource in the console for its resource type. For example, if the resource is an IAM resource, you can open the resource in the IAM console.

The resources page provides you with different ways to organize and search for resources. You can group resources by type. For example, you can group resources by account ID, finding type, AWS Region, resource category, resource name, and resource type. You can search for findings using filters. Quick filters help you review resources by category, accounts, and finding types.

The benefit of the **Resources** page is that it helps you monitor your security posture, organize your resources, and review details about your resources.

Viewing exposures in Security Hub with the potential attack path graph

> **Note**
>
> Security Hub is in preview release and is subject to change.

The potential attack path graph is an interactive visualization that shows how potential attackers can access and take control of resources associated with an exposure finding. You can access this graph only in the Security Hub console and from the **Exposures** screen. When you view details for an exposure finding, the **Overview** tab includes a section called **Potential attack path**.

In this section of the **Overview** tab, you can choose and drag AWS resources in the potential attack path graph. You can focus on specific areas of the attack path graph with the zoom-in and zoom-out icons. You can expand the attack path graph in and out of fullscreen mode through the fullscreen icon. The legend codes the primary resource, involved resource, and contributing trait count by color and shows the trait categories and number of traits in the attack path graph. You can view details for a resource by choosing a resource and choosing **View resource details**. You can also copy the ID and AWS account number associated with a resource. Exposure findings with a reachability trait show the public internet and collapsed network path in the attack path graph. You can view this detail by choosing the collapsed network path node.

Disabling Security Hub

> **Note**
>
> Security Hub is in preview release and is subject to change.

If your account is not part of an organization, you can disable Security Hub in the Security Hub console at any time. When you disable Security Hub, stops ingesting findings. You also lose access to existing findings and configurations. The following procedure describes how to disable Security Hub.

To disable Security Hub

1. Sign in to your AWS account with your credentials, and open the Security Hub console at https://console.aws.amazon.com/securityhub/v2/home.
2. From the navigation pane, choose **General**.
3. In **Security Hub**, choose **Disable**. In the pop-up window, enter *Disable*, and choose **Disable**.

Introduction to AWS Security Hub CSPM

AWS Security Hub Cloud Security Posture Management (AWS Security Hub CSPM) provides you with a comprehensive view of your security state in AWS and helps you assess your AWS environment against security industry standards and best practices.

AWS Security Hub CSPM collects security data across AWS accounts, AWS services, and supported third-party products and helps you analyze your security trends and identify the highest priority security issues.

To help you manage the security state of your organization, Security Hub CSPM supports multiple security standards. These include the AWS Foundational Security Best Practices (FSBP) standard developed by AWS, and external compliance frameworks such as the Center for Internet Security (CIS), the Payment Card Industry Data Security Standard (PCI DSS), and the National Institute of Standards and Technology (NIST). Each standard includes several security controls, each of which represents a security best practice. Security Hub CSPM runs checks against security controls and generates control findings to help you assess your compliance against security best practices.

In addition to generating control findings, Security Hub CSPM also receives findings from other AWS services—such as Amazon GuardDuty, Amazon Inspector, and Amazon Macie— and supported third-party products. This gives you a single pane of glass into a variety of security-related issues. You can also send Security Hub CSPM findings to other AWS services and supported third-party products.

Security Hub CSPM offers automation features that help you triage and remediate security issues. For example, you can use automation rules to automatically update critical findings when a security check fails. You can also leverage the integration with Amazon EventBridge to trigger automatic responses to specific findings.

Topics

- Understanding cross-Region aggregation in Security Hub CSPM
- Understanding security standards in Security Hub CSPM
- Understanding security controls in Security Hub CSPM
- Understanding integrations in Security Hub CSPM
- Creating and updating findings in Security Hub CSPM
- Viewing insights in Security Hub CSPM
- Automatically modifying and acting on findings in Security Hub CSPM
- Working with the dashboard in Security Hub CSPM
- Regional limits for Security Hub CSPM
- Creating Security Hub CSPM resources with CloudFormation
- Subscribing to Security Hub CSPM announcements with Amazon SNS
- Disabling Security Hub CSPM

Benefits of Security Hub CSPM

Here are some of the key ways that Security Hub CSPM helps you monitor your compliance and security posture across your AWS environment.

Reduced effort to collect and prioritize findings

Security Hub CSPM reduces the effort to collect and prioritize security findings across accounts from integrated AWS services and AWS partner products. Security Hub CSPM processes finding data using the AWS Security Finding Format (ASFF), a standard finding format. This eliminates the need to manage findings from myriad sources in multiple formats. Security Hub CSPM also correlates findings across providers to help you prioritize the most important ones.

Automatic security checks against best practices and standards

Security Hub CSPM automatically runs continuous, account-level configuration and security checks based on AWS best practices and industry standards. Security Hub CSPM uses the results of these checks to calculate security scores, and identifies specific accounts and resources that require attention.

Consolidated view of findings across accounts and providers

Security Hub CSPM consolidates your security findings across accounts and provider products and displays results on the Security Hub CSPM console. You can also retrieve findings through

the Security Hub CSPM API, AWS CLI, or SDKs. With a holistic view of your current security status, you can spot trends, identify potential issues, and take necessary remediation steps.

Ability to automate finding updates and remediation

You can create automation rules that modify or suppress findings based on your defined criteria. Security Hub CSPM also supports an integration with Amazon EventBridge. To automate the remediation of specific findings, you can define custom actions to take when a finding is generated. For example, you can configure custom actions to send findings to a ticketing system or to an automated remediation system.

Accessing Security Hub CSPM

Security Hub CSPM is available in most AWS Regions. For a list of Regions where Security Hub CSPM is currently available, see AWS Security Hub CSPM endpoints and quotas in the *AWS General Reference*. For information about managing AWS Regions for your AWS account, see Specifying which AWS Regions your account can use in the *AWS Account Management Reference Guide*.

In each Region, you can access and use Security Hub CSPM in any of the following ways:

Security Hub CSPM console

The AWS Management Console is a browser-based interface that you can use to create and manage AWS resources. As part of that console, the Security Hub CSPM console provides access to your Security Hub CSPM account, data, and resources. You can perform Security Hub CSPM tasks by using the Security Hub CSPM console—view findings, create automation rules, create an aggregation Region, and more.

Security Hub CSPM API

The Security Hub CSPM API gives you programmatic access to your Security Hub CSPM account, data, and resources. With the API, you can send HTTPS requests directly to Security Hub CSPM. For information about the API, see the *AWS Security Hub CSPM API Reference*.

AWS CLI

With the AWS CLI, you can run commands at your system's command line to perform Security Hub CSPM tasks. In some cases, using the command line can be faster and more convenient than using the console. The command line is also useful if you want to build scripts that perform tasks. For information about installing and using the AWS CLI, see the AWS Command Line Interface User Guide.

AWS SDKs

AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms—for example, Java, Go, Python, C++, and .NET. The SDKs provide convenient, programmatic access to Security Hub CSPM and other AWS services in your preferred language. They also handle tasks such as cryptographically signing requests, managing errors, and retrying requests automatically. For information about installing and using the AWS SDKs, see Tools to Build on AWS.

Important

Security Hub CSPM only detects and consolidates findings that are generated after you enable Security Hub CSPM. It doesn't retroactively detect and consolidate security findings that were generated before you enabled Security Hub CSPM.
Security Hub CSPM only receives and processes findings in the Region where you enabled Security Hub CSPM in your account.
For full compliance with CIS AWS Foundations Benchmark security checks, you must enable Security Hub CSPM in all supported AWS Regions.

Related services

To further secure your AWS environment, consider using other AWS services in combination with Security Hub CSPM. Some AWS services send their findings to Security Hub CSPM, and Security Hub CSPM normalizes the findings into a standard format. Some AWS services can also receive findings from Security Hub CSPM.

For a list of other AWS services that send or receive Security Hub CSPM findings, see AWS service integrations with Security Hub CSPM.

Security Hub CSPM uses service-linked rules from AWS Config to run security checks for most controls. Controls refer to specific AWS services and AWS resources. For a list of Security Hub CSPM controls, see Control reference for Security Hub CSPM. You must enable AWS Config and record resources in AWS Config for Security Hub CSPM to generate most control findings. For more information, see Considerations before enabling and configuring AWS Config.

Security Hub CSPM free trial and pricing

When you enable Security Hub CSPM in an AWS account for the first time, that account is automatically enrolled in a 30-day Security Hub CSPM free trial.

When you use Security Hub CSPM during the free trial, you are charged for usage of other services that Security Hub CSPM interacts with, such as AWS Config items. You are not charged for AWS Config rules that are activated only by Security Hub CSPM security standards.

You are not charged for using Security Hub CSPM until your free trial ends.

Viewing usage details and estimated cost

Security Hub CSPM provides usage information, including an estimated 30-day cost for using Security Hub CSPM. The usage details include the time remaining in the free trial. The usage information can help you to understand what your Security Hub CSPM costs may be after the free trial ends. The usage information is also available after the free trial ends.

To display usage information (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Usage** under **Settings**.

The estimated monthly cost is based on your account's Security Hub CSPM usage for findings and security checks projected over a 30-day period.

The usage information and estimated cost are only for the current account and current Region. In an aggregation Region, the usage information and estimated cost don't include linked Regions. For more information about linked Regions, see the section called “Types of data that are aggregated”.

Pricing details

For more information about how Security Hub CSPM charges for ingested findings and security checks, see Security Hub CSPM pricing.

Concepts and terminology in Security Hub CSPM

In AWS Security Hub CSPM, we build on common AWS concepts and terminology and use these additional terms.

Account

A standard Amazon Web Services (AWS) account that contains your AWS resources. You can sign in to AWS with your account and enable Security Hub CSPM.

An account can invite other accounts to enable Security Hub CSPM and become associated with that account in Security Hub CSPM. Accepting a membership invitation is optional. If the invitations are accepted, the account becomes an administrator account, and the added accounts are member accounts. Administrator accounts can view findings in their member accounts.

If you are enrolled in AWS Organizations, then your organization designates a Security Hub CSPM administrator account for the organization. The Security Hub CSPM administrator account can enable other organization accounts as member accounts.

An account cannot be both an administrator account and a member account at the same time. An account can only have one administrator account.

For more information, see Managing administrator and member accounts in Security Hub CSPM.

Administrator account

An account in Security Hub CSPM that is granted access to view findings for associated member accounts.

An account becomes an administrator account in one of the following ways:

- The account invites other accounts to become associated with it in Security Hub CSPM. When those accounts accept the invitation, they become member accounts, and the inviting account becomes their administrator account.
- The account is designated by an organization management account as the Security Hub CSPM administrator account. The Security Hub CSPM administrator account can enable any organization account as a member account, and can also invite other accounts to be member accounts.

An account can only have one administrator account. An account cannot be both an administrator account and a member account at the same time.

Aggregation Region

Setting an aggregation Region allows you to view security findings from multiple AWS Regions in a single pane of glass.

The aggregation Region is the Region from which you view and manage findings. Findings are aggregated to the aggregation Region from linked Regions. Updates to findings are replicated across Regions.

In the aggregation Region, the **Security standards**, **Insights**, and **Findings** pages include data from all linked Regions.

For more information, see the section called “Aggregating data across Regions”.

Archived finding

A finding whose record state (`RecordState`) is `ARCHIVED`. Archiving a finding indicates that the finding provider believes that the finding is no longer relevant. Record state is different from workflow status, which tracks the status of the investigation into a finding.

Finding providers can use the BatchImportFindings operation of the Security Hub CSPM API to archive findings that they created. Security Hub CSPM automatically archives control findings that meet certain criteria. For more information, see Generating, updating, and archiving control findings.

On the Security Hub CSPM console, default filter settings exclude archived findings from finding lists and tables. You can update the settings to include archived findings. If you retrieve findings by using the GetFindings operation of the Security Hub CSPM API, the operation retrieves both archived and active findings. To exclude archived findings, you can filter the results. For example:

```
"RecordState": [
    {
        "Comparison": "EQUALS",
        "Value": "ARCHIVED"
    }
],
```

AWS Security Finding Format (ASFF)

A standardized format for the contents of findings that Security Hub CSPM aggregates or generates. The AWS Security Finding Format enables you to use Security Hub CSPM to view and analyze findings that are generated by AWS security services, third-party solutions, or Security Hub CSPM itself from running security checks. For more information, see AWS Security Finding Format (ASFF).

Control

A safeguard or countermeasure prescribed for an information system or an organization designed to protect the confidentiality, integrity, and availability of its information and to meet a set of defined security requirements. A security standard is associated with a collection of controls.

The term *security control* refers to controls that have a single control ID and title across standards. The term *standard control* refers to controls that have standard-specific control IDs and titles. Currently, Security Hub CSPM supports standard controls only in the China Regions and AWS GovCloud (US) Regions. Security controls are supported in all other Regions.

Custom action

A Security Hub CSPM mechanism for sending selected findings to EventBridge. A custom action is created in Security Hub CSPM. It is then linked to an EventBridge rule. The rule defines a specific action to take when a finding is received that is associated with the custom action ID. Custom actions can be used, for example, to send a specific finding, or a small set of findings, to a response or remediation workflow. For more information, see the section called “Creating a custom action”.

Delegated administrator account (Organizations)

In AWS Organizations, the delegated administrator account for a service is able to manage the use of a service for the organization.

In Security Hub CSPM, the Security Hub CSPM administrator account is also the delegated administrator account for Security Hub CSPM. When the organization management account first designates a Security Hub CSPM administrator account, Security Hub CSPM calls Organizations to make that account the delegated administrator account.

The organization management account must then choose the delegated administrator account as the Security Hub CSPM administrator account in all Regions.

Finding

The observable record of a security check or security-related detection. Security Hub CSPM generates and updates findings after completing security checks for controls. These are called *control findings*. Findings can also come from integrations with other AWS services and third-party products.

For more information, see the section called “Findings”.

Cross-Region aggregation

The aggregation of findings, insights, control compliance statuses, and security scores from linked Regions to an aggregation Region. You can then view all of your data from the aggregation Region and update findings and insights from the aggregation Region.

For more information, see the section called “Aggregating data across Regions”.

Finding ingestion

The import of findings into Security Hub CSPM from other AWS services and from third-party partner providers.

Finding ingestion events include both new findings and updates to existing findings.

Insight

A collection of related findings defined by an aggregation statement and optional filters. An insight identifies a security area that requires attention and intervention. Security Hub CSPM offers several managed (default) insights that you can't modify. You can also create custom Security Hub CSPM insights to track security issues that are unique to your AWS environment and usage. For more information, see the section called “Insights”.

Linked Region

When you enable cross-Region aggregation, a linked Region is a region that aggregates findings, insights, control compliance statuses, and security scores to the aggregation Region.

In a linked Region, the **Findings** and **Insights** pages contain findings only from that Region.

For more information, see the section called “Aggregating data across Regions”.

Member account

An account that has granted permission to an administrator account to view and take action on their findings.

An account becomes a member account in one of the following ways:

- The account accepts an invitation from another account.
- For an organization account, the Security Hub CSPM administrator account enables the account as a member account.

Related requirements

A set of industry or regulatory requirements that are mapped to a control.

Rule

A set of automated criteria that is used to assess whether a control is being adhered to. When a rule is evaluated, it can pass or fail. If the evaluation cannot determine whether rule passes or fails, then the rule is in a warning state. If the rule cannot be evaluated, then it is in a not available state.

Security check

A specific point-in-time evaluation of a rule against a single resource resulting in a `PASSED`, `FAILED`, `WARNING`, or `NOT_AVAILABLE` state. Running a security check produces a finding.

Security Hub CSPM administrator account

An organization account that manages Security Hub CSPM membership for an organization.

The organization management account designates the Security Hub CSPM administrator account in each Region. The organization management account must choose the same Security Hub CSPM administrator account in all Regions.

The Security Hub CSPM administrator account is also the delegated administrator account for Security Hub CSPM in Organizations.

The Security Hub CSPM administrator account can enable any organization account as a member account. The Security Hub CSPM administrator account can also invite other accounts to be member accounts.

Security standard

A published statement on a topic specifying the characteristics, usually measurable and in the form of controls, that must be satisfied or achieved for compliance. Security standards can be based on regulatory frameworks, best practices, or internal company policies. A control may be associated with one or more supported standards in Security Hub CSPM. To learn more about security standards in Security Hub CSPM, see Understanding security standards in Security Hub CSPM.

Severity

The severity assigned to a Security Hub CSPM control identifies the importance of the control. The severity of a control can be **Critical**, **High**, **Medium**, **Low**, or **Informational**. The severity assigned to control findings is equal to the severity of the control itself. To learn about how Security Hub CSPM assigns severity to a control, see Severity levels for control findings.

Workflow status

The status of an investigation into a finding. This is tracked using the `Workflow.Status` attribute.

The workflow status is initially `NEW`. If you notified the resource owner to take action on the finding, you can set the workflow status to `NOTIFIED`. If the finding is not an issue, and does not require any action, set the workflow status to `SUPPRESSED`. After you review and remediate a finding, set the workflow status to `RESOLVED`.

By default, most finding lists only include findings with a workflow status of `NEW` or `NOTIFIED`. Finding lists for controls also include `RESOLVED` findings.

For the GetFindings operation, you can include a filter for the workflow status.

```
"WorkflowStatus": [
    {
        "Comparison": "EQUALS",
        "Value": "RESOLVED"
    }
],
```

The Security Hub CSPM console provides an option to set the workflow status for findings. Customers (or SIEM, ticketing, incident management, or SOAR tools working on behalf of a customer to update findings from finding providers) can also use BatchUpdateFindings to update the workflow status.

Enabling Security Hub CSPM

There are two ways to enable AWS Security Hub CSPM, by integrating with AWS Organizations or manually.

We strongly recommend integrating with Organizations for multi-account and multi-Region environments. If you have a standalone account, it's necessary to set up Security Hub CSPM manually.

Verifying necessary permissions

After you sign up for Amazon Web Services (AWS), you must enable Security Hub CSPM to use its capabilities and features. To enable Security Hub CSPM, you first have to set up permissions

that allow you to access the Security Hub CSPM console and API operations. You or your AWS administrator can do this by using AWS Identity and Access Management (IAM) to attach the AWS managed policy called `AWSSecurityHubFullAccess` to your IAM identity.

To enable and manage Security Hub CSPM through the Organizations integration, you also should attach the AWS managed policy called `AWSSecurityHubOrganizationsAccess`.

For more information, see AWS managed policies for Security Hub.

Enabling Security Hub CSPM with Organizations integration

To start using Security Hub CSPM with AWS Organizations, the AWS Organizations management account for the organization designates an account as the delegated Security Hub CSPM administrator account for the organization. Security Hub CSPM is automatically enabled in the delegated administrator account in the current Region.

Choose your preferred method, and follow the steps to designate the delegated administrator.

Security Hub CSPM console

To designate the delegated Security Hub CSPM administrator when onboarding

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Choose **Go to Security Hub CSPM**. You're prompted to sign in to the Organizations management account.
3. On the **Designate delegated administrator** page, in the **Delegated administrator account** section, specify the delegated administrator account. We recommend choosing the same delegated administrator that you have set for other AWS security and compliance services.
4. Choose **Set delegated administrator**.

Security Hub CSPM API

Invoke the `EnableOrganizationAdminAccount` API from the Organizations management account. Provide the AWS account ID of the Security Hub CSPM delegated administrator account.

AWS CLI

Run the `enable-organization-admin-account` command from the Organizations management account. Provide the AWS account ID of the Security Hub CSPM delegated administrator account.

Example command:

```
aws securityhub enable-organization-admin-account --admin-account-id 777788889999
```

For more information about the integration with Organizations, see Integrating Security Hub CSPM with AWS Organizations.

Central configuration

When you integrate Security Hub CSPM and Organizations, you have the option to use a feature called central configuration to set up and manage Security Hub CSPM for your organization. We strongly recommend using central configuration because it lets the administrator customize security coverage for the organization. Where appropriate, the delegated administrator can allow a member account to configure its own security coverage settings.

Central configuration lets the delegated administrator configure Security Hub CSPM across accounts, OUs, and AWS Regions. The delegated administrator configures Security Hub CSPM by creating configuration policies. Within a configuration policy, you can specify the following settings:

- Whether Security Hub CSPM is enabled or disabled
- Which security standards are enabled and disabled
- Which security controls are enabled and disabled
- Whether to customize parameters for select controls

As the delegated administrator, you can create a single configuration policy for your entire organization or different configuration policies for your various accounts and OUs. For example, test accounts and production accounts can use different configuration policies.

Member accounts and OUs that use a configuration policy are *centrally managed* and can be configured only by the delegated administrator. The delegated administrator can designate specific

member accounts and OUs as *self-managed* to give the member the ability to configure its own settings on a Region-by-Region basis.

If you don't use central configuration, you must largely configure Security Hub CSPM separately in each account and Region. This is called local configuration. Under local configuration, the delegated administrator can automatically enable Security Hub CSPM and a limited set of security standards in new organization accounts in the current Region. Local configuration doesn't apply to existing organization accounts or to Regions other than the current Region. Local configuration also doesn't support the use of configuration policies.

Enabling Security Hub CSPM manually

You must enable Security Hub CSPM manually if you have a standalone account, or if you don't integrate with AWS Organizations. Standalone accounts can't integrate with AWS Organizations and must use manual enablement.

When you enable Security Hub CSPM manually, you designate a Security Hub CSPM administrator account and invite other accounts to become member accounts. The administrator-member relationship is established when a prospective member account accepts the invitation.

Choose your preferred method, and follow the steps to enable Security Hub CSPM. When you enable Security Hub CSPM from the console, you also have the option to enable the supported security standards.

Security Hub CSPM console

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. When you open the Security Hub CSPM console for the first time, choose **Go to Security Hub CSPM**.
3. On the welcome page, the **Security standards** section lists the security standards that Security Hub CSPM supports.

   Select the check box for a standard to enable it, and clear the check box to disable it.

   You can enable or disable a standard or its individual controls at any time. For information about managing security standards, see Understanding security standards in Security Hub CSPM.
4. Choose **Enable Security Hub**.

Security Hub CSPM API

Invoke the `EnableSecurityHub` API. When you enable Security Hub CSPM from the API, it automatically enables the following default security standards:

- AWS Foundational Security Best Practices
- Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0

If you do not want to enable these standards, then set `EnableDefaultStandards` to `false`.

You can also use the `Tags` parameter to assign tag values to the hub resource.

AWS CLI

Run the `enable-security-hub` command. To enable the default standards, include `--enable-default-standards`. To not enable the default standards, include `--no-enable-default-standards`. The default security standards are as follows:

- AWS Foundational Security Best Practices
- Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0

```
aws securityhub enable-security-hub [--tags <tag values>] [--enable-default-
standards | --no-enable-default-standards]
```

Example

```
aws securityhub enable-security-hub --enable-default-standards --tags
 '{"Department": "Security"}'
```

Multi-account enablement script

> **Note**
>
> Instead of this script, we recommend using central configuration to enable and configure Security Hub CSPM across multiple accounts and Regions.

The Security Hub CSPM multi-account enablement script in GitHub allows you to enable Security Hub CSPM across accounts and Regions. The script also automates the process of sending invitations to member accounts and enabling AWS Config.

The script automatically enables AWS Config resource recording for all resources, including global resources, in all Regions. It does not limit recording of global resources to a single Region. To conserve costs, we recommend recording global resources in a single Region only. If you use central configuration or cross-Region aggregation, this should be your home Region. For more information, see Recording resources in AWS Config.

There is a corresponding script to disable Security Hub CSPM across accounts and Regions.

Next steps: Posture management and integrations

After enabling Security Hub CSPM, we recommend enabling security standards and controls to monitor your security posture. After you enable controls, Security Hub CSPM begins running security checks and generating control findings that help you detect misconfigurations in your AWS environment. To receive control findings, you must enable and configure AWS Config for Security Hub CSPM. For more information, see Enabling and configuring AWS Config for Security Hub CSPM.

After enabling Security Hub CSPM, you can also leverage integrations between Security Hub CSPM and other AWS services and third-party solutions to see their findings in Security Hub CSPM. Security Hub CSPM aggregates findings from different sources and ingests them in a consistent format. For more information, see Understanding integrations in Security Hub CSPM.

Enabling and configuring AWS Config for Security Hub CSPM

AWS Security Hub CSPM uses AWS Config rules to run security checks and generate findings for most controls. AWS Config provides a detailed view of the configuration of AWS resources in your AWS account. It uses rules to establish a baseline configuration for your resources and a configuration recorder to detect whether a particular resource violates the conditions of a rule. Some rules, called AWS Config managed rules, are predefined and developed by AWS Config. Other rules are AWS Config custom rules that Security Hub CSPM develops.

AWS Config rules that Security Hub CSPM uses for controls are referred to as *service-linked rules*. Service-linked rules allow AWS services such as Security Hub CSPM to create AWS Config rules in your account.

To receive control findings in Security Hub CSPM, you must enable AWS Config in your account and turn on recording for resources that your enabled controls evaluate. This page explains how to enable AWS Config for Security Hub CSPM and turn on resource recording.

Considerations before enabling and configuring AWS Config

To receive control findings in Security Hub CSPM, your account must have AWS Config enabled in each AWS Region where Security Hub CSPM is enabled. If you use Security Hub CSPM for a multi-account environment, AWS Config must be enabled in each Region for the administrator account and all member accounts.

We strongly recommend that you turn on resource recording in AWS Config *before* you enable any Security Hub CSPM standards and controls. This helps you ensure that your control findings are accurate.

To turn on resource recording in AWS Config, you must have sufficient permissions to record resources in the AWS Identity and Access Management (IAM) role that is attached to the configuration recorder. In addition, make sure there is no IAM policy or policy managed in AWS Organizations that prevents AWS Config from having permission to record your resources. Security Hub CSPM control checks evaluate the configuration of a resource directly and don't take AWS Organizations policies into account. For more information about AWS Config recording, see Working with the configuration recorder in the *AWS Config Developer Guide*.

If you enable a standard in Security Hub CSPM but haven't enabled AWS Config, Security Hub CSPM tries to create AWS Config rules according to the following schedule:

- On the day that you enable the standard.
- The day after you enable the standard.
- 3 days after you enable the standard.
- 7 days after you enable the standard, and continuously every 7 days thereafter.

If you use central configuration, Security Hub CSPM also tries to create service-linked AWS Config rules each time you associate a configuration policy that enables one or more standards with accounts, organizational units (OUs), or the root.

Recording resources in AWS Config

When you enable AWS Config, you must specify which AWS resources you want the AWS Config configuration recorder to record. Through the service-linked rules, the configuration recorder allows Security Hub CSPM to detect changes to your resource configurations.

For Security Hub CSPM to generate accurate control findings, you must turn on recording in AWS Config for the resources that correspond to your enabled controls. It's primarily enabled controls with a *change triggered* schedule type that require resource recording. Some controls with a *periodic* schedule type also require resource recording. For a list of these controls and their corresponding resources, see Required AWS Config resources for control findings.

Warning

If you don't configure AWS Config recording correctly for Security Hub CSPM controls, it can result in inaccurate control findings, particularly in the following instances:

- You never recorded the resource for a given control, or you disabled recording of a resource before creating that type of resource. In these cases, you receive a `WARNING` finding for the control at issue, even though you might have created resources in scope of the control after you disabled recording. This `WARNING` finding is a default finding that doesn't actually evaluate the configuration state of the resource.
- You disable recording for a resource that's evaluated by a particular control. In this case, Security Hub CSPM retains the control findings that were generated before you disabled recording, even though the control isn't evaluating new or updated resources. Security Hub CSPM also changes the compliance status of the findings to `WARNING`. These retained findings might not accurately reflect a resource's current configuration state.

By default, AWS Config records all supported *Regional resources* that it discovers in the AWS Region in which it is running. To receive all Security Hub CSPM control findings, you must also configure AWS Config to record *global resources*. To conserve costs, we recommend recording global resources in a single Region only. If you use central configuration or cross-Region aggregation, this Region should be your home Region.

In AWS Config, you can choose between *continuous recording* and *daily recording* of changes in resource state. If you choose daily recording, AWS Config delivers resource configuration data at

the end of each 24–hour period if there are changes in resource state. If there are no changes, no data is delivered. This can delay the generation of Security Hub CSPM findings for change-triggered controls until a 24–hour period is complete.

For more information about AWS Config recording, see Recording AWS resources in the *AWS Config Developer Guide*.

Ways to enable and configure AWS Config

You can enable AWS Config and turn on resource recording in any of the following ways:

- **AWS Config console** – You can enable AWS Config for an account by using the AWS Config console. For instructions, see Setting up AWS Config with the console in the *AWS Config Developer Guide*.
- **AWS CLI or SDKs** – You can enable AWS Config for an account by using the AWS Command Line Interface (AWS CLI). For instructions, see Setting up AWS Config with the AWS CLI in the *AWS Config Developer Guide*. AWS software development kits (SDKs) are also available for many programming languages.
- **CloudFormation template** – To enable AWS Config for many accounts, we recommend using the AWS CloudFormation template named **Enable AWS Config**. To access this template, see AWS CloudFormation StackSet sample templates in the *AWS CloudFormation User Guide*.

  By default, this template excludes recording for IAM global resources. Ensure that you turn on recording for IAM global resources in only one AWS Region to conserve recording costs. If you have cross-Region aggregation enabled, this should be your Security Hub CSPM home Region. Otherwise, it can be any Region that Security Hub CSPM is available in that supports recording of IAM global resources. We recommend running one StackSet to record all resources, including IAM global resources, in the home Region or other selected Region. Then, run a second StackSet to record all resources except IAM global resources in other Regions.
- **GitHub script** – Security Hub CSPM offers a GitHub script that enables Security Hub CSPM and AWS Config for multiple accounts across Regions. This script is useful if you haven't integrated with AWS Organizations, or you have some member accounts that aren't part of an organization.

For more information, see the following blog post on the *AWS Security blog*: Optimize AWS Config for AWS Security Hub CSPM to effectively manage your cloud security posture.

Config.1 control

In Security Hub CSPM, the Config.1 control generates `FAILED` findings in your account if AWS Config is disabled. It also generates `FAILED` findings in your account if AWS Config is enabled but resource recording isn't turned on.

If AWS Config is enabled and resource recording is turned on, but resource recording isn't turned on for a type of resource that an enabled control checks, Security Hub CSPM generates a `FAILED` finding for the Config.1 control. In addition to this `FAILED` finding, Security Hub CSPM generates `WARNING` findings for the enabled control and the types of resources that the control checks. For example, if you enable the KMS.5 control and resource recording isn't turned on for AWS KMS keys, Security Hub CSPM generates a `FAILED` finding for the Config.1 control. Security Hub CSPM also generates `WARNING` findings for the KMS.5 control and your KMS keys.

To receive a `PASSED` finding for the Config.1 control, turn on resource recording for all the resource types that correspond to enabled controls. Also disable controls that aren't required for your organization. This helps ensure that you don't have configuration gaps in your security control checks. It also helps ensure that you receive accurate findings about misconfigured resources.

If you're the delegated Security Hub CSPM administrator for an organization, AWS Config recording must be configured correctly for your account and your member accounts. If you use cross-Region aggregation, AWS Config recording must be configured correctly in the home Region and all linked Regions. Global resources do not need to be recorded in linked Regions.

Generating the service-linked rules

For every control that uses a service-linked AWS Config rule, Security Hub CSPM creates instances of the required rule in your AWS environment.

These service-linked rules are specific to Security Hub CSPM. Security Hub CSPM creates these service-linked rules even if other instances of the same rules already exist. The service-linked rule adds `securityhub` before the original rule name and a unique identifier after the rule name. For example, for the AWS Config managed rule `vpc-flow-logs-enabled`, the service-linked rule name might be `securityhub-vpc-flow-logs-enabled-12345`.

There are quotas for the number of AWS Config managed rules that can be used to evaluate controls. AWS Config rules that Security Hub CSPM creates don't count towards those quotas. You can enable a security standard even if you've already reached the AWS Config quota for managed rules in your account. To learn more about quotas for AWS Config rules, see Service limits for AWS Config in the *AWS Config Developer Guide*.

Cost considerations

Security Hub CSPM can impact your AWS Config configuration recorder costs by updating the `AWS::Config::ResourceCompliance` configuration item. Updates can occur each time a Security Hub CSPM control associated with an AWS Config rule changes compliance state, is enabled or disabled, or has parameter updates. If you use the AWS Config configuration recorder only for Security Hub CSPM, and don't use this configuration item for other purposes, we recommend turning off recording for it in AWS Config. This can reduce your AWS Config costs. You don't need to record `AWS::Config::ResourceCompliance` for security checks to work in Security Hub CSPM.

For information about the costs associated with resource recording, see AWS Security Hub CSPM pricing and AWS Config pricing.

Understanding local configuration in Security Hub CSPM

Local configuration is the default way that an AWS organization is configured in Security Hub CSPM. If you don't opt in to and enable central configuration, your organization uses local configuration by default.

Under local configuration, the delegated Security Hub CSPM administrator account has limited control over configuration settings. The only settings that the delegated administrator can enforce are automatically enabling Security Hub CSPM and default security standards in new organization accounts. These settings apply only in the Region in which you designated the delegated administrator account. The default security standards are AWS Foundational Security Best Practices (FSBP) and Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0. Local configuration settings don't apply to existing organization accounts or to Regions other than the one in which the delegated administrator account was designated.

Aside from enabling Security Hub CSPM and default standards in new organization accounts in a single Region, you must configure other Security Hub CSPM settings, including standards and controls, separately in each Region and account. Because this can be a duplicative process, we recommend using central configuration for a multi-account environment if one or more of the following applies to you:

- You want different configuration settings for various parts of your organization (for example, different enabled standards or controls for different teams).
- You operate in multiple Regions and want to reduce the time and complexity of configuring the service across these Regions.

- You want new accounts to use specific configuration settings when they join the organization.
- You want organization accounts to inherit specific configuration settings from a parent account or root.

For information about central configuration, see Understanding central configuration in Security Hub CSPM.

Understanding central configuration in Security Hub CSPM

Central configuration is an AWS Security Hub CSPM feature that helps you set up and manage Security Hub CSPM across multiple AWS accounts and AWS Regions. To use central configuration, you must first integrate Security Hub CSPM and AWS Organizations. You can integrate the services by creating an organization and designating a delegated Security Hub CSPM administrator account for the organization.

From the delegated Security Hub CSPM administrator account, you can enable Security Hub CSPM for your organization's accounts and organizational units (OUs) across Regions. You can also enable, configure, and disable individual security standards and security controls for accounts and OUs across Regions. You can configure these settings in just a few steps from one primary Region, referred to as the *home Region*.

When you use central configuration, the delegated administrator can choose which accounts and OUs to configure. If the delegated administrator designates a member account or OU as *self-managed*, the member can configure its own settings separately in each Region. If the delegated administrator designates a member account or OU as *centrally managed*, only the delegated administrator can configure the member account or OU across Regions. You can designate all accounts and OUs in your organization as centrally managed, all self-managed, or a combination of both.

To configure centrally managed accounts, the delegated administrator uses Security Hub CSPM configuration policies. Configuration policies let the delegated administrator specify whether Security Hub CSPM is enabled or disabled, and which standards and controls are enabled or disabled. They can also be used to customize parameters for certain controls.

Configuration policies take effect in the home Region and all linked Regions. The delegated administrator specifies the organization's home Region and linked Regions before starting to use central configuration. Specifying linked Regions is optional. The delegated administrator can create

a single configuration policy for the whole organization, or create multiple configuration policies to configure variable settings for different accounts and OUs.

> **Tip**
>
> If you don't use central configuration, you must largely configure Security Hub CSPM separately in each account and Region. This is called *local configuration*. Under local configuration, the delegated administrator can automatically enable Security Hub CSPM and a limited set of security standards in new organization accounts in the current Region. Local configuration doesn't apply to existing organization accounts or to Regions other than the current Region. Local configuration also doesn't support the use of configuration policies.

This section provides an overview of central configuration.

Benefits of using central configuration

Benefits of central configuration include the following:

Simplify configuration of the Security Hub CSPM service and capabilities

When you use central configuration, Security Hub CSPM guides you through the process of configuring security best practices for your organization. It also deploys the resulting configuration policies to specified accounts and OUs automatically. If you have existing Security Hub CSPM settings, such as automatically enabling new security controls, you can use those as a starting point for your configuration policies. In addition, the **Configuration** page on the Security Hub CSPM console displays a real-time summary of your configuration policies and which accounts and OUs use each policy.

Configure across accounts and Regions

You can use central configuration to configure Security Hub CSPM across multiple accounts and Regions. This helps ensure that each part of your organization maintains a consistent configuration and adequate security coverage.

Accommodate different configurations in different accounts and OUs

With central configuration, you can choose to configure your organization's accounts and OUs in different ways. For example, your test accounts and production accounts might require different

configurations. You can also create a configuration policy that covers new accounts when they join the organization.

Prevent configuration drift

Configuration drift occurs when a user makes a change to a service or feature that conflicts with the delegated administrator's selections. Central configuration prevents this drift. When you designate an account or OU as centrally managed, it's configurable only by the delegated administrator for the organization. If you prefer a specific account or OU to configure its own settings, you can designate it as self-managed.

When to use central configuration?

Central configuration is most beneficial for AWS environments that include multiple Security Hub CSPM accounts. It's designed to help you centrally manage Security Hub CSPM for multiple accounts.

You can use central configuration to configure the Security Hub CSPM service, security standards, and security controls. You can also use it to customize parameters of certain controls. For more information about security standards, see Understanding security standards in Security Hub CSPM. For more information about security controls, see Understanding security controls in Security Hub CSPM.

Central configuration terms and concepts

Understanding the following key terms and concepts can help you use Security Hub CSPM central configuration.

Central configuration

A Security Hub CSPM feature that helps the delegated Security Hub CSPM administrator account for an organization configure the Security Hub CSPM service, security standards, and security controls across multiple accounts and Regions. To configure these settings, the delegated administrator creates and manages Security Hub CSPM configuration policies for centrally managed accounts in their organization. Self-managed accounts can configure their own settings separately in each Region. To use central configuration, you must integrate Security Hub CSPM and AWS Organizations.

Home Region

The AWS Region from which the delegated administrator centrally configures Security Hub CSPM, by creating and managing configuration policies. Configuration policies take effect in the home Region and all linked Regions.

The home Region also serves as the Security Hub CSPM aggregation Region, receiving findings, insights, and other data from linked Regions.

Regions that AWS introduced on or after March 20, 2019 are known as opt-in Regions. An opt-in Region can't be the home Region, but it can be a linked Region. For a list of opt-in Regions, see Considerations before enabling and disabling Regions in the *AWS Account Management Reference Guide*.

Linked Region

An AWS Region that is configurable from the home Region. Configuration policies are created by the delegated administrator in the home Region. The policies take effect in the home Region and all linked Regions. Specifying linked Regions is optional.

A linked Region also sends findings, insights, and other data to the home Region.

Regions that AWS introduced on or after March 20, 2019 are known as opt-in Regions. You must enable such a Region for an account before a configuration policy can be applied to it. The Organizations management account can enable opt-in Regions for a member account. For more information, see Specify which AWS Regions your account can use in the *AWS Account Management Reference Guide*.

Target

An AWS account, organizational unit (OU), or the organization root.

Security Hub CSPM configuration policy

A collection of Security Hub CSPM settings that the delegated administrator can configure for centrally managed targets. This includes:

- Whether to enable or disable Security Hub CSPM.
- Whether to enable one or more security standards.
- Which security controls to enable across the enabled standards. The delegated administrator can do this by providing a list of specific controls that should be enabled, and Security Hub CSPM disables all other controls (including new controls when they are released). Alternatively, the delegated administrator can provide a list of specific controls that should

be disabled, and Security Hub CSPM enables all other controls (including new controls when they are released).

- Optionally, customize parameters for select enabled controls across the enabled standards.

A configuration policy takes effect in the home Region and all linked Regions after it's associated with at least one account, organizational unit (OU), or the root.

On the Security Hub CSPM console, the delegated administrator can choose the Security Hub CSPM recommended configuration policy or create custom configuration policies. With the Security Hub CSPM API and AWS CLI, the delegated administrator can only create custom configuration policies. The delegated administrator can create a maximum of 20 custom configuration policies.

In the recommended configuration policy, Security Hub CSPM, the AWS Foundational Security Best Practices (FSBP) standard, and all existing and new FSBP controls are enabled. Controls that accept parameters use the default values. The recommended configuration policy applies to the entire organization.

To apply different settings to the organization, or apply different configuration policies to different accounts and OUs, create a custom configuration policy.

Local configuration

The default configuration type for an organization, after integrating Security Hub CSPM and AWS Organizations. With local configuration, the delegated administrator can choose to automatically enable Security Hub CSPM and default security standards in *new* organization accounts in the current Region. If the delegated administrator automatically enables default standards, all controls that are part of these standards are also automatically enabled with default parameters for new organization accounts. These settings don't apply to existing accounts, so configuration drift is possible after an account joins the organization. Disabling specific controls that are part of the default standards, and configuring additional standards and controls, must be done separately in each account and Region.

Local configuration doesn't support the use of configuration policies. To use configuration policies, you must switch to central configuration.

Manual account management

If you don't integrate Security Hub CSPM with AWS Organizations or you have a standalone account, you must specify settings for each account separately in each Region. Manual account management doesn't support the use of configuration policies.

Central configuration APIs

Security Hub CSPM operations that only the Security Hub CSPM delegated Security Hub CSPM administrator can use in the home Region to manage configuration policies for centrally managed accounts. The operations include:

- `CreateConfigurationPolicy`
- `DeleteConfigurationPolicy`
- `GetConfigurationPolicy`
- `ListConfigurationPolicies`
- `UpdateConfigurationPolicy`
- `StartConfigurationPolicyAssociation`
- `StartConfigurationPolicyDisassociation`
- `GetConfigurationPolicyAssociation`
- `BatchGetConfigurationPolicyAssociations`
- `ListConfigurationPolicyAssociations`

Account-specific APIs

Security Hub CSPM operations that can be used to enable or disable Security Hub CSPM, standards, and controls on an account-by-account basis. These operations are used in each individual Region.

Self-managed accounts can use account-specific operations to configure their own settings. Centrally managed accounts can't use the following account-specific operations in the home Region and linked Regions. In those Regions, only the delegated administrator can configure centrally managed accounts through central configuration operations and configuration policies.

- `BatchDisableStandards`
- `BatchEnableStandards`
- `BatchUpdateStandardsControlAssociations`
- `DisableSecurityHub`
- `EnableSecurityHub`
- `UpdateStandardsControl`

To check account status, the owner of a centrally managed account *can* use any `Get` or `Describe` operations of the Security Hub CSPM API.

If you use local configuration or manual account management, instead of central configuration, these account-specific operations can be used.

Self-managed accounts can also use `*Invitations` and `*Members` operations. However, we recommend that self-managed accounts don't use these operations. Policy associations can fail if a member account has its own members that are part of a different organization than the delegated administrator's.

Organizational unit (OU)

In AWS Organizations and Security Hub CSPM, a container for a group of AWS accounts. An organizational unit (OU) also can contain other OUs, enabling you to create a hierarchy that resembles an upside-down tree, with a parent OU at the top and branches of OUs that reach down, ending in accounts that are the leaves of the tree. An OU can have exactly one parent, and each organization account can be a member of exactly one OU.

You can manage OUs in AWS Organizations or AWS Control Tower. For more information, see Managing organizational units in the *AWS Organizations User Guide* or Govern organizations and accounts with AWS Control Tower in the *AWS Control Tower User Guide*.

The delegated administrator can associate configuration policies with specific accounts or OUs, or with the root to cover all accounts and OUs in an organization.

Centrally managed

A target that only the delegated administrator can configure across Regions by using configuration policies.

The delegated administrator account specifies whether a target is centrally managed. The delegated administrator can also change a target's status from centrally managed to self-managed, or the other way around.

Self-managed

A target that manages its own Security Hub CSPM settings. A self-managed target uses account-specific operations to configure Security Hub CSPM for itself separately in each Region. This is in contrast to centrally managed targets, which are configurable only by the delegated administrator across Regions through configuration policies.

The delegated administrator account specifies whether a target is self-managed. The delegated administrator can apply self-managed behavior to a target. Alternatively, an account or OU can inherit self-managed behavior from a parent.

The delegated administrator account can itself be a self-managed account.The delegated administrator account can change a target's status from self-managed to centrally managed, or the other way around.

Configuration policy association

A link between a configuration policy and an account, organizational unit (OU), or root. When a policy association exists, the account, OU, or root uses the settings defined by the configuration policy. An association exists in either of these cases:

- When the delegated administrator directly applies a configuration policy to an account, OU, or root
- When an account or OU inherits a configuration policy from a parent OU or the root

An association exists until a different configuration is applied or inherited.

Applied configuration policy

A type of configuration policy association in which the delegated administrator directly applies a configuration policy to target accounts, OUs, or the root. Targets are configured in the way that the configuration policy defines, and only the delegated administrator can change their configuration. If applied to root, the configuration policy affects all accounts and OUs in the organization that don't use a different configuration through application or inheritance from the closest parent.

The delegated administrator can also apply a self-managed configuration to specific accounts, OUs, or the root.

Inherited configuration policy

A type of configuration policy association in which an account or OU adopts the configuration of the closest parent OU or the root. If a configuration policy isn't directly applied to an account or OU, it inherits the configuration of the closest parent. All elements of a policy are inherited. In other words, an account or OU can't choose to selectively inherit only parts of a policy. If the closest parent is self-managed, the child account or OU inherits the self-managed behavior of the parent.

Inheritance can't override an applied configuration. That is, if a configuration policy or self-managed configuration is directly applied to an account or OU, it uses that configuration and doesn't inherit the configuration of the parent.

Root

In AWS Organizations and Security Hub CSPM, the top-level parent node in an organization. If the delegated administrator applies a configuration policy to root, the policy is associated with all accounts and OUs in the organization unless they use a different policy, through application or inheritance, or are designated as self-managed. If the administrator designates the root as self-managed, all accounts and OUs in the organization are self-managed unless they use a configuration policy through application or inheritance. If the root is self-managed and no configuration policies currently exist, all new accounts in the organization retain their current settings.

New accounts that join an organization fall under the root until they are assigned to a specific OU. If a new account isn't assigned to an OU, it inherits the root configuration unless the delegated administrator designates it as a self-managed account.

Enabling central configuration in Security Hub CSPM

The delegated AWS Security Hub CSPM administrator account can use central configuration to configure Security Hub CSPM, standards, and controls for multiple accounts and organizational units (OUs) across AWS Regions.

For background information about the benefits of central configuration and how it works, see Understanding central configuration in Security Hub CSPM.

This section explains prerequisites for central configuration and how to begin using it.

Prerequisites for central configuration

Before you can start using central configuration, you must integrate Security Hub CSPM with AWS Organizations and designate a home Region. If you use the Security Hub CSPM console, these prerequisites are included in the opt-in workflow for central configuration.

Integrate with Organizations

You must integrate Security Hub CSPM and Organizations to use central configuration.

To integrate these services, you begin by creating an organization in Organizations. From the Organizations management account, you then designate a Security Hub CSPM delegated administrator account. For instructions, see Integrating Security Hub CSPM with AWS Organizations.

Ensure that you designate your delegated administrator in your **intended home Region**. When you start using central configuration, the same delegated administrator is automatically set in all linked Regions as well. The Organizations management account *cannot* be set as the delegated administrator account.

Important

When you use central configuration, you can't use the Security Hub CSPM console or Security Hub CSPM APIs to change or remove the delegated administrator account. If the Organizations management account uses AWS Organizations APIs to change or remove the Security Hub CSPM delegated administrator, Security Hub CSPM automatically stops central configuration. Your configuration policies are also disassociated and deleted. Member accounts retain the configuration that they had before the delegated administrator was changed or removed.

Designate a home Region

You must designate a home Region to use central configuration. The home Region is the Region from which the delegated administrator configures the organization.

Note

The home Region cannot be a Region that AWS has designated as an opt-in Region. An opt-in Region is disabled by default. For a list of opt-in Regions, see Considerations before enabling and disabling Regions in the *AWS Account Management Reference Guide*.

Optionally, you can specify one or more linked Regions that are configurable from the home Region.

The delegated administrator can create and manage configuration policies only from the home Region. Configuration policies take effect in the home Region and all linked Regions. You can't create a configuration policy that applies only to a subset of these Regions, and not others. The

exception to this is controls that involve global resources. If you use central configuration, Security Hub CSPM automatically disables controls that involve global resources in all Regions except the home Region. For more information, see Controls that use global resources.

The home Region is also your Security Hub CSPM aggregation Region that receives findings, insights, and other data from linked Regions.

If you have already set an aggregation Region for cross-Region aggregation, then that's your default home Region for central configuration. You can change the home Region before you start to use central configuration by deleting your current finding aggregator and creating a new one in your desired home Region. A finding aggregator is a Security Hub CSPM resource that specifies the home Region and linked Regions.

To designate a home Region, see the steps for setting an aggregation Region. If you already have a home Region, you can invoke the GetFindingAggregator API to see details about it, including which Regions currently are linked to it.

Instructions for enabling central configuration

Choose your preferred method, and follow the steps to enable central configuration for your organization.

Security Hub CSPM console

To enable central configuration (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. On the navigation pane, choose **Settings** and **Configuration**. Then, choose **Start central configuration**.

   If you're onboarding to Security Hub CSPM, choose **Go to Security Hub CSPM**.
3. On the **Designate delegated administrator** page, select your delegated administrator account or enter its account ID. If applicable, we recommend choosing the same delegated administrator that you have set for other AWS security and compliance services. Choose **Set delegated administrator**.
4. On the **Centralize organization** page, in the **Regions** section, select your home Region. You must be signed in to the home Region to proceed. If you've already set an aggregation Region for cross-Region aggregation, it's displayed as the home Region. To change the

home Region, choose **Edit Region settings**. You can then select your preferred home Region and return to this workflow.

5. Select at least one Region to link to the home Region. Optionally, choose whether you want to automatically link future supported Regions to the home Region. The Regions you select here will be configurable from the home Region by the delegated administrator. Configuration policies take effect in your home Region and all linked Regions.
6. Choose **Confirm and continue**.
7. You can now use central configuration. Continue following the console prompts to create your first configuration policy. If you're not ready to create a configuration policy yet, choose **I'm not ready to configure yet**. You can create a policy later by choosing **Settings** and **Configuration** in the navigation pane. For instructions on creating a configuration policy, see Creating and associating configuration policies.

Security Hub CSPM API

To enable central configuration (API)

1. Using the credentials of the delegated administrator account, invoke the UpdateOrganizationConfiguration API from the home Region.
2. Set the `AutoEnable` field to `false`.
3. Set the `ConfigurationType` field in the `OrganizationConfiguration` object to `CENTRAL`. This action has the following impact:

   - Designates the calling account as the Security Hub CSPM delegated administrator in all linked Regions.
   - Enables Security Hub CSPM in the delegated administrator account in all linked Regions.
   - Designates the calling account as the Security Hub CSPM delegated administrator for new and existing accounts that use Security Hub CSPM and belong to the organization. This occurs in the home Region and all linked Regions. The calling account is set as the delegated administrator for new organization accounts only if they are associated with a configuration policy that has Security Hub CSPM enabled. The calling account is set as the delegated administrator for existing organization accounts only if they already have Security Hub CSPM enabled.
   - Sets AutoEnable to `false` in all linked Regions, and sets AutoEnableStandards to `NONE` in the home Region and all linked Regions. These parameters aren't relevant in the home

and linked Regions when you use central configuration, but you can automatically enable Security Hub CSPM and default security standards in organization accounts through the use of configuration policies.

4. You can now use central configuration. The delegated administrator can create configuration policies to configure Security Hub CSPM in your organization. For instructions on creating a configuration policy, see Creating and associating configuration policies.

Example API request:

```
{
    "AutoEnable": false,
    "OrganizationConfiguration": {
        "ConfigurationType": "CENTRAL"
    }
}
```

AWS CLI

To enable central configuration (AWS CLI)

1. Using the credentials of the delegated administrator account, run the update-organization-configuration command from the home Region.
2. Include the `no-auto-enable` parameter.
3. Set the `ConfigurationType` field in the `organization-configuration` object to `CENTRAL`. This action has the following impact:

   - Designates the calling account as the Security Hub CSPM delegated administrator in all linked Regions.
   - Enables Security Hub CSPM in the delegated administrator account in all linked Regions.
   - Designates the calling account as the Security Hub CSPM delegated administrator for new and existing accounts that use Security Hub CSPM and belong to the organization. This occurs in the home Region and all linked Regions. The calling account is set as the delegated administrator for new organization accounts only if they are associated with a configuration policy that has Security Hub enabled. The calling account is set as the delegated administrator for existing organization accounts only if they already have Security Hub CSPM enabled.

- Sets the auto-enablement option to no-auto-enable in all linked Regions, and sets auto-enable-standards to NONE in the home Region and all linked Regions. These parameters aren't relevant in the home and linked Regions when you use central configuration, but you can automatically enable Security Hub CSPM and default security standards in organization accounts through the use of configuration policies.

4. You can now use central configuration. The delegated administrator can create configuration policies to configure Security Hub CSPM in your organization. For instructions on creating a configuration policy, see Creating and associating configuration policies.

Example command:

```
aws securityhub --region us-east-1 update-organization-configuration \
--no-auto-enable \
--organization-configuration '{"ConfigurationType": "CENTRAL"}'
```

Centrally managed versus self-managed targets

When you enable central configuration, the delegated AWS Security Hub CSPM administrator can designate each organization account, organizational unit (OU), and the root as *centrally managed* or *self-managed*. The management type of a target determines how you can specify its Security Hub CSPM settings.

For background information about the benefits of central configuration and how it works, see Understanding central configuration in Security Hub CSPM.

This section explains the differences between a centrally managed and self-managed designation and how to choose the management type of an account, OU, or the root.

Self-managed

The owner of a self-managed account, OU, or root must configure its settings separately in each AWS Region. The delegated administrator can't create configuration policies for self-managed targets.

Centrally managed

Only the delegated Security Hub CSPM administrator can configure settings for centrally managed accounts, OUs, or the root across the home Region and linked Regions. Configuration policies can be associated with centrally managed accounts and OUs.

The delegated administrator can switch the status of a target between self-managed and centrally managed. By default, all accounts and OU are self-managed when you start central configuration through the Security Hub CSPM API. In the console, management type depends on your first configuration policy. Accounts and OUs that you associate with your first policy are centrally managed. Other accounts and OUs are self-managed by default.

If you associate a configuration policy with a previously self-managed account, the policy settings override the self-managed designation. The account becomes centrally managed and adopts the settings reflected in the configuration policy.

If you change a centrally managed account to a self-managed account, the settings that were previously applied to the account through a configuration policy remain in place. For example, a centrally managed account could initially be associated with a policy that enabled Security Hub CSPM, enabled AWS Foundational Security Best Practices, and disabled CloudTrail.1. If you then designate the account as self-managed, all of the settings remain unchanged. However, the account owner can independently change the settings for the account going forward.

Child accounts and OUs can inherit self-managed behavior from a self-managed parent, in the same way that child accounts and OUs can inherit configuration policies from a centrally managed parent. For more information, see Policy association through application and inheritance.

A self-managed account or OU can't inherit a configuration policy from a parent node or from the root. For example, if you want all accounts and OUs in your organization to inherit a configuration policy from the root, you must change the management type of self-managed nodes to centrally managed.

Options to configure settings in self-managed accounts

Self-managed accounts must configure their own settings separately in each Region.

Owners of self-managed accounts can invoke the following operations of the Security Hub CSPM API in each Region to configure their settings:

- `EnableSecurityHub` and `DisableSecurityHub` to enable or disable the Security Hub CSPM service (if a self-managed account has a delegated Security Hub CSPM administrator, the administrator must disassociate the account before the account owner can disable Security Hub CSPM).
- `BatchEnableStandards` and `BatchDisableStandards` to enable or disable standards
- `BatchUpdateStandardsControlAssociations` or `UpdateStandardsControl` to enable or disable controls

Self-managed accounts can also use `*Invitations` and `*Members` operations. However, we recommend that self-managed accounts don't use these operations. Policy associations can fail if a member account has its own members that are part of a different organization than the delegated administrator's.

For descriptions of Security Hub CSPM API actions, see the *AWS Security Hub CSPM API Reference*.

Self-managed accounts can also use the Security Hub CSPM console or AWS CLI to configure their settings in each Region.

Self-managed accounts can't invoke any APIs related to Security Hub CSPM configuration policies and policy associations. Only the delegated administrator can invoke central configuration APIs and use configuration policies to configure centrally managed accounts.

Choosing the management type of a target

Choose your preferred method, and follow the steps to designate an account or OU as centrally managed or self-managed in AWS Security Hub CSPM.

Security Hub CSPM console

To choose the management type of an account or OU

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.
2. Choose **Configuration**.
3. On the **Organization** tab, select the target account or OU. Choose **Edit**.

4. On the **Define configuration** page, for **Management type**, choose **Centrally managed** if you want the delegated administrator to configure the target account or OU. Then, choose **Apply a specific policy** if you want to associate an existing configuration policy with the target. Choose **Inherit from my organization** if you want the target to inherit the configuration of its closest parent. Choose **Self-managed** if you want the account or OU to configure its own settings.
5. Choose **Next**. Review your changes, and choose **Save**.

Security Hub CSPM API

To choose the management type of an account or OU

1. Invoke the StartConfigurationPolicyAssociation API from the Security Hub CSPM delegated administrator account in the home Region.
2. For the `ConfigurationPolicyIdentifier` field, provide `SELF_MANAGED_SECURITY_HUB` if you want the account or OU to control its own settings. Provide the Amazon Resource Name (ARN) or ID of the relevant configuration policy if you want the delegated administrator to control settings for the account or OU.
3. For the `Target` field, provide the AWS account ID, OU ID, or root ID of the target whose management type you want to change. This associates the self-managed behavior or specified configuration policy with the target. Child accounts of the target may inherit the self-managed behavior or configuration policy.

Example API request to designate a self-managed account:

```
{
    "ConfigurationPolicyIdentifier": "SELF_MANAGED_SECURITY_HUB",
    "Target": {"AccountId": "123456789012"}
}
```

AWS CLI

To choose the management type of an account or OU

1. Run the start-configuration-policy-association command from the Security Hub CSPM delegated administrator account in the home Region.

2. For `configuration-policy-identifier` field, provide `SELF_MANAGED_SECURITY_HUB` if you want the account or OU to control its own settings. Provide the Amazon Resource Name (ARN) or ID of the relevant configuration policy if you want the delegated administrator to control settings for the account or OU..
3. For the `target` field, provide the AWS account ID, OU ID, or root ID of the target whose management type you want to change. This associates the self-managed behavior or specified configuration policy with the target. Child accounts of the target may inherit the self-managed behavior or configuration policy.

Example command to designate a self-managed account:

```
aws securityhub --region us-east-1 start-configuration-policy-association \
--configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \
--target '{"AccountId": "123456789012"}'
```

How configuration policies work in Security Hub CSPM

The delegated AWS Security Hub CSPM administrator can create configuration policies to configure Security Hub CSPM, security standards, and security controls for an organization. After creating a configuration policy, the delegated administrator can associate it with specific accounts, organizational units (OUs), or the root. The policy then takes effect in the specified accounts, OUs, or the root.

For background information about the benefits of central configuration and how it works, see Understanding central configuration in Security Hub CSPM.

This section provides a detailed overview of configuration policies.

Policy considerations

Before you create a configuration policy in Security Hub CSPM, consider the following details.

- **Configuration policies must be associated to take effect** – After you create a configuration policy, you can associate it with one or more accounts, organizational units (OUs), or the root. A configuration policy can be associated with accounts or OUs through direct application, or through inheritance from a parent OU.

- **An account or OU can be associated with only one configuration policy** – To prevent conflicting settings, an account or OU can only be associated with one configuration policy at any given time. Alternatively, an account or OU can be self-managed.
- **Configuration policies are complete** – Configuration policies provide a complete specification of settings. For example, a child account can't accept settings for some controls from one policy and settings for other controls from another policy. When you associate a policy with a child account, ensure that the policy specifies all of the settings that you want the child account to use.
- **Configuration policies can't be reverted** – There's no option to revert a configuration policy after you associate it with accounts or OUs. For example, if you associate a configuration policy that disables CloudWatch controls with a specific account, and then dissociate that policy, the CloudWatch controls continue to be disabled in that account. To enable CloudWatch controls again, you can associate the account with a new policy that enables the controls. Alternatively, you can change the account to self-managed and enable each CloudWatch control in the account.
- **Configuration policies take effect in your home Region and all linked Regions** – A configuration policy affects all associated accounts in the home Region and all linked Regions. You can't create a configuration policy that takes effect in only some of these Regions and not others. The exception to this is controls that use global resources. Security Hub CSPM automatically disables controls that involve global resources in all Regions except the home Region.

  Regions that AWS introduced on or after March 20, 2019 are known as opt-in Regions. You must enable such a Region for an account before a configuration policy takes effect there. The Organizations management account can enable opt-in Regions for a member account. For instructions on enabling opt-in Regions, see Specify which AWS Regions your account can use in the *AWS Account Management Reference Guide*.

  If your policy configures a control that isn't available in the home Region or one or more linked Regions, Security Hub CSPM skips the control configuration in unavailable Regions but applies the configuration in Regions where the control is available. You lack coverage for a control that isn't available in the home Region or any of the linked Regions.
- **Configuration policies are resources** – As a resource, a configuration policy has an Amazon Resource Name (ARN) and a universally unique identifier (UUID). The ARN uses the following format: `arn:`*`partition:`*`securityhub:`*`region`*`:`*`delegated administrator account ID`*`:configuration-policy/`*`configuration policy UUID`*. A self-managed

configuration has no ARN or UUID. The identifier for a self-managed configuration is `SELF_MANAGED_SECURITY_HUB`.

Types of configuration policies

Each configuration policy specifies the following settings:

- Enable or disable Security Hub CSPM.
- Enable one or more security standards.
- Indicate which security controls are enabled across enabled standards. You can do this by providing a list of specific controls that should be enabled, and Security Hub CSPM disables all other controls, including new controls when they are released. Alternatively, you can provide a list of specific controls that should be disabled, and Security Hub CSPM enables all other controls, including new controls when they are released.
- Optionally, customize parameters for select enabled controls across enabled standards.

Central configuration policies don't include AWS Config recorder settings. You must separately enable AWS Config and turn on recording for required resources in order for Security Hub CSPM to generate control findings. For more information, see Considerations before enabling and configuring AWS Config.

If you use central configuration, Security Hub CSPM automatically disables controls that involve global resources in all Regions except the home Region. Other controls that you choose to enable though a configuration policy are enabled in all Regions where they are available. To limit findings for these controls to just one Region, you can update your AWS Config recorder settings and turn off global resource recording in all Regions except the home Region.

If an enabled control that involves global resources isn't supported in the home Region, Security Hub CSPM tries to enable the control in one linked Region where the control is supported. With central configuration, you lack coverage for a control that isn't available in the home Region or any of the linked Regions.

For a list of controls that involve global resources, see Controls that use global resources.

Recommended configuration policy

When creating a configuration policy for the *first time in the Security Hub CSPM console*, you have the option to choose the Security Hub CSPM recommended policy.

The recommended policy enables Security Hub CSPM, the AWS Foundational Security Best Practices (FSBP) standard, and all existing and new FSBP controls. Controls that accept parameters use the default values. The recommended policy applies to root (all accounts and OUs, both new and existing). After creating the recommended policy for your organization, you can modify it from the delegated administrator account. For example, you can enable additional standards or controls or disable specific FSBP controls. For instructions on modifying a configuration policy, see Updating configuration policies.

Custom configuration policy

Instead of the recommended policy, the delegated administrator can create up to 20 custom configuration policies. You can associate a single custom policy with your entire organization or different custom policies with different accounts and OUs. For a custom configuration policy, you specify your desired settings. For example, you can create a custom policy that enables FSBP, the Center for Internet Security (CIS) AWS Foundations Benchmark v1.4.0, and all controls in those standards except Amazon Redshift controls. The level of granularity that you use in custom configuration policies depends on the intended scope of security coverage throughout your organization.

> **Note**
>
> You can't associate a configuration policy that disables Security Hub CSPM with the delegated administrator account. Such a policy can be associated with other accounts but skips association with the delegated administrator. The delegated administrator account retains its current configuration.

After creating a custom configuration policy, you can switch to the recommended configuration policy by updating your configuration policy to reflect the recommended configuration. However, you don't see the choice to create the recommended configuration policy in the Security Hub CSPM console after your first policy is created.

Policy association through application and inheritance

When you first opt in to central configuration, your organization has no associations and behaves in the same way that it did prior to opt-in. The delegated administrator can then establish associations between a configuration policy or self-managed behavior and accounts, OUs, or the root. Associations can be established through *application* or *inheritance*.

From the delegated administrator account, you can directly apply a configuration policy to an account, OU, or the root. Alternatively, the delegated administrator can directly apply a self-managed designation to an account, OU, or the root.

In the absence of direct application, an account or OU inherits the settings of the closest parent that has a configuration policy or self-managed behavior. If the closest parent is associated with a configuration policy, the child inherits that policy and is configurable only by the delegated administrator from the home Region. If the closest parent is self-managed, the child inherits the self-managed behavior and has the ability to specify its own settings in each AWS Region.

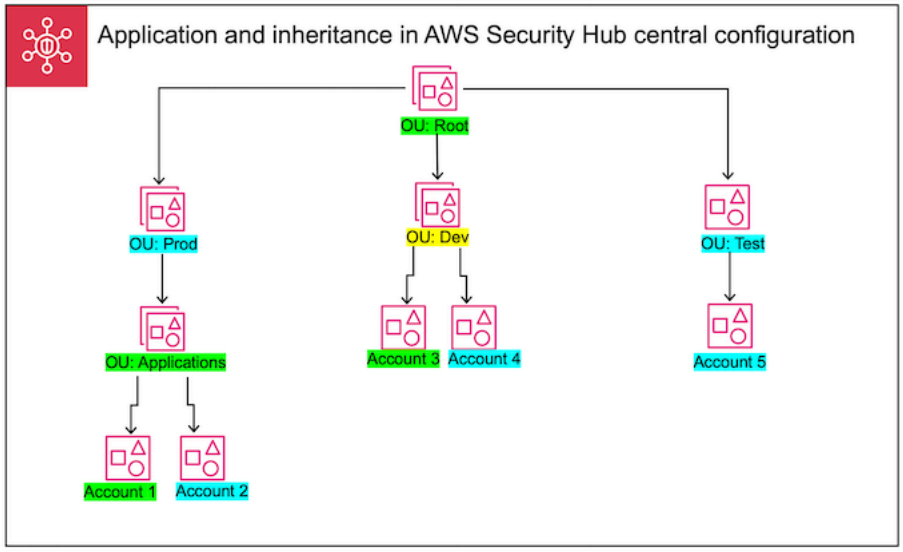
Application takes precedence over inheritance. In other words, inheritance doesn't override a configuration policy or self-managed designation that the delegated administrator has directly applied to an account or OU.

If you directly apply a configuration policy to a self-managed account, the policy overrides the self-managed designation. The account becomes centrally managed and adopts the settings reflected in the configuration policy.

We recommend directly applying a configuration policy to the root. If you apply a policy to the root, then new accounts that join your organization will automatically inherit the root policy unless you associate them with a different policy or designate them as self-managed.

Only one configuration policy can be associated with an account or OU at a given time, either through application or inheritance. This is designed to prevent conflicting settings.

The following diagram illustrates how policy application and inheritance work in central configuration.



In this example, a node highlighted in green has a configuration policy that's been applied to it. A node highlighted in blue has no configuration policy that's been applied to it. A node highlighted in yellow has been designated as self-managed. Each account and OU uses the following configuration:

- **OU:Root (Green)** – This OU uses the configuration policy that's been applied to it.
- **OU:Prod (Blue)** – This OU inherits the configuration policy from OU:Root.
- **OU:Applications (Green)** – This OU uses the configuration policy that's been applied to it.
- **Account 1 (Green)** – This account uses the configuration policy that's been applied to it.
- **Account 2 (Blue)** – This account inherits the configuration policy from OU:Applications.
- **OU:Dev (Yellow)** – This OU is self-managed.
- **Account 3 (Green)** – This account uses the configuration policy that's been applied to it.
- **Account 4 (Blue)** – This account inherits self-managed behavior from OU:Dev.
- **OU:Test (Blue)** – This account inherits the configuration policy from OU:Root.
- **Account 5 (Blue)** – This account inherits the configuration policy from OU:Root since its immediate parent, OU:Test, isn't associated with a configuration policy.

Testing a configuration policy

To make sure you understand how configuration policies work, we recommend creating one policy and associating it with a test account or OU.

To test a configuration policy

1. Create a custom configuration policy, and verify that the specified settings for Security Hub CSPM enablement, standards, and controls are correct. For instructions, see Creating and associating configuration policies.
2. Apply the configuration policy to a test account or OU that doesn't have any child accounts or OUs.
3. Verify that the test account or OU uses the configuration policy in the expected way in your home Region and all linked Regions. You can also verify that all other accounts and OUs in your organization remain self-managed and can change their own settings in each Region.

After you've tested a configuration policy in a single account or OU, you can associate it with other accounts and OUs.

Creating and associating configuration policies

The delegated AWS Security Hub CSPM administrator account can create configuration policies that specify how Security Hub CSPM, standards, and controls are configured in specified accounts and organizational units (OUs). A configuration policy takes effect only after the delegated administrator associates it with at least one account or organizational unit (OUs), or the root. The delegated administrator can also associate a self-managed configuration with accounts, OUs, or the root.

If this is your first time creating a configuration policy, we recommend first reviewing How configuration policies work in Security Hub CSPM.

Choose your preferred access method, and follow the steps to create and associate a configuration policy or self-managed configuration. When using the Security Hub CSPM console, you can associate a configuration with multiple accounts or OUs at the same time. When using the Security Hub CSPM API or AWS CLI, you can associate a configuration with only one account or OU in each request.

> **Note**
>
> If you use central configuration, Security Hub CSPM automatically disables controls that involve global resources in all Regions except the home Region. Other controls that you choose to enable though a configuration policy are enabled in all Regions where they are available. To limit findings for these controls to just one Region, you can update your AWS Config recorder settings and turn off global resource recording in all Regions except the home Region.
>
> If an enabled control that involves global resources isn't supported in the home Region, Security Hub CSPM tries to enable the control in one linked Region where the control is supported. With central configuration, you lack coverage for a control that isn't available in the home Region or any of the linked Regions.
>
> For a list of controls that involve global resources, see Controls that use global resources.

Security Hub CSPM console

To create and associate configuration policies

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.

2. In the navigation pane, choose **Configuration** and the **Policies** tab. Then, choose **Create policy**.
3. On the **Configure organization** page, if this is your first time creating an configuration policy, you see three options under **Configuration type**. If you've already created at least one configuration policy, you only see the **Custom policy** option.

   - Choose **Use the AWS recommended Security Hub CSPM configuration across my entire organization** to use our recommended policy. The recommended policy enables Security Hub CSPM in all organization accounts, enables the AWS Foundational Security Best Practices (FSBP) standard, and enables all new and existing FSBP controls. The controls use default parameter values.
   - Choose **I'm not ready to configure yet** to create a configuration policy later.
   - Choose **Custom policy** to create a custom configuration policy. Specify whether to enable or disable Security Hub CSPM, which standards to enable, and which controls to enable across those standards. Optionally, specify custom parameter values for one or more enabled controls that support custom parameters.
4. In the **Accounts** section, choose which target accounts, OUs, or the root that you want your configuration policy to apply to.

   - Choose **All accounts** if you want to apply the configuration policy to the root. This includes all accounts and OUs in the organization that don't have another policy applied to them or inherited.
   - Choose **Specific accounts** if you want to apply the configuration policy to specific accounts or OUs. Enter the account IDs, or select the accounts and OUs from the organization structure. You can apply the policy to a maximum of 15 targets (accounts, OUs, or root) when you create it. To specify a larger number, edit your policy after creation, and apply it to additional targets.
   - Choose **The delegated administrator only** to apply the configuration policy to the current delegated administrator account.
5. Choose **Next**.
6. On the **Review and apply** page, review your configuration policy details. Then, choose **Create policy and apply**. In your home Region and linked Regions, this action overrides the existing configuration settings of accounts that are associated with this configuration

policy. Accounts may be associated with the configuration policy through application, or inheritance from a parent node. Child accounts and OUs of the applied targets will automatically inherit this configuration policy unless they are specifically excluded, self-managed, or use a different configuration policy.

Security Hub CSPM API

To create and associate configuration policies

1. Invoke the CreateConfigurationPolicy API from the Security Hub CSPM delegated administrator account in the home Region.
2. For `Name`, provide a unique name for the configuration policy. Optionally, for `Description`, provide a description for the configuration policy.
3. For the `ServiceEnabled` field, specify if you want Security Hub CSPM to be enabled or disabled in this configuration policy.
4. For the `EnabledStandardIdentifiers` field, specify which Security Hub CSPM standards you want to enable in this configuration policy.
5. For the `SecurityControlsConfiguration` object, specify which controls you want to enable or disable in this configuration policy. Choosing `EnabledSecurityControlIdentifiers` means that the specified controls are enabled. Other controls that are part of your enabled standards (including newly released controls) are disabled. Choosing `DisabledSecurityControlIdentifiers` means that the specified controls are disabled. Other controls that are part of your enabled standards (including newly released controls) are enabled.
6. Optionally, for the `SecurityControlCustomParameters` field, specify enabled controls for which you want to customize parameters. Provide `CUSTOM` for the `ValueType` field and the custom parameter value for the `Value` field. The value must be the correct data type and within valid ranges specified by Security Hub CSPM. Only select controls support custom parameter values. For more information, see Understanding control parameters in Security Hub CSPM.
7. To apply your configuration policy to accounts or OUs, invoke the StartConfigurationPolicyAssociation API from the Security Hub CSPM delegated administrator account in the home Region.
8. For the `ConfigurationPolicyIdentifier` field, provide the Amazon Resource Name (ARN) or universally unique identifier (UUID) of the policy. The ARN and UUID are returned

by the `CreateConfigurationPolicy` API. For a self-managed configuration, the `ConfigurationPolicyIdentifier` field is equal to `SELF_MANAGED_SECURITY_HUB`.

9. For the `Target` field, provide the OU, account, or the root ID to which you want this configuration policy to apply. You can only provide one target in each API request. Child accounts and OUs of the selected target will automatically inherit this configuration policy unless they are self-managed or use a different configuration policy.

Example API request to create a configuration policy:

```
{
    "Name": "SampleConfigurationPolicy",
    "Description": "Configuration policy for production accounts",
    "ConfigurationPolicy": {
        "SecurityHub": {
             "ServiceEnabled": true,
             "EnabledStandardIdentifiers": [
                    "arn:aws:securityhub:us-east-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
                    "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
                ],
            "SecurityControlsConfiguration": {
                "DisabledSecurityControlIdentifiers": [
                    "CloudTrail.2"
                ],
                "SecurityControlCustomParameters": [
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "CUSTOM",
                                "Value": {
                                    "Integer": 15
                                }
                            }
                        }
                    }
                ]
            }
        }
    }
```

```
}
```

Example API request to associate a configuration policy:

```
{
    "ConfigurationPolicyIdentifier": "arn:aws:securityhub:us-
east-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Target": {"OrganizationalUnitId": "ou-examplerootid111-exampleouid111"}
}
```

AWS CLI

To create and associate configuration policies

1. Run the create-configuration-policy command from the Security Hub CSPM delegated administrator account in the home Region.
2. For `name`, provide a unique name for the configuration policy. Optionally, for `description`, provide a description for the configuration policy.
3. For the `ServiceEnabled` field, specify if you want Security Hub CSPM to be enabled or disabled in this configuration policy.
4. For the `EnabledStandardIdentifiers` field, specify which Security Hub CSPM standards you want to enable in this configuration policy.
5. For the `SecurityControlsConfiguration` field, specify which controls you want to enable or disable in this configuration policy. Choosing `EnabledSecurityControlIdentifiers` means that the specified controls are enabled. Other controls that are part of your enabled standards (including newly released controls) are disabled. Choosing `DisabledSecurityControlIdentifiers` means that the specified controls are disabled. Other controls that apply to your enabled standards (including newly released controls) are enabled.
6. Optionally, for the `SecurityControlCustomParameters` field, specify enabled controls for which you want to customize parameters. Provide `CUSTOM` for the `ValueType` field and the custom parameter value for the `Value` field. The value must be the correct data type and within valid ranges specified by Security Hub CSPM. Only select controls support custom parameter values. For more information, see Understanding control parameters in Security Hub CSPM.

7. To apply your configuration policy to accounts or OUs, run the start-configuration-policy-association command from the Security Hub CSPM delegated administrator account in the home Region.
8. For the `configuration-policy-identifier` field, provide the Amazon Resource Name (ARN) or ID of the configuration policy. This ARN and ID are returned by the `create-configuration-policy` command.
9. For the `target` field, provide the OU, account, or the root ID to which you want this configuration policy to apply. You can only provide one target each time you run the command. Children of the selected target will automatically inherit this configuration policy unless they are self-managed or use a different configuration policy.

Example command to create a configuration policy:

```
aws securityhub --region us-east-1 create-configuration-policy \
--name "SampleConfigurationPolicy" \
--description "Configuration policy for production accounts" \
--configuration-policy '{"SecurityHub": {"ServiceEnabled": true,
 "EnabledStandardIdentifiers": ["arn:aws:securityhub:us-east-1::standards/aws-
foundational-security-best-practices/v/1.0.0","arn:aws:securityhub:::ruleset/
cis-aws-foundations-benchmark/v/1.2.0"],"SecurityControlsConfiguration":
{"DisabledSecurityControlIdentifiers": ["CloudTrail.2"],
 "SecurityControlCustomParameters": [{"SecurityControlId": "ACM.1", "Parameters":
 {"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer": 15}}}}]}}'
```

Example command to associate a configuration policy:

```
aws securityhub --region us-east-1 start-configuration-policy-association \
--configuration-policy-identifier "arn:aws:securityhub:us-
east-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
--target '{"OrganizationalUnitId": "ou-examplerootid111-exampleouid111"}'
```

The `StartConfigurationPolicyAssociation` API returns a field called `AssociationStatus`. This field tells you whether a policy association is pending or in a state of success or failure. It can take up to 24 hours for the status to change from `PENDING` to `SUCCESS` or `FAILURE`. For more information about association status, see Reviewing the association status of a configuration policy.

Reviewing the status and details of configuration policies

The delegated AWS Security Hub CSPM administrator can view configuration policies for an organization and their details. This includes which accounts and organizational units (OUs) a policy is associated with.

For background information about the benefits of central configuration and how it works, see Understanding central configuration in Security Hub CSPM.

Choose your preferred method, and follow the steps to view your configuration policies.

Security Hub CSPM console

To view configuration policies (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.
2. In the navigation pane, choose **Settings** and **Configuration**.
3. Choose the **Policies** tab for an overview of your configuration policies.
4. Select a configuration policy, and choose **View details** to see additional details about it, including which accounts and OUs it's associated with.

Security Hub CSPM API

To view a summary list of all your configuration policies, use the ListConfigurationPolicies operation of the Security Hub CSPM API. If you use the AWS CLI, run the list-configuration-policies command. The delegated Security Hub CSPM administrator account should invoke the operation in the home Region.

```
$ aws securityhub list-configuration-policies \
--max-items 5 \
--starting-token U2FsdGVkX19nUI2zoh+Pou9YyutlYJHWpn9xnG4hqSOhvw3o2JqjI23QDxdf
```

To view details about a specific configuration policy, use the GetConfigurationPolicy operation. If you use the AWS CLI, run the get-configuration-policy. The delegated administrator account

should invoke the operation in the home Region. Provide the Amazon Resource Name (ARN) or ID of the configuration policy whose details you want to see.

```
$ aws securityhub get-configuration-policy \
--identifier "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

To view a summary list of all your configuration policies and their account associations, use the use the ListConfigurationPolicyAssociations operation. If you use the AWS CLI, run the list-configuration-policy-associations command. The delegated administrator account should invoke the operation in the home Region. Optionally, you can provide pagination parameters or filter the results by a specific policy ID, association type, or association status.

```
$ aws securityhub list-configuration-policy-associations \
--filters '{"AssociationType": "APPLIED"}'
```

To view associations for a specific account, use the GetConfigurationPolicyAssociation operation. If you use the AWS CLI, run the get-configuration-policy-association command. The delegated administrator account should invoke the operation in the home Region. For `target`, provide the account number, OU ID, or root ID.

```
$ aws securityhub get-configuration-policy-association \
--target '{"AccountId": "123456789012"}'
```

Reviewing the association status of a configuration policy

The following central configuration API operations return a field called `AssociationStatus`:

- `BatchGetConfigurationPolicyAssociations`
- `GetConfigurationPolicyAssociation`
- `ListConfigurationPolicyAssociations`
- `StartConfigurationPolicyAssociation`

This field is returned both when the underlying configuration is a configuration policy and when it's self-managed behavior.

The value of `AssociationStatus` tells you whether a policy association is pending or in a state of success or failure for a specific account. It can take up to 24 hours for the status to change from `PENDING` to `SUCCESS` or `FAILED`. A status of `SUCCESS` means that all settings specified in the configuration policy are associated with the account. A status of `FAILED` means that one or more settings specified in the configuration policy failed to associate with the account. Despite a `FAILED` status, the account could be partially configured in accordance with the policy. For example, you might try to associate an account with a configuration policy that enables Security Hub CSPM, enables AWS Foundational Security Best Practices, and disables CloudTrail.1. The initial two settings could succeed, but the CloudTrail.1 setting could fail. In this example, the association status is `FAILED` even though some settings were correctly configured.

The association status of a parent OU or the root depends on the status of its children. If the association status of all the children is `SUCCESS`, the association status of the parent is `SUCCESS`. If the association status of one or more children is `FAILED`, the association status of the parent is `FAILED`.

The value of `AssociationStatus` depends on the association status of the policy in all relevant Regions. If the association succeeds in the home Region and all linked Regions, the value of `AssociationStatus` is `SUCCESS`. If the association fails in one or more of these Regions, the value of `AssociationStatus` is `FAILED`.

The following behavior also impacts the value of `AssociationStatus`:

- If the target is a parent OU or the root, it has an `AssociationStatus` of `SUCCESS` or `FAILED` only when all of the children have a `SUCCESS` or `FAILED` status. If the association status of a child account or OU changes (for example, when a linked Region is added or removed) after you first associate the parent with a configuration, the change doesn't update the association status of the parent unless you invoke the `StartConfigurationPolicyAssociation` API again.
- If the target is an account, it has an `AssociationStatus` of `SUCCESS` or `FAILED` only if the association has a result of `SUCCESS` or `FAILED` in the home Region and all linked Regions. If the association status of a target account changes (for example, when a linked Region is added or removed) after you first associate it with a configuration, its association status is updated. However, the change doesn't update the association status of the parent unless you invoke the `StartConfigurationPolicyAssociation` API again.

If you add a new linked Region, Security Hub CSPM replicates your existing associations that are in a `PENDING`, `SUCCESS`, or `FAILED` state in the new Region.

Even when the association status is SUCCESS, the enablement status of a standard that is part of the policy can transition into an incomplete state. In that case, Security Hub CSPM can't generate findings for the standard's controls. For more information, see Checking the status of a standard.

Troubleshooting association failure

In AWS Security Hub CSPM, a configuration policy association might fail for the following common reasons.

- **Organizations management account isn't a member** – If you want to associate a configuration policy with the Organizations management account, that account must already have AWS Security Hub CSPM enabled. This makes the management account a member account in the organization.
- **AWS Config isn't enabled or properly configured** – To enable standards in a configuration policy, AWS Config must be enabled and configured to record relevant resources.
- **Must associate from delegated administrator account** – You can only associate a policy with target accounts and OUs when you're signed in to the delegated Security Hub CSPM administrator account.
- **Must associate from home Region** – You can only associate a policy with target accounts and OUs when you're signed in to your home Region.
- **Opt-in Region not enabled** – Policy association fails for a member account or OU in a linked Region if it's an opt-in Region that the delegated administrator hasn't enabled. You can retry after enabling the Region from the delegated administrator account.
- **Member account suspended** – Policy association fails if you try to associate a policy with a suspended member account.

Updating configuration policies

After creating a configuration policy, the delegated AWS Security Hub CSPM administrator account can update the policy details and policy associations. When policy details are updated, accounts that are associated with the configuration policy automatically start using the updated policy.

For background information about the benefits of central configuration and how it works, see Understanding central configuration in Security Hub CSPM.

The delegated administrator can update the following policy settings:

- Enable or disable Security Hub CSPM.

- Enable one or more security standards.
- Indicate which security controls are enabled across enabled standards. You can do this by providing a list of specific controls that should be enabled, and Security Hub CSPM disables all other controls, including new controls when they are released. Alternatively, you can provide a list of specific controls that should be disabled, and Security Hub CSPM enables all other controls, including new controls when they are released.
- Optionally, customize parameters for select enabled controls across enabled standards.

Choose your preferred method, and follow the steps to update a configuration policy.

> **Note**
>
> If you use central configuration, Security Hub CSPM automatically disables controls that involve global resources in all Regions except the home Region. Other controls that you choose to enable though a configuration policy are enabled in all Regions where they are available. To limit findings for these controls to just one Region, you can update your AWS Config recorder settings and turn off global resource recording in all Regions except the home Region.
> If an enabled control that involves global resources isn't supported in the home Region, Security Hub CSPM tries to enable the control in one linked Region where the control is supported. With central configuration, you lack coverage for a control that isn't available in the home Region or any of the linked Regions.
> For a list of controls that involve global resources, see Controls that use global resources.

Console

To update configuration policies

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.
2. In the navigation pane, choose **Settings** and **Configuration**.
3. Choose the **Policies** tab.

4. Select the configuration policy that you want to edit, and choose **Edit**. If desired, edit the policy settings. Leave this section as is if you want to keep the policy settings unchanged.
5. Choose **Next**.If desired, edit the policy associations. Leave this section as is if you want to keep the policy associations unchanged. You can associate or disassociate the policy with a maximum of 15 targets (accounts, OUs, or root) when you update it.
6. Choose **Next**.
7. Review your changes, and choose **Save and apply**. In your home Region and linked Regions, this action overrides the existing configuration settings of accounts that are associated with this configuration policy. Accounts may be associated with a configuration policy through application, or inheritance from a parent node.

API

To update configuration policies

1. To update the settings in a configuration policy, invoke the UpdateConfigurationPolicy API from the Security Hub CSPM delegated administrator account in the home Region.
2. Provide the Amazon Resource Name (ARN) or ID of the configuration policy that you want to update.
3. Provide updated values for the fields under `ConfigurationPolicy`. Optionally, you can also provide a reason for the update.
4. To add new associations for this configuration policy, invoke the StartConfigurationPolicyAssociation API from the Security Hub CSPM delegated administrator account in the home Region. To remove one or more current associations, invoke the StartConfigurationPolicyDisassociation API from the Security Hub CSPM delegated administrator account in the home Region.
5. For the `ConfigurationPolicyIdentifier` field, provide the ARN or ID of the configuration policy whose associations you want to update.
6. For the `Target` field, provide the accounts, OUs, or root ID that you want to associate or disassociate. This action overrides previous policy associations for the specified OUs or accounts.

> **Note**
>
> When you invoke the `UpdateConfigurationPolicy` API, Security Hub CSPM performs a full list replacement for the `EnabledStandardIdentifiers`, `EnabledSecurityControlIdentifiers`, `DisabledSecurityControlIdentifiers`, and `SecurityControlCustomParameters` fields. Each time you invoke this API, provide the full list of standards that you want to enable and the full list of controls that you want to enable or disable and customize parameters for.

Example API request to update a configuration policy:

```
{
    "Identifier": "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Description": "Updated configuration policy",
    "UpdatedReason": "Disabling CloudWatch.1",
    "ConfigurationPolicy": {
        "SecurityHub": {
             "ServiceEnabled": true,
             "EnabledStandardIdentifiers": [
                    "arn:aws:securityhub:us-east-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
                    "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
                ],
            "SecurityControlsConfiguration": {
                "DisabledSecurityControlIdentifiers": [
                    "CloudTrail.2",
                    "CloudWatch.1"
                ],
                "SecurityControlCustomParameters": [
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "CUSTOM",
                                "Value": {
                                    "Integer": 15
                                }
```

```
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

AWS CLI

To update configuration policies

1. To update the settings in a configuration policy, run the update-configuration-policy command from the Security Hub CSPM delegated administrator account in the home Region.
2. Provide the Amazon Resource Name (ARN) or ID of the configuration policy that you want to update.
3. Provide updated values for the fields under `configuration-policy`. Optionally, you can also provide a reason for the update.
4. To add new associations for this configuration policy, run the start-configuration-policy-association command from the Security Hub CSPM delegated administrator account in the home Region. To remove one or more current associations, run the start-configuration-policy-disassociation command from the Security Hub CSPM delegated administrator account in the home Region.
5. For the `configuration-policy-identifier` field, provide the ARN or ID of the configuration policy whose associations you want to update.
6. For the `target` field, provide the accounts, OUs, or root ID that you want to associate or disassociate. This action overrides previous policy associations for the specified OUs or accounts.

Note

When you run the `update-configuration-policy` command, Security Hub CSPM performs a full list replacement for the `EnabledStandardIdentifiers`, `EnabledSecurityControlIdentifiers`,

`DisabledSecurityControlIdentifiers`, and `SecurityControlCustomParameters` fields. Each time you run this command, provide the full list of standards that you want to enable and the full list of controls that you want to enable or disable and customize parameters for.

Example command to update a configuration policy:

```
aws securityhub update-configuration-policy \
--region us-east-1 \
--identifier "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
--description "Updated configuration policy" \
--updated-reason "Disabling CloudWatch.1" \
--configuration-policy '{"SecurityHub": {"ServiceEnabled": true,
 "EnabledStandardIdentifiers": ["arn:aws:securityhub:us-east-1::standards/aws-
foundational-security-best-practices/v/1.0.0","arn:aws:securityhub:::ruleset/
cis-aws-foundations-benchmark/v/1.2.0"],"SecurityControlsConfiguration":
{"DisabledSecurityControlIdentifiers": ["CloudTrail.2","CloudWatch.1"],
 "SecurityControlCustomParameters": [{"SecurityControlId": "ACM.1", "Parameters":
 {"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer": 15}}}}]}}}'
```

The `StartConfigurationPolicyAssociation` API returns a field called `AssociationStatus`. This field tells you whether a policy association is pending or in a state of success or failure. It can take up to 24 hours for the status to change from `PENDING` to `SUCCESS` or `FAILURE`. For more information about association status, see Reviewing the association status of a configuration policy.

Deleting configuration policies

After creating a configuration policy, the delegated AWS Security Hub CSPM administrator can delete it. Alternatively, the delegated administrator can retain the policy, but disassociate it from specific accounts or organizational units (OUs), or from the root. For instructions on disassociating a policy, see Disassociating a configuration from its targets.

For background information about the benefits of central configuration and how it works, see Understanding central configuration in Security Hub CSPM.

This section explains how to delete configuration policies.

When you delete a configuration policy, it no longer exists for your organization. Target accounts, OUs, and the organization root can no longer use the configuration policy. Targets that were associated with a deleted configuration policy inherit the configuration policy of the closest parent, or become self-managed if the closest parent is self-managed. If you want a target to use a different configuration, you can associate the target with a new configuration policy. For more information, see Creating and associating configuration policies.

We recommend creating and associating at least one configuration policy with your organization to provide adequate security coverage.

Before you can delete a configuration policy, you must disassociate the policy from any accounts, OUs, or the root to which it currently applies.

Choose your preferred method, and follow the steps to delete a configuration policy.

Console

To delete a configuration policy

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.
2. In the navigation pane, choose **Settings** and **Configuration**.
3. Choose the **Policies** tab. Select the configuration policy that you want to delete, and choose **Delete**. If the configuration policy is still associated with any accounts or OUs, you're prompted to first disassociate the policy from those targets before you can delete it.
4. Review the confirmation message. Enter **`confirm`**, and choose **Delete**.

API

To delete a configuration policy

Invoke the DeleteConfigurationPolicy API from the Security Hub CSPM delegated administrator account in the home Region.

Provide the Amazon Resource Name (ARN) or ID of the configuration policy that you want to delete. If you receive a `ConflictException` error, the configuration policy still applies to

accounts or OUs in your organization. To resolve the error, disassociate the configuration policy from these accounts or OUs before trying to delete it.

Example API request to delete a configuration policy:

```
{
    "Identifier": "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

AWS CLI

To delete a configuration policy

Run the delete-configuration-policy command from the Security Hub CSPM delegated administrator account in the home Region.

Provide the Amazon Resource Name (ARN) or ID of the configuration policy that you want to delete. If you receive a `ConflictException` error, the configuration policy still applies to accounts or OUs in your organization. To resolve the error, disassociate the configuration policy from these accounts or OUs before trying to delete it.

```
aws securityhub --region us-east-1 delete-configuration-policy \
--identifier "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Disassociating a configuration from its targets

From the delegated AWS Security Hub CSPM administrator account, you can disassociate a configuration policy or self-managed configuration from an account, OU, or root. Disassociation retains the policy for future use, but removes existing associations from specific accounts, OUs, or the root.You can disassociate only a directly applied configuration, not an inherited configuration. To change an inherited configuration, you can apply a configuration policy or self-managed behavior to the affected account or OU. You can also apply a new configuration policy, which includes your desired modifications, to the closest parent.

Disassociation *doesn't* delete a configuration policy. The policy is retained in your account, so you can associate it with other targets in your organization. For instructions on deleting a configuration

policy, see Deleting configuration policies. When disassociation is complete, an affected target inherits the configuration policy or self-managed behavior of the closest parent. If there's no inheritable configuration, a target retains the settings it had prior to disassociation but becomes self-managed.

Choose your preferred method, and follow the steps to disassociate an account, OU, or root from its current configuration.

Console

To disassociate an account or OU from its current configuration

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.
2. In the navigation pane, choose **Settings** and **Configuration**.
3. On the **Organizations** tab, select the account, OU, or the root that you want to disassociate from its current configuration. Choose **Edit**.
4. On the **Define configuration** page, for **Management**, choose **Policy applied** if you want the delegated administrator to be able to apply policies directly to the target. Choose **Inherited** if you want the target to inherit the configuration of its closest parent. In either of these cases, the delegated administrator controls settings for the target. Choose **Self-managed** if you want the account or OU to control its own settings.
5. After reviewing your changes, choose **Next** and **Apply**. This action overrides existing configurations of any accounts or OUs that are in scope, if those configurations conflict with your current selections.

API

To disassociate an account or OU from its current configuration

1. Invoke the StartConfigurationPolicyDisassociation API from the Security Hub CSPM delegated administrator account in the home Region.
2. For `ConfigurationPolicyIdentifier`, provide the Amazon Resource Name (ARN) or ID of the configuration policy that you want to disassociate. Provide `SELF_MANAGED_SECURITY_HUB` for this field to disassociate self-managed behavior.

3. For `Target`, provide the accounts, OUs, or the root that you want to dissociate from this configuration policy.

Example API request to disassociate a configuration policy:

```
{
    "ConfigurationPolicyIdentifier": "arn:aws:securityhub:us-
east-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Target": {"RootId": "r-f6g7h8i9j0example"}
}
```

AWS CLI

To disassociate an account or OU from its current configuration

1. Run the start-configuration-policy-disassociation command from the Security Hub CSPM delegated administrator account in the home Region.
2. For `configuration-policy-identifier`, provide the Amazon Resource Name (ARN) or ID of the configuration policy that you want to disassociate. Provide `SELF_MANAGED_SECURITY_HUB` for this field to disassociate self-managed behavior.
3. For `target`, provide the accounts, OUs, or the root that you want to dissociate from this configuration policy.

Example command to disassociate a configuration policy:

```
aws securityhub --region us-east-1 start-configuration-policy-disassociation \
--configuration-policy-identifier "arn:aws:securityhub:us-
east-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
--target '{"RootId": "r-f6g7h8i9j0example"}'
```

Configuring a standard or control in context

When you use central configuration in AWS Security Hub CSPM, the delegated Security Hub CSPM administrator can create configuration policies that specify how Security Hub CSPM, security standards, and security controls are configured for an organization. The delegated administrator

can associate policies with specific accounts and organizational units (OU). The policies take effect in your home Region and all linked Regions. The delegated administrator can update configuration policies as necessary.

On the Security Hub CSPM console, the delegated administrator can update configuration policies in two ways—from the **Configuration** page, or in context with existing workflows. The latter can be beneficial because, as you view security findings, you can discover which standards and controls are most relevant to your environment and configure them at the same time.

In-context configuration is available only on the Security Hub CSPM console. Programmatically, the delegated administrator must invoke the UpdateConfigurationPolicy operation of the Security Hub CSPM API to change how specific standards or controls are configured in the organization.

Follow these steps to configure a Security Hub CSPM standard or control in context.

To configure a standard or control in context (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.
2. In the navigation pane, choose one of the follow options:

   - To configure a standard, choose **Security standards**, and choose a specific standard.
   - To configure a control, choose **Controls**, and choose a specific control.
3. The console lists your existing Security Hub CSPM configuration policies and the status of the selected standard or control in each one. Choose the options to enable or disable the standard or control in each existing configuration policy. For controls, you can also choose to customize control parameters. You can't create a new policy during in-context configuration. To create a new policy, you must go to the **Configuration** page, choose the **Policies** tab, and then choose **Create policy**.
4. After making your changes, choose **Next**.
5. Review your changes, and choose **Apply**. The updates affect all accounts and OUs that are associated with a changed configuration policy. The updates also take effect in the home Region and all linked Regions.

Disabling central configuration in Security Hub CSPM

When you disable central configuration in AWS Security Hub CSPM, the delegated administrator loses the ability to configure Security Hub CSPM, security standards, and security controls across multiple AWS accounts, organizational units (OUs), and AWS Regions. Instead, you must configure most settings separately for each account in each Region.

> **Important**
>
> Before you can disable central configuration, you must first disassociate your accounts and OUs from their current configuration, whether that's a configuration policy or self-managed behavior.
>
> Before you can disable central configuration, you must also delete existing configuration policies.

When you disable central configuration, the following changes occur:

- The delegated administrator can no longer create configuration policies for the organization.
- Accounts that had an applied or inherited configuration policy retain their current settings, but become self-managed.
- Your organization switches to *local configuration*. Under local configuration, the majority of Security Hub CSPM settings must be configured separately in each organization account and Region. The delegated administrator can choose to automatically enable Security Hub CSPM, default security standards, and all controls that are part of the default standards in new organization accounts. The default standards are AWS Foundational Security Best Practices (FSBP) and Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0. These settings take effect in the current Region only and impact new organization accounts only. The delegated administrator can't change which standards are default. Local configuration doesn't support the use of configuration policies or configuration at the OU level.

The identity of the delegated administrator account remains the same when you stop using central configuration. Your home Region and linked Regions also remain the same (your home Region is now called the aggregation Region, and can be used for finding aggregation).

Choose your preferred method, and follow the steps to stop using central configuration and switch to local configuration.

Security Hub CSPM console

To disable central configuration (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.
2. On the navigation pane, choose **Settings** and **Configuration**.
3. In the **Overview** section, choose **Edit**.
4. In the **Edit organization configuration** box, choose **Local configuration**. If you haven't already, you're prompted to disassociate and delete your current configuration policies before you can stop central configuration. Accounts or OUs that are designated as self-managed must be disassociated from their self-managed configuration. You can do this in the console by changing the management type of each self-managed account or OU to **Centrally managed** and **Inherit from my organization**.
5. Optionally, select the local configuration default settings for new organization accounts.
6. Choose **Confirm**.

Security Hub CSPM API

To disable central configuration (API)

1. Invoke the UpdateOrganizationConfiguration API.
2. Set the `ConfigurationType` field in the `OrganizationConfiguration` object to `LOCAL`. The API returns an error if you have existing configuration policies or policy associations. To disassociate a configuration policy, invoke the `StartConfigurationPolicyDisassociation` API. To delete a configuration policy, invoke the `DeleteConfigurationPolicy` API.
3. If you want to automatically enable Security Hub CSPM in new organization accounts, set the `AutoEnable` field to `true`. By default, the value of this field is `false`, and Security Hub CSPM isn't automatically enabled in new organization accounts. Optionally, if you want to automatically enable default security standards in new organization accounts, set the `AutoEnableStandards` field to `DEFAULT`. This the default value. If you don't want

to automatically enable default security standards in new organization accounts, set the `AutoEnableStandards` field to `NONE`.

Example API request:

```
{
    "AutoEnable": true,
    "OrganizationConfiguration": {
        "ConfigurationType" : "LOCAL"
    }
}
```

AWS CLI

To disable central configuration (AWS CLI)

1. Run the update-organization-configuration command.
2. Set the `ConfigurationType` field in the `organization-configuration` object to `LOCAL`. The command returns an error if you have existing configuration policies or policy associations. To disassociate a configuration policy, run the `start-configuration-policy-disassociation` command. To delete a configuration policy, run the `delete-configuration-policy` command.
3. If you want to automatically enable Security Hub CSPM in new organization accounts, include the `auto-enable` parameter. By default, the value of this parameter is `no-auto-enable`, and Security Hub CSPM isn't automatically enabled in new organization accounts. Optionally, if you want to automatically enable default security standards in new organization accounts, set the `auto-enable-standards` field to `DEFAULT`. This the default value. If you don't want to automatically enable default security standards in new organization accounts, set the `auto-enable-standards` field to `NONE`.

```
aws securityhub --region us-east-1 update-organization-configuration \
--auto-enable \
--organization-configuration '{"ConfigurationType": "LOCAL"}'
```

Managing administrator and member accounts in Security Hub CSPM

If your AWS environment has multiple accounts, you can treat the accounts that use AWS Security Hub CSPM as member accounts and associate them with a single administrator account. The administrator can monitor your overall security posture and take allowed actions on member accounts. The administrator can also perform various account management and administration tasks at scale, such as monitoring estimated usage costs and assessing account quotas.

You can associate member accounts with an administrator in two ways, by integrating Security Hub CSPM with AWS Organizations or by manually sending and accepting membership invitations in Security Hub CSPM.

Managing accounts with AWS Organizations

AWS Organizations is a global account management service that lets AWS administrators to consolidate and manage multiple AWS accounts. It provides account management and consolidated billing features that are designed to support budgetary, security, and compliance needs. It's offered at no additional charge, and it integrates with multiple AWS services, including AWS Security Hub CSPM, Amazon Macie, and Amazon GuardDuty. For more information, see the *AWS Organizations User Guide*.

When you integrate Security Hub CSPM and AWS Organizations, the Organizations management account designates a Security Hub CSPM delegated administrator. Security Hub CSPM is automatically enabled in the delegated administrator account in the AWS Region in which it was designated.

After designating a delegated administrator, we recommend managing accounts in Security Hub CSPM with central configuration. This is the most efficient way to customize Security Hub CSPM and ensure adequate security coverage for your organization.

Central configuration lets the delegated administrator customize Security Hub CSPM across multiple organization accounts and Regions rather than configuring Region-by-Region. You can create a configuration policy for your entire organization, or create different configuration policies for different accounts and OUs. The policies specify whether Security Hub CSPM is enabled or disabled in associated accounts and which security standards and controls are enabled.

The delegated administrator can designate accounts as centrally managed or self-managed. Centrally managed accounts are configurable only by the delegated administrator. Self-managed accounts can specify their own settings.

If you don't opt in to central configuration, the delegated administrator has a more limited ability to configure Security Hub CSPM, called *local configuration*. Under local configuration, the delegated administrator can automatically enable Security Hub CSPM and default security standards in new organization accounts in the current Region. However, existing accounts don't use these settings, so configuration drift can occur after an account joins the organization.

Aside from these new account settings, local configuration is account-specific and Region-specific. Each organization account must configure the Security Hub CSPM service, standards, and controls separately in each Region. Local configuration also doesn't support the use of configuration policies.

Managing accounts manually by invitation

You must manually manage member accounts by invitation in Security Hub CSPM if you have a standalone account or if you don't integrate with Organizations. A standalone account can't integrate with Organizations, so it's necessary to manage it manually. We recommend integrating with AWS Organizations and using central configuration if you add additional accounts in the future.

When you use manual account management, you designate an account to be the Security Hub CSPM administrator. The administrator account can view data in member accounts and take certain actions on member account findings. The Security Hub CSPM administrator invites other accounts to be member accounts, and the administrator-member relationship is established when a prospective member account accepts the invitation.

Manual account management doesn't support the use of configuration policies. Without configuration policies, the administrator can't centrally customize Security Hub CSPM by configuring variable settings for different accounts. Instead, each organization account must enable and configure Security Hub CSPM for itself separately in each Region. This can make it more difficult and time consuming to ensure adequate security coverage across all of the accounts and Regions in which you use Security Hub CSPM. It can also cause configuration drift as member accounts can specify their own settings without input from the administrator.

To manage accounts by invitation, see Managing accounts by invitation in Security Hub CSPM.

Recommendations for managing multiple accounts in Security Hub CSPM

The following section summarizes some restrictions and recommendations to keep in mind when managing member accounts in AWS Security Hub CSPM.

Maximum number of member accounts

If you use the integration with AWS Organizations, Security Hub CSPM supports up to 10,000 member accounts per delegated administrator account in each AWS Region. If you enable and manage Security Hub CSPM manually, Security Hub CSPM supports up to 1,000 member account invitations per administrator account in each Region.

Creating administrator-member relationships

> **Note**
>
> If you use the Security Hub CSPM integration with AWS Organizations, and haven't manually invited any member accounts, this section doesn't apply to you.

An account can't be an administrator account and a member account at the same time.

A member account can only be associated with one administrator account. If an organization account is enabled by the Security Hub CSPM administrator account, the account cannot accept an invitation from another account. If an account has already accepted an invitation, the account cannot be enabled by the Security Hub CSPM administrator account for the organization. It also cannot receive invitations from other accounts.

For the manual invitation process, accepting a membership invitation is optional.

Membership through AWS Organizations

If you integrate Security Hub CSPM with AWS Organizations, the Organizations management account can designate a delegated administrator (DA) account for Security Hub CSPM. The organization management account can't be set as the DA in Organizations. While this is permitted in Security Hub CSPM, we recommend that the Organizations management account should *not* be the DA.

We recommend that you choose the same DA account in all Regions. If you use central configuration, then Security Hub CSPM sets the same DA account in all Regions in which you configure Security Hub CSPM for your organization.

We also recommend that you choose the same DA account across AWS security and compliance services to help you manage security-related issues in a single pane of glass.

Membership by invitation

For member accounts created by invitation, the administrator-member account association is created only in the Region that the invitation is sent from. The administrator account must enable Security Hub CSPM in each Region that you want to use it in. The administrator account then invites each account to become a member account in that Region.

> **Note**
>
> We recommend using AWS Organizations instead of Security Hub CSPM invitations to manage your member accounts.

Coordinating administrator accounts across services

Security Hub CSPM aggregates findings from various AWS services, such as Amazon GuardDuty, Amazon Inspector, and Amazon Macie. Security Hub CSPM also allows users to pivot from a GuardDuty finding to start an investigation in Amazon Detective.

However, the administrator-member relationships that you set up in these other services do not automatically apply to Security Hub CSPM. Security Hub CSPM recommends that you use the same account as the administrator account for all of these services. This administrator account should be an account that is responsible for security tools. The same account should also be the aggregator account for AWS Config.

For example, a user from the GuardDuty administrator account A can see findings for GuardDuty member accounts B and C on the GuardDuty console. If account A then enables Security Hub CSPM, users from account A do *not* automatically see GuardDuty findings for accounts B and C in Security Hub CSPM. A Security Hub CSPM administrator-member relationship is also required for these accounts.

To do this, make account A the Security Hub CSPM administrator account and enable accounts B and C to become Security Hub CSPM member accounts.

Managing Security Hub CSPM for multiple accounts with AWS Organizations

You can integrate AWS Security Hub CSPM with AWS Organizations, and then manage Security Hub CSPM for accounts in your organization.

To integrate Security Hub CSPM with AWS Organizations, you create an organization in AWS Organizations. The Organizations management account designates one account as the Security Hub CSPM delegated administrator for the organization. The delegated administrator can then enable Security Hub CSPM for other accounts in the organization, add those accounts as Security Hub CSPM member accounts, and take allowed actions on the member accounts. The Security Hub CSPM delegated administrator can enable and manage Security Hub CSPM for up to 10,000 member accounts.

The extent of the delegated administrator's configuration abilities depend on whether you use central configuration. With central configuration enabled, you don't need to configure Security Hub CSPM separately in each member account and AWS Region. The delegated administrator can enforce specific Security Hub CSPM settings in specified member accounts and organizational units (OUs) across Regions.

The Security Hub CSPM delegated administrator account can perform the following actions on member accounts:

- If using central configuration, centrally configure Security Hub CSPM for member accounts and OUs by creating Security Hub CSPM configuration policies. Configuration policies can be used to enable and disable Security Hub CSPM, enable and disable standards, and enable and disable controls.
- Automatically treat *new* accounts as Security Hub CSPM member accounts when they join the organization. If you use central configuration, a configuration policy that is associated with an OU includes existing and new accounts that are part of the OU.
- Treat *existing* organization accounts as Security Hub CSPM member accounts. This happens automatically if you use central configuration.
- Disassociate member accounts that belong to the organization. If you use central configuration, you can disassociate a member account only after designating it as self-managed. Alternatively, you can associate a configuration policy that disables Security Hub CSPM with specific centrally managed member accounts.

If you don't opt in to central configuration, your organization uses the default configuration type called local configuration. Under local configuration, the delegated administrator has a more limited ability to enforce settings in member accounts. For more information, see Understanding local configuration in Security Hub CSPM.

For a full list of actions that the delegated administrator can perform on member accounts, see Allowed actions by administrator and member accounts in Security Hub CSPM.

The topics in this section explain how to integrate Security Hub CSPM with AWS Organizations and how to manage Security Hub CSPM for accounts in an organization. Where relevant, each section identifies management benefits and differences for users of central configuration.

Topics

Integrating Security Hub CSPM with AWS Organizations

To integrate AWS Security Hub CSPM and AWS Organizations, you create an organization in Organizations and use the organization management account to designate a delegated Security Hub CSPM administrator account. This enables Security Hub CSPM as a trusted service in Organizations. It also enables Security Hub CSPM in the current AWS Region for the delegated administrator account, and it allows the delegated administrator to enable Security Hub CSPM for member accounts, view data in member accounts, and perform other allowed actions on member accounts.

If you use central configuration, then the delegated administrator can also create Security Hub CSPM configuration policies that specify how the Security Hub CSPM service, standards, and controls should be configured in organization accounts.

Creating an organization

An organization is an entity that you create to consolidate your AWS accounts so that you can administer them as a single unit.

You can create an organization by using either the AWS Organizations console or by using a command from the AWS CLI or one of the SDK APIs. For detailed instructions, see Create an organization in the *AWS Organizations User Guide*.

You can use AWS Organizations to centrally view and manage all of the accounts within your organization. An organization has one management account along with zero or more member accounts. You can organize the accounts in a hierarchical, tree-like structure with a root at the top and organizational units (OUs) nested under the root. Each account can be directly under the root, or placed in one of the OUs in the hierarchy. An OU is a container for specific accounts. For example, you can create a finance OU that includes all accounts related to financial operations.

Recommendations for choosing the delegated Security Hub CSPM administrator

If you have an administrator account in place from the manual invitation process and are transitioning to account management with AWS Organizations, we recommend designating that account as the delegated Security Hub CSPM administrator.

Although the Security Hub CSPM APIs and console allow the organization management account to be the delegated Security Hub CSPM administrator, we recommend choosing two different accounts. This is because users who have access to the organization management account to manage billing are likely to be different from users who need access to Security Hub CSPM for security management.

We recommend using the same delegated administrator across Regions. If you opt in to central configuration, Security Hub CSPM automatically designates the same delegated administrator in your home Region and any linked Regions.

Verify permissions to configure the delegated administrator

To designate and remove a delegated Security Hub CSPM administrator account, the organization management account must have permissions for the `EnableOrganizationAdminAccount` and `DisableOrganizationAdminAccount` actions in Security Hub CSPM. The Organizations management account must also have administrative permissions for Organizations.

To grant all of the required permissions, attach the following Security Hub CSPM managed policies to the IAM principal for the organization management account:

- AWSSecurityHubFullAccess
- AWSSecurityHubOrganizationsAccess

Designating the delegated administrator

To designate the delegated Security Hub CSPM administrator account, you can use the Security Hub CSPM console, Security Hub CSPM API, or AWS CLI. Security Hub CSPM sets the delegated administrator in the current AWS Region only, and you must repeat the action in other Regions. If you start using central configuration, then Security Hub CSPM automatically sets the same delegated administrator in the home Region and linked Regions.

The organization management account doesn't have to enable Security Hub CSPM in order to designate the delegated Security Hub CSPM administrator account.

We recommend that the organization management account is not the delegated Security Hub CSPM administrator account. However, if you do choose the organization management account as the Security Hub CSPM delegated administrator, the management account must have Security Hub CSPM enabled. If the management account does not have Security Hub CSPM enabled, you must enable Security Hub CSPM for it manually. Security Hub CSPM can't be enabled automatically for the organization management account.

You must designate the delegated Security Hub CSPM administrator using one of the following methods. Designating the delegated Security Hub CSPM administrator with Organizations APIs doesn't reflect in Security Hub CSPM.

Choose your preferred method, and follow the steps to designate the delegated Security Hub CSPM administrator account.

Security Hub CSPM console

To designate the delegated administrator while onboarding

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Choose **Go to Security Hub CSPM**. You're prompted to sign in to the organization management account.
3. On the **Designate delegated administrator** page, in the **Delegated administrator account** section, specify the delegated administrator account. We recommend choosing the same delegated administrator that you have set for other AWS security and compliance services.
4. Choose **Set delegated administrator**. You're prompted to sign in to the delegated administrator account (if you're not already) to continue onboarding with central

configuration. If you don't want to start central configuration, choose **Cancel**. Your delegated administrator is set, but you aren't yet using central configuration.

To designate the delegated administrator from the Settings page

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the Security Hub CSPM navigation pane, choose **Settings**. Then choose **General**.
3. If a Security Hub CSPM administrator account is currently assigned, then before you can designate a new account, you must remove the current account.

   Under **Delegated Administrator**, to remove the current account, choose **Remove**.
4. Enter the account ID of the account you want to designate as the **Security Hub CSPM** administrator account.

   You must designate the same Security Hub CSPM administrator account in all Regions. If you designate an account that is different from the account designated in other Regions, the console returns an error.
5. Choose **Delegate**.

Security Hub CSPM API, AWS CLI

From the organization management account, use the EnableOrganizationAdminAccount operation of the Security Hub CSPM API. If you're using the AWS CLI, run the enable-organization-admin-account command. Provide the AWS account ID of the delegated Security Hub CSPM administrator.

The following example designates the delegated Security Hub CSPM administrator. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub enable-organization-admin-account --admin-account-id 123456789012
```

Removing or changing the delegated administrator

Only the organization management account can remove the delegated Security Hub CSPM administrator account.

To change the delegated Security Hub CSPM administrator, you must first remove the current delegated administrator account and then designate a new one.

> **Warning**
>
> When you use central configuration, you can't use the Security Hub CSPM console or Security Hub CSPM APIs to change or remove the delegated administrator account. If the organization management account uses the AWS Organizations console or AWS Organizations APIs to change or remove the delegated Security Hub CSPM administrator, Security Hub CSPM automatically stops central configuration, and deletes your configuration policies and policy associations. Member accounts retain the configurations they had before the delegated administrator was changed or removed.

If you use the Security Hub CSPM console to remove the delegated administrator in one Region, it is automatically removed in all Regions.

The Security Hub CSPM API only removes the delegated Security Hub CSPM administrator account from the Region where the API call or command is issued. You must repeat the action in other Regions.

If you use the Organizations API to remove the delegated Security Hub CSPM administrator account, it is automatically removed in all Regions.

Removing the delegated administrator (Organizations API, AWS CLI)

You can use Organizations to remove the delegated Security Hub CSPM administrator in all Regions.

If you use central configuration to manage accounts, removing the delegated administrator account results in the deletion of your configuration policies and policy associations. Member accounts retain the configurations that they had before the delegated administrator was changed or removed. However, these accounts can't be managed by the removed delegated administrator account anymore. They become self-managed accounts that must be configured separately in each Region.

Choose your preferred method, and follow the instructions to remove the delegated Security Hub CSPM administrator account with AWS Organizations.

Organizations API, AWS CLI

To remove the delegated Security Hub CSPM administrator

From the organization management account, use the DeregisterDelegatedAdministrator operation of the Organizations API. If you're using the AWS CLI, run the deregister-delegated-administrator command. Provide the account ID of the delegated administrator, and the service principal for Security Hub CSPM, which is `securityhub.amazonaws.com`.

The following example removes the delegated Security Hub CSPM administrator. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws organizations deregister-delegated-administrator --account-id 123456789012 --
service-principal securityhub.amazonaws.com
```

Removing the delegated administrator (Security Hub CSPM console)

You can use the Security Hub CSPM console to remove the delegated Security Hub CSPM administrator in all Regions.

When the delegated Security Hub CSPM administrator account is removed, the member accounts are disassociated from the removed delegated Security Hub CSPM administrator account.

Security Hub CSPM is still enabled in the member accounts. They become standalone accounts until a new Security Hub CSPM administrator enables them as member accounts.

If the organization management account isn't an enabled account in Security Hub CSPM, then use the option on the **Welcome to Security Hub CSPM** page.

To remove the delegated Security Hub CSPM administrator account from the Welcome to Security Hub CSPM page

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Choose **Go to Security Hub**.
3. Under **Delegated Administrator**, choose **Remove**.

If the organization management account is an enabled account in **Security Hub**, then use the option on the **General** tab of the **Settings** page.

To remove the delegated Security Hub CSPM administrator account from the Settings page

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the Security Hub CSPM navigation pane, choose **Settings**. Then choose **General**.
3. Under **Delegated Administrator**, choose **Remove**.

Removing the delegated administrator (Security Hub CSPM API, AWS CLI)

You can use the Security Hub CSPM API or Security Hub CSPM operations for the AWS CLI to remove the delegated Security Hub CSPM administrator. When you remove the delegated administrator with one of these methods, it is only removed in the Region where the API call or command was issued. Security Hub CSPM doesn't update other Regions, and it doesn't remove the delegated administrator account in AWS Organizations.

Choose your preferred method, and follow these steps to remove the delegated Security Hub CSPM administrator account with Security Hub CSPM.

Security Hub CSPM API, AWS CLI

To remove the delegated Security Hub CSPM administrator

From the organization management account, use the DisableOrganizationAdminAccount operation of the Security Hub CSPM API. If you're using the AWS CLI, run the disable-organization-admin-account command. Provide the account ID of the delegated Security Hub CSPM administrator.

The following example removes the delegated Security Hub CSPM administrator. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub disable-organization-admin-account --admin-account-id 123456789012
```

Disabling Security Hub CSPM integration with AWS Organizations

After an AWS Organizations organization is integrated with AWS Security Hub CSPM, the Organizations management account can subsequently disable the integration. As a user of the Organizations management account, you can do this by disabling trusted access for Security Hub CSPM in AWS Organizations.

When you disable trusted access for Security Hub CSPM, the following occurs:

- Security Hub CSPM loses its status as a trusted service in AWS Organizations.
- The Security Hub CSPM delegated administrator account loses access to Security Hub CSPM settings, data, and resources for all Security Hub CSPM member accounts in all AWS Regions.
- If you were using central configuration, Security Hub CSPM automatically stops using it for your organization. Your configuration policies and policy associations are deleted. Accounts retain the configurations that they had before you disabled trusted access.
- All Security Hub CSPM member accounts become standalone accounts and retain their current settings. If Security Hub CSPM was enabled for a member account in one or more Regions, Security Hub CSPM continues to be enabled for the account in those Regions. Enabled standards and controls are also unchanged. You can change these settings separately in each account and Region. However, the account is no longer associated with a delegated administrator in any Region.

For additional information about the results of disabling trusted service access, see Using AWS Organizations with other AWS services in the *AWS Organizations User Guide*.

To disable trusted access, you can use the AWS Organizations console, Organizations API, or the AWS CLI. Only a user of the Organizations management account can disable trusted service access for Security Hub CSPM. For details about the permissions that you need, see Permissions required to disable trusted access in the *AWS Organizations User Guide*.

Before you disable trusted access, we recommend working with the delegated administrator for your organization to disable Security Hub CSPM in member accounts and to clean up Security Hub CSPM resources in those accounts.

Choose your preferred method, and follow the steps to disable trusted access for Security Hub CSPM.

Organizations console

To disable trusted access for Security Hub CSPM

1. Sign in to the AWS Management Console using the credentials of the AWS Organizations management account.
2. Open the Organizations console at https://console.aws.amazon.com/organizations/.
3. In the navigation pane, choose **Services**.

4. Under **Integrated services**, choose **AWS Security Hub CSPM**.
5. Choose **Disable trusted access**.
6. Confirm that you want to disable trusted access.

Organizations API

To disable trusted access for Security Hub CSPM

Invoke the DisableAWSServiceAccess operation of the AWS Organizations API. For the `ServicePrincipal` parameter, specify the Security Hub CSPM service principal (`securityhub.amazonaws.com`).

AWS CLI

To disable trusted access for Security Hub CSPM

Run the disable-aws-service-access command of the AWS Organizations API. For the `service-principal` parameter, specify the Security Hub CSPM service principal (`securityhub.amazonaws.com`).

Example:

```
aws organizations disable-aws-service-access --service-principal
 securityhub.amazonaws.com
```

Automatically enabling Security Hub CSPM in new organization accounts

When new accounts join your organization, they are added to the list on the **Accounts** page of the AWS Security Hub CSPM console. For organization accounts, **Type** is **By organization**. By default, new accounts don't become Security Hub CSPM members when they join the organization. Their status is **Not a member**. The delegated administrator account can automatically add new accounts as members and enable Security Hub CSPM in these accounts when they join the organization.

> **Note**
>
> Although many AWS Regions are active by default for your AWS account, you must activate certain Regions manually. These Regions are called opt-in Regions in this document. To automatically enable Security Hub CSPM in a new account in an opt-in Region, the account must have that Region activated first. Only the account owner can activate the opt-in

Region. For more information about opt-in Regions, see Specify which AWS Regions your account can use.

This process is different based on whether you use central configuration (recommended) or local configuration.

Automatically enabling new organization accounts (central configuration)

If you use central configuration, you can automatically enable Security Hub CSPM in new and existing organization accounts by creating a configuration policy in which Security Hub CSPM is enabled. You can then associate the policy with the organization root or specific organizational units (OUs).

If you associate a configuration policy in which Security Hub CSPM is enabled with a specific OU, Security Hub CSPM is automatically enabled in all accounts (existing and new) that belong to that OU. New accounts that don't belong to the OU are self-managed and don't automatically have Security Hub CSPM enabled. If you associate a configuration policy in which Security Hub CSPM is enabled with the root, Security Hub CSPM is automatically enabled in all accounts (existing and new) that join the organization. The exceptions are if an account uses a different policy through application or inheritance, or is self-managed.

In your configuration policy, you can also define which security standards and controls should be enabled in the OU. To generate control findings for enabled standards, the accounts in the OU must have AWS Config enabled and configured to record required resources. For more information about AWS Config recording, see Enabling and configuring AWS Config.

For instructions on creating a configuration policy, see Creating and associating configuration policies.

Automatically enabling new organization accounts (local configuration)

When you use local configuration and turn on automatic enablement of default standards, Security Hub CSPM adds *new* organization accounts as members and enables Security Hub CSPM in them in the current Region. Other Regions aren't affected. In addition, turning on automatic enablement doesn't enable Security Hub CSPM in *existing* organization accounts unless they were already added as member accounts.

After turning on automatic enablement, default security standards are enabled for new member accounts in the current Region when they join the organization. The default standards are AWS

Foundational Security Best Practices (FSBP) and Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0. You can't change the default standards. If you want to enable other standards throughout your organization, or enable standards for select accounts and OUs, we recommend using central configuration.

To generate control findings for the default standards (and other enabled standards), accounts in your organization must have AWS Config enabled and configured to record required resources. For more information about AWS Config recording, see Enabling and configuring AWS Config.

Choose your preferred method, and follow the steps to automatically enable Security Hub CSPM in new organization accounts. These instructions apply only if you use local configuration.

Security Hub CSPM console

To automatically enable new organization accounts as Security Hub CSPM members

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign is using the credentials of the delegated administrator account.
2. In the Security Hub CSPM navigation pane, under **Settings**, choose **Configuration**.
3. In the **Accounts** section, turn on **Auto-enable accounts**.

Security Hub CSPM API

To automatically enable new organization accounts as Security Hub CSPM members

Invoke the `UpdateOrganizationConfiguration` API from the delegated administrator account. Set the `AutoEnable` field to `true` to automatically enable Security Hub CSPM in new organization accounts.

AWS CLI

To automatically enable new organization accounts as Security Hub CSPM members

Run the `update-organization-configuration` command from the delegated administrator account. Include the `auto-enable` parameter to automatically enable Security Hub CSPM in new organization accounts.

```
aws securityhub update-organization-configuration --auto-enable
```

Manually enabling Security Hub CSPM in new organization accounts

If you don't automatically enable Security Hub CSPM in new organization accounts when they join the organization, then you can add those accounts as members and enable Security Hub CSPM in them manually after they join the organization. You must also manually enable Security Hub CSPM in AWS accounts that you previously disassociated from an organization.

> **Note**
>
> This section doesn't apply to you if you use central configuration. If you use central configuration, you can create configuration policies that enable Security Hub CSPM in specified member accounts and organizational units (OUs). You can also enable specific standards and controls in those accounts and OUs.

You can't enable Security Hub CSPM in an account if it is already a member account within a different organization.

You also can't enable Security Hub CSPM in an account that is currently suspended. If you try to enable the service in a suspended account, the account status changes to **Account Suspended**.

- If the account doesn't have Security Hub CSPM enabled, Security Hub CSPM is enabled in that account. The AWS Foundational Security Best Practices (FSBP) standard and CIS AWS Foundations Benchmark v1.2.0 also are enabled in the account unless your turn off default security standards.

  The exception to this is the Organizations management account. Security Hub CSPM cannot be enabled automatically in the Organizations management account. You must manually enable Security Hub CSPM in the Organizations management account before you can add it as a member account.
- If the account already has Security Hub CSPM enabled, Security Hub CSPM doesn't make any other changes to the account. It only enables the membership.

In order for Security Hub CSPM to generate control findings, member accounts must have AWS Config enabled and configured to record required resources. For more information, see Enabling and configuring AWS Config.

Choose your preferred method, and follow the steps to enable an organization account as a Security Hub CSPM member account.

Security Hub CSPM console

To manually enable organization accounts as Security Hub CSPM members

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated administrator account.
2. In the Security Hub CSPM navigation pane, under **Settings**, choose **Configuration**.
3. In the **Accounts** list, select each organization account that you want to enable.
4. Choose **Actions**, and then choose **Add member**.

Security Hub CSPM API

To manually enable organization accounts as Security Hub CSPM members

Invoke the `CreateMembers` API from the delegated administrator account. For each account to enable, provide the account ID.

Unlike the manual invitation process, when you invoke `CreateMembers` to enable an organization account, you don't need to send an invitation.

AWS CLI

To manually enable organization accounts as Security Hub CSPM members

Run the `create-members` command from the delegated administrator account. For each account to enable, provide the account ID.

Unlike the manual invitation process, when you run `create-members` to enable an organization account, you don't need to send an invitation.

```
aws securityhub create-members --account-details '[{"AccountId": "<accountId>"}]'
```

Example

```
aws securityhub create-members --account-details '[{"AccountId": "123456789111"},
 {"AccountId": "123456789222"}]'
```

Disassociating Security Hub CSPM member accounts from your organization

To stop receiving and viewing findings from an AWS Security Hub CSPM member account, you can disassociate the member account from your organization.

> **Note**
>
> If you use central configuration, disassociation works differently. You can create a configuration policy that disables Security Hub CSPM in one or more centrally managed member accounts. After that, these accounts are still part of the organization, but won't generate Security Hub CSPM findings. If you use central configuration but also have manually-invited member accounts, you can disassociate one or more manually-invited accounts.

Member accounts that are managed using AWS Organizations can't disassociate their accounts from the administrator account. Only the administrator account can disassociate a member account.

Disassociating a member account does not close the account. Instead, it removes the member account from the organization. The disassociated member account becomes a standalone AWS account that is no longer managed by the Security Hub CSPM integration with AWS Organizations.

Choose your preferred method, and follow the steps to disassociate a member account from the organization.

Security Hub CSPM console

To disassociate a member account from the organization

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated administrator account.
2. In the navigation pane, under **Settings**, choose **Configuration**.
3. In the **Accounts** section, select the accounts that you want to disassociate. If you use central configuration, you can select a manually-invited account to disassociate from the `Invitation accounts` tab. This tab is visible only if you use central configuration.
4. Choose **Actions**, and then choose **Disassociate account**.

Security Hub CSPM API

To disassociate a member account from the organization

Invoke the `DisassociateMembers` API from the delegated administrator account. You must provide the AWS account IDs for the member accounts to disassociate. To view a list of member accounts, invoke the `ListMembers` API.

AWS CLI

To disassociate a member account from the organization

Run the `>disassociate-members` command from the delegated administrator account. You must provide the AWS account IDs for the member accounts to disassociate. To view a list of member accounts, run the `>list-members` command.

```
aws securityhub disassociate-members --account-ids "<accountIds>"
```

Example

```
aws securityhub disassociate-members --account-ids "123456789111" "123456789222"
```

You can also use the AWS Organizations console, AWS CLI, or AWS SDKs to disassociate a member account from your organization. For more information, see Removing a member account from your organization in the *AWS Organizations User Guide*.

Managing accounts by invitation in Security Hub CSPM

You can centrally manage multiple AWS Security Hub CSPM accounts in two ways, by integrating Security Hub CSPM with AWS Organizations or by manually sending and accepting membership invitations. You must use the manual process if you have a standalone account or you don't integrate with AWS Organizations. In manual account management, the Security Hub CSPM administrator invites accounts to become members. The administrator-member relationship is established when a prospective member accepts the invitation. A Security Hub CSPM administrator account can manage Security Hub CSPM for up 1,000 invitation-based member accounts.

> **Note**
>
> If you create an invitation-based organization in Security Hub CSPM, you can subsequently transition to using AWS Organizations instead. If you have more than one member account,

we recommend using AWS Organizations instead of Security Hub CSPM invitations to manage your member accounts. For information, see Managing Security Hub CSPM for multiple accounts with AWS Organizations.

Cross-Region aggregation of findings and other data is available for accounts that you invite through the manual invitation process. However, the administrator must invite the member account from the aggregation Region and all linked Regions in order for cross-Region aggregation to work. In addition, the member account must have Security Hub CSPM enabled in the aggregation Region and all linked Regions to give the administrator the ability to view findings from the member account.

Configuration policies aren't supported for manually-invited member accounts. Instead, you must configure Security Hub CSPM settings separately in each member account and AWS Region when you use the manual invitation process.

You must also use the manual invitation-based process for accounts that don't belong to your organization. For example, you might not include a test account in your organization. Or, you might want to consolidate accounts from multiple organizations under a single Security Hub CSPM administrator account. The Security Hub CSPM administrator account must send invitations to accounts that belong to other organizations.

On the **Configuration** page of the Security Hub CSPM console, accounts that were added by invitation are listed in the **Invitation accounts** tab. If you use central configuration, but also invite accounts outside of your organization, you can view findings from invitation-based accounts in this tab. However, the Security Hub CSPM administrator can't configure invitation-based accounts across Regions through the use of configuration policies.

The topics in this section explain how to manage member accounts through invitations.

Topics

Adding and inviting member accounts in Security Hub CSPM

> **Note**
>
> We recommend using AWS Organizations instead of Security Hub CSPM invitations to manage your member accounts. For information, see Managing Security Hub CSPM for multiple accounts with AWS Organizations.

Your account becomes the AWS Security Hub CSPM administrator for accounts that accept your invitation to become a Security Hub CSPM member account.

When you accept an invitation from another account, your account becomes a member account, and that account becomes your administrator.

If your account is an administrator account, you can't accept an invitation to become a member account.

Adding a member account consists of the following steps:

1. The administrator account adds the member account to their list of member accounts.
2. The administrator account sends an invitation to the member account.
3. The member account accepts the invitation.

Adding member accounts

From the Security Hub CSPM console, you can add accounts to your list of member accounts. In the Security Hub CSPM console, you can select accounts individually, or upload a `.csv` file that contains the account information.

For each account, you must provide the account ID and an email address. The email address should be the email address to contact about security issues in the account. It is not used to verify the account.

Choose your preferred method, and follow the steps to add member accounts.

Security Hub CSPM console

To add accounts to your list of member accounts

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the administrator account.
2. In the left pane, choose **Settings**.
3. On the **Settings** page, choose **Accounts** and then choose **Add accounts**. You can then either add accounts individually or upload a `.csv` file containing the list of accounts.
4. To select the accounts, do one of the following:

   - To add the accounts individually, under **Enter accounts**, enter the account ID and email address of the account to add, and then choose **Add**.

     Repeat this process for each account.
   - To use a comma-separated values (.csv) file to add multiple accounts, first create the file. The file must contain the account ID and email address for each account to add.

     In your `.csv` list, accounts must appear one per line. The first line of the `.csv` file must contain the header. In the header, the first column is **`Account ID`** and the second column is **`Email`**.

     Each subsequent line must contain a valid account ID and email address for the account to add.

     Here is an example of a `.csv` file when viewed in a text editor.

     ```
     Account ID,Email
     111111111111,user@example.com
     ```

     In a spreadsheet program, the fields appear in separate columns. The underlying format is still comma-separated. You must format the account IDs as non-decimal numbers. For example, the account ID 444455556666 cannot be formatted as 444455556666.0. Also make sure that the number formatting does not remove any leading zeros from the account ID.

     To select the file, on the console, choose **Upload list (.csv)**. Then choose **Browse**.

After you select the file, choose **Add accounts**.

5. After you finish adding accounts, under **Accounts to be added**, choose **Next**.

Security Hub CSPM API

To add accounts to your list of member accounts

Invoke the `CreateMembers` API from the administrator account. For each member account to add, you must provide the AWS account ID.

AWS CLI

To add accounts to your list of member accounts

Run the `create-members` command from the administrator account. For each member account to add, you must provide the AWS account ID.

```
aws securityhub create-members --account-details '[{"AccountId": "<accountID1>"}]'
```

Example

```
aws securityhub create-members --account-details '[{"AccountId": "123456789111"},
 {"AccountId": "123456789222"}]'
```

Inviting member accounts

After you add the member accounts, you send an invitation to the member account. You can also resend an invitation to an account that you disassociated from the administrator.

Security Hub CSPM console

To invite prospective member accounts

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the administrator account.
2. In the navigation pane, choose **Settings**, and then choose **Accounts**.
3. For the account to invite, choose **Invite** in the **Status** column.

4. When prompted to confirm, choose **Invite**.

> **Note**
>
> To resend invitations to disassociated accounts, select each disassociated account on the **Accounts** page. For **Actions**, choose **Resend invitation**.

Security Hub CSPM API

To invite prospective member accounts

Invoke the `InviteMembers` API from the administrator account. For each account to invite, you must provide the AWS account ID.

AWS CLI

To invite prospective member accounts

Run the `invite-members` command from the administrator account. For each account to invite, you must provide the AWS account ID.

```
aws securityhub invite-members --account-ids <accountIDs>
```

Example

```
aws securityhub invite-members --account-ids "123456789111" "123456789222"
```

Responding to an invitation to be a Security Hub CSPM member account

> **Note**
>
> We recommend using AWS Organizations instead of Security Hub CSPM invitations to manage your member accounts. For information, see Managing Security Hub CSPM for multiple accounts with AWS Organizations.

You can accept or decline an invitation to be an AWS Security Hub CSPM member account.

If you accept an invitation, your account becomes a Security Hub CSPM member account. The account that sent the invitation becomes your Security Hub CSPM administrator account. The administrator account user can view findings for your member account in Security Hub CSPM.

If you decline the invitation, then your account is marked as **Resigned** on the administrator account's list of member accounts.

You can only accept one invitation to be a member account.

Before you can accept or decline an invitation, you must enable Security Hub CSPM.

Remember that all Security Hub CSPM accounts must have AWS Config enabled and configured to record all resources. For details on the requirement for AWS Config, see Enabling and configuring AWS Config.

Accepting an invitation

You can send an invitation to be a Security Hub CSPM member account from the administrator account. You can then accept the invitation after signing in to the member account.

Choose your preferred method, and follow the steps to accept an invitation to be a member account.

Security Hub CSPM console

To accept a membership invitation

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Settings**, and then choose **Accounts**.
3. In the **Administrator account** section, turn on **Accept**, and then choose **Accept invitation**.

Security Hub CSPM API

To accept a membership invitation

Invoke the `AcceptAdministratorInvitation` API. You must provide the invitation identifier and the AWS account ID of the administrator account. To retrieve details about the invitation, use the `ListInvitations` operation.

AWS CLI

To accept a membership invitation

Run the `accept-administrator-invitation` command. You must provide the invitation identifier and the AWS account ID of the administrator account. To retrieve details about the invitation, run the `list-invitations` command.

```
aws securityhub accept-administrator-invitation --administrator-
id <administratorAccountID> --invitation-id <invitationID>
```

Example

```
aws securityhub accept-administrator-invitation --administrator-id 123456789012 --
invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Note

The Security Hub CSPM console continues to use `AcceptInvitation`. It will eventually change to use `AcceptAdministratorInvitation`. Any IAM policies that specifically control access to this function must continue to use `AcceptInvitation`. You should also add `AcceptAdministratorInvitation` to your policies to ensure that the correct permissions are in place after the console begins to use `AcceptAdministratorInvitation`.

Declining an invitation

You can decline an invitation to be a Security Hub CSPM member account. When you decline an invitation in the Security Hub CSPM console, your account is marked as **Resigned** on the administrator account's list of member accounts. The **Resigned** status appears only when you sign in to the Security Hub CSPM console using the administrator account. However, the invitation remains unchanged in the console for the member account until you sign in to the administrator account and delete the invitation.

To decline an invitation, you must sign in to the member account that received the invitation.

Choose your preferred method, and follow the steps to decline an invitation to be a member account.

Security Hub CSPM console

To decline a membership invitation

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Settings**, and then choose **Accounts**.
3. In the **Administrator account** section, choose **Decline invitation**.

Security Hub CSPM API

To decline a membership invitation

Invoke the `DeclineInvitations` API. You must provide the AWS account ID of the administrator account that issued the invitation. To view information about your invitations, use the `ListInvitations` operation.

AWS CLI

To decline a membership invitation

Run the `decline-invitations` command. You must provide the AWS account ID of the administrator account that issued the invitation. To view information about your invitations, run the `list-invitations` command.

```
aws securityhub decline-invitations --account-ids "<administratorAccountId>"
```

Example

```
aws securityhub decline-invitations --account-ids "123456789012"
```

Disassociating member accounts in Security Hub CSPM

> **Note**
>
> We recommend using AWS Organizations instead of Security Hub CSPM invitations to manage your member accounts. For information, see Managing Security Hub CSPM for multiple accounts with AWS Organizations.

An AWS Security Hub CSPM administrator account can disassociate a member account to stop receiving and viewing findings from that account. You must disassociate a member account before you can delete it.

When you disassociate a member account, it remains in your list of member accounts with a status of **Removed (Disassociated)**. Your account is removed from the administrator account information for the member account.

To resume receiving findings for the account, you can resend the invitation. To remove the member account entirely, you can delete the member account.

Choose your preferred method, and follow the steps to disassociate a manually-invited member account from the administrator account.

Security Hub CSPM console

To disassociate a manually-invited member account

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the administrator account.
2. In the navigation pane, under **Settings**, choose **Configuration**.
3. In the **Accounts** section, select the accounts that you want to disassociate.
4. Choose **Actions**, and then choose **Disassociate account**.

Security Hub CSPM API

To disassociate a manually-invited member account

Invoke the `DisassociateMembers` API from the administrator account. You must provide the AWS account IDs of the member accounts that you want to disassociate. To view a list of member accounts, use the `ListMembers` operation.

AWS CLI

To disassociate a manually-invited member account

Run the `disassociate-members` command from the administrator account. You must provide the AWS account IDs of the member accounts that you want to disassociate. To view a list of member accounts, run the `list-members` command.

```
aws securityhub disassociate-members --account-ids <accountIds>
```

Example

```
aws securityhub disassociate-members --account-ids "123456789111" "123456789222"
```

Deleting member accounts in Security Hub CSPM

> **Note**
>
> We recommend using AWS Organizations instead of Security Hub CSPM invitations to manage your member accounts. For information, see Managing Security Hub CSPM for multiple accounts with AWS Organizations.

As an AWS Security Hub CSPM administrator account, you can delete member accounts that were added by invitation. Before you can delete an enabled account, you must disassociate it.

When you delete a member account, it is completely removed from the list. To restore the account's membership, you must add and invite it again as if it were a completely new member account.

You can't delete accounts that belong to an organization and that are managed using the integration with AWS Organizations.

Choose your preferred method, and follow the steps to delete manually-invited member accounts.

Security Hub CSPM console

To delete a manually-invited member account

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the administrator account.
2. In the navigation pane, choose **Settings**, and then choose **Configuration**.
3. Choose the **Invitation accounts** tab. Then, select the accounts to delete.

4. Choose **Actions**, and then choose **Delete**. This option is available only if you have disassociated the account. You must disassociate a member account before it can be deleted.

Security Hub CSPM API

To delete a manually-invited member account

Invoke the `DeleteMembers` API from the administrator account. You must provide the AWS account IDs of the member accounts that you want to delete. To retrieve the list of member accounts, invoke the `ListMembers` API.

AWS CLI

To delete a manually-invited member account

Run the `delete-members` command from the administrator account. You must provide the AWS account IDs of the member accounts that you want to delete. To retrieve the list of member accounts, run the `list-members` command.

```
aws securityhub delete-members --account-ids <memberAccountIDs>
```

Example

```
aws securityhub delete-members --account-ids "123456789111" "123456789222"
```

Disassociating from a Security Hub CSPM administrator account

> **Note**
>
> We recommend using AWS Organizations instead of Security Hub CSPM invitations to manage your member accounts. For information, see Managing Security Hub CSPM for multiple accounts with AWS Organizations.

If your account was added as an AWS Security Hub CSPM member account by invitation, you can disassociate the member account from the administrator account. After you disassociate a member account, Security Hub CSPM doesn't send findings from the account to the administrator account.

Member accounts that are managed using the integration with AWS Organizations can't disassociate their accounts from the administrator account. Only the Security Hub CSPM delegated administrator can disassociate member accounts that are managed with Organizations.

When you disassociate from your administrator account, your account remains in the administrator account's member list with a status of **Resigned**. However, the administrator account does not receive any findings for your account.

After you disassociate yourself from the administrator account, the invitation to be a member still remains. You can accept the invitation again in the future.

Security Hub CSPM console

To disassociate from your administrator account

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Settings**, and then choose **Accounts**.
3. In the **Administrator account** section, turn off **Accept**, and then choose **Update**.

Security Hub CSPM API

To disassociate from your administrator account

Invoke the `DisassociateFromAdministratorAccount` API.

AWS CLI

To disassociate from your administrator account

Run the `disassociate-from-administrator-account` command.

```
aws securityhub disassociate-from-administrator-account
```

Note

The Security Hub CSPM console continues to use `DisassociateFromMasterAccount`. It will eventually change to use `DisassociateFromAdministratorAccount`. Any IAM policies that specifically control access to this function must continue to use `DisassociateFromMasterAccount`. You should also add

`DisassociateFromAdministratorAccount` to your policies to ensure that the correct permissions are in place after the console begins to use `DisassociateFromAdministratorAccount`.

Transitioning to Organizations to manage accounts in Security Hub CSPM

When you manage accounts manually in AWS Security Hub CSPM, you must invite prospective member accounts and configure each member account separately in each AWS Region.

By integrating Security Hub CSPM and AWS Organizations, you can eliminate the need to send invitations and gain more control over how Security Hub CSPM is configured and customized in your organization. For this reason, we recommend using AWS Organizations instead of Security Hub CSPM invitations to manage your member accounts. For information, see Managing Security Hub CSPM for multiple accounts with AWS Organizations.

It's possible to use a combined approach in which you use the AWS Organizations integration, but also manually invite accounts outside of your organization. However, we recommend exclusively using the Organizations integration. Central configuration, a feature which helps you manage Security Hub CSPM across multiple accounts and Regions, is only available when you integrate with Organizations.

This section covers how you can transition from manual invitation-based account management to managing accounts with AWS Organizations.

Integrating Security Hub CSPM with AWS Organizations

First, you must integrate Security Hub CSPM and AWS Organizations.

You can integrate these services by completing the following steps:

- Create an organization in AWS Organizations. For instructions, see Create an organization in the *AWS Organizations User Guide*.
- From the Organizations management account, designate a Security Hub CSPM delegated administrator account.

Note

The organization management account *cannot* be set as the DA account.

For detailed instructions, see Integrating Security Hub CSPM with AWS Organizations.

By completing the preceding steps, you grant trusted access for Security Hub CSPM in AWS Organizations. This also enables Security Hub CSPM in the current AWS Region for the delegated administrator account.

The delegated administrator can manage the organization in Security Hub CSPM, primarily by adding the organization's accounts as Security Hub CSPM member accounts. The administrator can also access certain Security Hub CSPM settings, data, and resources for those accounts.

When you transition to account management using Organizations, invitation-based accounts don't automatically become Security Hub CSPM members. Only the accounts that you add to your new organization can become Security Hub CSPM members.

After activating the integration, you can manage accounts with Organizations. For information, see Managing Security Hub CSPM for multiple accounts with AWS Organizations. Account management varies based on your organization's configuration type.

Allowed actions by administrator and member accounts in Security Hub CSPM

Administrator and member accounts have access to AWS Security Hub CSPM actions noted in the following tables. In the tables, the values have the following meanings:

- **Any –** The account can perform the action for any member account under the same administrator.
- **Current –** The account can perform the action only for itself (the account that you're currently signed in to).
- **Dash –** Indicates that the account cannot perform the action.

As noted in the tables, allowed actions differ based on whether you integrate with AWS Organizations and which configuration type your organization uses. For information about the difference between central and local configuration, see Managing accounts with AWS Organizations.

Security Hub CSPM doesn't copy member account findings into the administrator account. In Security Hub CSPM, all findings are ingested into a specific Region for a specific account. In each Region, the administrator account can view and manage findings for their member accounts in that Region.

If you set an aggregation Region, the administrator account can view and manage member account findings from linked Regions that are replicated to the aggregation Region. For more information about cross-Region aggregation, see Cross-Region aggregation.

The following tables specify the default permissions for administrator and member accounts. You can use custom IAM policies to further restrict access to Security Hub CSPM features and functions. For guidance and examples, see the blog post Aligning IAM policies to user personas for AWS Security Hub CSPM.

Allowed actions if you integrate with Organizations and use central configuration

Administrator and member accounts can access Security Hub CSPM actions as follows if you integrate with Organizations and use central configuration.

Action	Security Hub CSPM delegated administrator account	Centrally managed member account	Self-managed member account
Create and manage Security Hub CSPM configuration policies	For self and centrally managed accounts	–	–
View organization accounts	Any	–	–
Disassociate member account	Any	–	–
Delete member account	Any non-organization account	–	–
Disable Security Hub CSPM	For current account and centrally managed accounts	–	Current (must be disassociated from the administrator account)
View findings and finding history	Any	Current	Current
Update findings	Any	Current	Current

Action	Security Hub CSPM delegated administrator account	Centrally managed member account	Self-managed member account
View insight results	Any	Current	Current
View control details	Any	Current	Current
Turn consolidated control findings on or off	Any	–	–
Enable and disable standards	For current account and centrally managed accounts	–	Current
Enable and disable controls	For current account and centrally managed accounts	–	Current
Enable and disable integrations	Current	Current	Current
Configure cross-Region aggregation	Any	–	–
Select home Region and linked Regions	Any (must stop and restart central configuration to change home Region)	–	–
Configure custom actions	Current	Current	Current
Configure automation rules	Any	–	–
Configure custom insights	Current	Current	Current

Allowed actions if you integrate with Organizations and use local configuration

Administrator and member accounts can access Security Hub CSPM actions as follows if you integrate with Organizations and use local configuration.

Action	Security Hub CSPM delegated administrator account	Member account
Create and manage Security Hub CSPM configuration policies	–	–
View organization accounts	Any	–
Disassociate member account	Any	–
Delete member account	–	–
Disable Security Hub CSPM	–	Current (if account is disassociated from delegated administrator)
View findings and finding history	Any	Current
Update findings	Any	Current
View insight results	Any	Current
View control details	Any	Current
Turn consolidated control findings on or off	Any	–
Enable and disable standards	Current	Current
Automatically enable Security Hub CSPM and default	For current account and new organization accounts	–

Action	Security Hub CSPM delegated administrator account	Member account
standards in new organization accounts		
Enable and disable controls	Current	Current
Enable and disable integrati ons	Current	Current
Configure cross-Region aggregation	Any	–
Configure custom actions	Current	Current
Configure automation rules	Any	–
Configure custom insights	Current	Current

Allowed actions for invitation-based accounts

Administrator and member accounts can access Security Hub CSPM actions as follows if you use the invitation-based method to manually manage accounts instead of integrating with AWS Organizations.

Action	Security Hub CSPM administrator account	Member account
Create and manage Security Hub CSPM configuration policies	–	–
View organization accounts	Any	–
Disassociate member account	Any	Current
Delete member account	Any	–

Action	Security Hub CSPM administrator account	Member account
Disable Security Hub CSPM	Current (if there are no enabled member accounts)	Current (if account is disassociated from administrator account)
View findings and finding history	Any	Current
Update findings	Any	Current
View insight results	Any	Current
View control details	Any	Current
Turn consolidated control findings on or off	Any	–
Enable and disable standards	Current	Current
Automatically enable Security Hub CSPM and default standards in new organization accounts	–	–
Enable and disable controls	Current	Current
Enable and disable integrations	Current	Current
Configure cross-Region aggregation	Any	–
Configure custom actions	Current	Current
Configure automation rules	Any	–
Configure custom insights	Current	Current

Effect of account actions on Security Hub CSPM data

These account actions have the following effects on AWS Security Hub CSPM data.

Security Hub CSPM disabled

If you use central configuration, the delegated administrator (DA) can create Security Hub CSPM configuration policies that disable AWS Security Hub CSPM in specific accounts and organizational units (OUs). In this case, Security Hub CSPM is disabled in the specified accounts and OUs in your home Region and any linked Regions. If you don't use central configuration, you must disable Security Hub CSPM separately in each account and Region where you enabled it. You can't use central configuration if Security Hub CSPM is disabled in the DA account.

No findings are generated or updated for the administrator account if Security Hub CSPM is disabled in the administrator account. Existing archived findings are deleted after 30 days. Existing active findings are deleted after 90 days.

Integrations with other AWS services are removed.

Enabled security standards and controls are disabled.

Other Security Hub CSPM data and settings, including custom actions, insights, and subscriptions to third-party products are retained for 90 days.

Member account disassociated from administrator account

When a member account is disassociated from the administrator account, the administrator account loses permission to view findings in the member account. However, Security Hub CSPM is still enabled in both accounts.

If you use central configuration, the DA can't configure Security Hub CSPM for a member account that's disassociated from the DA account.

Custom settings or integrations that are defined for the administrator account are not applied to findings from the former member account. For example, after the accounts are disassociated, you might have a custom action in the administrator account used as the event pattern in an Amazon EventBridge rule. However, this custom action cannot be used in the member account.

In the **Accounts** list for the Security Hub CSPM administrator account, a removed account has a status of **Disassociated**.

Member account is removed from an organization

When a member account is removed from an organization, the Security Hub CSPM administrator account loses permission to view findings in the member account. However, Security Hub CSPM is still enabled in both accounts with the same settings they had before removal.

If you use central configuration, you can't configure Security Hub CSPM for a member account after it's removed from the organization to which the delegated administrator belongs. However, the account retains the settings it had prior to removal unless you manually change them.

In the **Accounts** list for the Security Hub CSPM administrator account, a removed account has a status of **Deleted**.

Account is suspended

When an AWS account is suspended, the account loses permission to view their findings in Security Hub CSPM. No findings are generated or updated for that account. The administrator account for a suspended account can view existing findings for the account.

For an organization account, the member account status can also change to **Account Suspended**. This happens if the account is suspended at the same time that the administrator account attempts to enable the account. The administrator account for an **Account Suspended** account cannot view findings for that account. Otherwise, the suspended status doesn't affect the member account status.

If you use central configuration, policy association fails if the delegated administrator tries to associate a configuration policy with a suspended account.

After 90 days, the account is either terminated or reactivated. When the account is reactivated, its Security Hub CSPM permissions are restored. If the member account status is **Account Suspended**, the administrator account must enable the account manually.

Account is closed

When an AWS account is closed, Security Hub CSPM responds to the closure as follows.

If the account is a Security Hub CSPM administrator account, it is removed as an administrator account and all the member accounts are removed. If the account is a member account, it is disassociated and removed as a member from the Security Hub CSPM administrator account.

Security Hub CSPM retains existing archived findings in the account for 30 days. For a control finding, the calculation of 30 days is based on the value for the `UpdatedAt` field of the finding. For another type of finding, the calculation is based on the value for the `UpdatedAt` or `ProcessedAt` field of the finding, whichever date is latest. At the end of this 30-day period, Security Hub CSPM permanently deletes the finding from the account.

Security Hub CSPM retains existing active findings in the account for 90 days. For a control finding, the calculation of 90 days is based on the value for the `UpdatedAt` field of the finding. For another type of finding, the calculation is based on the value for the `UpdatedAt` or `ProcessedAt` field of the finding, whichever date is latest. At the end of this 90-day period, Security Hub CSPM permanently deletes the finding from the account.

For longer-term retention of existing findings, you can export the findings to an S3 bucket. You can do this by using a custom action with an Amazon EventBridge rule. For more information, see Using EventBridge for automated response and remediation.

Important

For customers in AWS GovCloud (US) Regions, back up and then delete your policy data and other account resources before you close your account. You won't have access to the resources and data after you close your account.

For more information, see Close an AWS account in the *AWS Account Management Reference Guide*.

Understanding cross-Region aggregation in Security Hub CSPM

Note

The *aggregation Region* is now called the *home Region*. Some Security Hub CSPM API operations still use the older term aggregation Region.

By using cross-Region aggregation in AWS Security Hub CSPM, you can aggregate findings, finding updates, insights, control compliance statuses, and security scores from multiple AWS Regions to a single home Region. You can then manage all of this data from the home Region.

Suppose you set US East (N. Virginia) as the home Region, and US West (Oregon) and US West (N. California) as the linked Regions. When you view the **Findings** page in US East (N. Virginia), you

see the findings from all three Regions. Updates to those findings are also reflected in all three Regions.

> **Note**
>
> In AWS GovCloud (US), cross-Region aggregation is supported only for findings, finding updates, and insights across AWS GovCloud (US). Specifically, you can only aggregate findings, finding updates, and insights between AWS GovCloud (US-East) and AWS GovCloud (US-West). In the China Regions, cross-Region aggregation is supported only for findings, finding updates, and insights across the China Regions. Specifically, you can only aggregate findings, finding updates, and insights between China (Beijing) and China (Ningxia).

If a control is enabled in a linked Region but disabled in the home Region, you can see the compliance status of the control from the home Region, but you can't enable or disable that control from the home Region. The exception is if you use central configuration. If you use central configuration, the delegated Security Hub CSPM administrator can configure controls in the home Region and linked Regions from the home Region.

If you have set an home Region, security scores account for control statuses in all linked Regions. To view cross-Region security scores and compliance statuses, add the following permissions to your IAM role that uses Security Hub CSPM:

- `ListSecurityControlDefinitions`
- `BatchGetStandardsControlAssociations`
- `BatchUpdateStandardsControlAssociations`

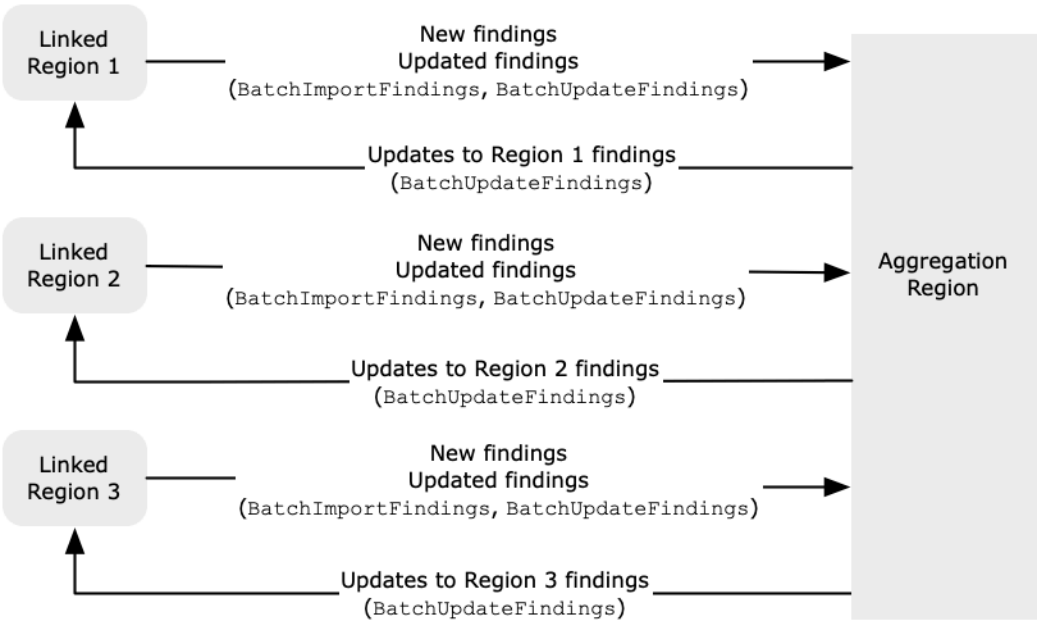
Types of data that are aggregated

When cross-Region aggregation is enabled with one or more linked Regions, Security Hub CSPM replicates the following data from the linked Regions to the home Region. This occurs in every account that has cross-Region aggregation enabled.

- Findings
- Insights
- Control compliance statuses

- Security scores

In addition to new data in the previous list, Security Hub CSPM also replicates updates to this data between the linked Regions and the home Region. Updates that occur in a linked Region are replicated to the home Region. Updates that occur in the home Region are replicated back to the linked Region. If there are conflicting updates in the home Region and the linked Region, then the most recent update is used.



Cross-Region aggregation does not add to the cost of Security Hub CSPM. You are not charged when Security Hub CSPM replicates new data or updates.

In the home Region, the **Summary** page provides a view of your active findings across linked Regions. For information, see Viewing a cross-Region summary of findings by severity. Other **Summary** page panels that analyze findings also display information from across the linked Regions.

Your security scores in the home Region are calculated by comparing the number of passed controls to the number of enabled controls in all linked Regions. In addition, if a control is enabled in at least one linked Region, it is visible on the **Security standards** details pages of the home Region. The compliance status of controls on the standards details pages reflects findings across linked Regions. If a security check associated with a control fails in one or more linked Regions, the compliance status of that control shows as **Failed** on the standards details pages of the home Region. The number of security checks includes findings from all linked Regions.

Security Hub CSPM only aggregates data from Regions where an account has Security Hub CSPM enabled. Security Hub CSPM is not automatically enabled for an account based on the cross-Region aggregation configuration.

It's possible to have cross-Region aggregation enabled without any linked Regions selected. In this case, no data replication occurs.

Aggregation for administrator and member accounts

Standalone accounts, member accounts, and administrator accounts can configure cross-Region aggregation. If configured by an administrator, the presence of the administrator account is essential for cross-Region aggregation to work in administered accounts. If the administrator account is removed or disassociated from a member account, cross-Region aggregation for the member account stops. This is true even if the account had cross-Region aggregation enabled before the administrator-member relationship begins.

When an administrator account enables cross-Region aggregation, Security Hub CSPM replicates the data that the administrator account generates in all linked Regions to the home Region. In addition, Security Hub CSPM identifies the member accounts that are associated with that administrator, and each member account inherits the cross-Region aggregation settings of the administrator. Security Hub CSPM replicates the data that a member account generates in all linked Regions to the home Region.

The administrator can access and manage security findings from all member accounts within the administered regions. However, as a Security Hub CSPM administrator, you must be signed in to the home Region to view aggregated data from all member accounts and linked Regions.

As a Security Hub CSPM member account, you must be signed in to the home Region to view aggregated data from your account from all linked Regions. Member accounts don't have permissions to view data from other member accounts.

An administrator account may manually invite member accounts or serve as the delegated administrator of an organization that is integrated with AWS Organizations. For a manually-invited member account, the administrator must invite the account from the home Region and all linked Regions in order for cross-Region aggregation to work. In addition, the member account must have Security Hub CSPM enabled in the home Region and all linked Regions to give the administrator the ability to view findings from the member account. If you don't use the home Region for other purposes, you can disable Security Hub CSPM standards and integrations in that Region to prevent charges.

If you plan to use cross-Region aggregation, and have multiple administrator accounts, we recommend following these best practices:

- Each administrator account has different member accounts.
- Each administrator account has the same member accounts across Regions.
- Each administrator account uses a different home Region.

> **Note**
>
> To understand how cross-Region aggregation impacts central configuration, see Impact of central configuration on cross-Region aggregation.

Impact of central configuration on cross-Region aggregation

Central configuration is an opt-in feature in AWS Security Hub CSPM that you can use if you integrate with AWS Organizations. If you use central configuration, the delegated administrator account can configure the Security Hub CSPM service, standards, and controls for accounts and organizational units (OU) in the organization. To configure accounts and OUs, the delegated administrator creates Security Hub CSPM configuration policies. Configuration policies can be used to define whether Security Hub CSPM is enabled or disabled, and which standards and controls are enabled. The delegated administrator associates configuration policies with specific accounts, OUs, or the root (the entire organization).

The delegated administrator can create and manage configuration policies for the organization only from the home Region. In addition, configuration policies take effect in the home Region and all linked Regions. You can't create a configuration policy that applies only in some linked Regions and not others. For information about cross-Region aggregation, see Cross-Region aggregation.

To use central configuration, you must designate a home Region. Optionally, you can choose one or more Regions as linked Regions. You can also choose to designate a home Region without any linked Regions.

Changing your cross-Region aggregation settings can impact your configuration policies. When you add a linked Region, your configuration policies take effect in that Region. If the Region is an opt-in Region, the Region must be enabled in order for your configuration policies to take effect there. Conversely, when you remove a linked Region, configuration policies no longer take effect in

that Region. In that Region, accounts maintain the settings they had when the linked Region was removed. You can change those settings, but must do so separately in each account and Region.

If you remove or change the home Region, your configuration policies and policy associations are deleted. You can no longer use central configuration or create configuration policies in any Region. Accounts maintain the settings they had before the home Region was changed or removed. You can change those settings at any time, but since you no longer use central configuration, settings must be modified separately in each account and Region. You can use central configuration and create configuration policies again if you designate a new home Region.

For more information about central configuration, see Understanding central configuration in Security Hub CSPM.

Enabling cross-Region aggregation

> **Note**
>
> The *aggregation Region* is now called the *home Region*. Some Security Hub CSPM API operations still use the older term aggregation Region.

You must enable cross-Region aggregation from the AWS Region that you want to designate as the home Region.

To enable cross-Region aggregation, you create a Security Hub CSPM resource called a finding aggregator. The finding aggregator resource specifies your home Region and linked Regions (if any).

You can't use an AWS Region that is disabled by default as your home Region. For a list of Regions that are disabled by default, see Enabling a Region in the *AWS General Reference*.

When you enable cross-Region aggregation, you choose to specify one or more linked Regions if you wish. You can also choose whether to automatically link new Regions when Security Hub CSPM begins to support them and you have opted into them.

Security Hub CSPM console

To enable cross-Region aggregation

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

2. Using the AWS Region selector, sign in to the Region that you want to use as the aggregation Region.
3. In the Security Hub CSPM navigation menu, choose **Settings** and then **Regions**.
4. For **Finding aggregation**, choose **Configure finding aggregation**.

   By default, the home Region is set to **No aggregation Region**.
5. Under **Aggregation Region**, select the option to designate the current Region as the home Region.
6. Optionally, for **Linked Regions**, select the Regions to aggregate data from.
7. To automatically aggregate data from new Regions in the partition as Security Hub CSPM supports them and you opt into them, select **Link future Regions**.
8. Choose **Save**.

Security Hub CSPM API

From the Region that you want to use as the home Region, use the `CreateFindingAggregator` operation of the Security Hub CSPM API. If you use the AWS CLI, run the create-finding-aggregator command.

For `RegionLinkingMode`, choose one of the following options:

- `ALL_REGIONS` – Security Hub CSPM aggregates data from all Regions. Security Hub CSPM also aggregates data from new Regions as they are supported and you opt into them.
- `ALL_REGIONS_EXCEPT_SPECIFIED` – Security Hub CSPM aggregates data from all Regions except for Regions that you want to exclude. Security Hub CSPM also aggregates data from new Regions as they are supported and you opt into them. Use `Regions` to provide the list of Regions to exclude from aggregation.
- `SPECIFIED_REGIONS` – Security Hub CSPM aggregates data from a selected list of Regions. Security Hub CSPM does not aggregate data automatically from new Regions. Use `Regions` to provide the list of Regions to aggregate from.
- `NO_REGIONS` – Security Hub CSPM doesn't aggregate data because you don't select any linked Regions.

The following example configures cross-Region aggregation. The home Region is US East (N. Virginia). The linked Regions are US West (N. California) and US West (Oregon). This example is

formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub create-finding-aggregator --region us-east-1 --region-linking-
mode SPECIFIED_REGIONS --regions us-west-1 us-west-2
```

Reviewing cross-Region aggregation settings

> **Note**
>
> The *aggregation Region* is now called the *home Region*. Some Security Hub CSPM API operations still use the older term aggregation Region.

You can view the current cross-Region aggregation configuration in AWS Security Hub CSPM from any AWS Region. The configuration includes the home Region, the linked Regions (if any), and whether to automatically link new Regions as Security Hub CSPM supports them.

Member accounts can view the cross-Region aggregation settings that the administrator account configured.

Choose your preferred method, and follow the steps to view your current cross-Region aggregation settings.

Security Hub CSPM console

To view cross-Region aggregation settings (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. On the navigation pane, choose **Settings** and then the **Regions** tab.

If cross-Region aggregation is not enabled, then the **Regions** tab displays the option to enable cross-Region aggregation. Only administrator accounts and standalone accounts can enable cross-Region aggregation.

If cross-Region aggregation is enabled, then the **Regions** tab displays the following information:

- The home Region

- Whether to automatically aggregate findings, insights, control statuses, and security scores from new Regions that Security Hub CSPM supports and that you opt into
- The list of linked Regions (if any are selected)

Security Hub CSPM API

To review cross-Region aggregation settings (Security Hub CSPM API)

Use the `GetFindingAggregator` operation of the Security Hub CSPM API. If you use the AWS CLI, run the get-finding-aggregator command.

When you make the request, provide the finding aggregator ARN. To obtain the finding aggregator ARN, use the `ListFindingAggregators` operation or list-finding-aggregators command.

The following example shows the cross-Region aggregation settings for the specified finding aggregator ARN. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability

```
$aws securityhub get-finding-aggregator --finding-aggregator-
arn arn:aws:securityhub:us-east-1:222222222222:finding-aggregator/123e4567-
e89b-12d3-a456-426652340000
```

Updating cross-Region aggregation settings

> **Note**
>
> The *aggregation Region* is now called the *home Region*. Some Security Hub CSPM API operations still use the older term aggregation Region.

You can update your current cross-Region aggregation settings in AWS Security Hub CSPM by changing the linked Regions or the current home Region. You can also change whether to automatically aggregate data from new AWS Regions that Security Hub CSPM is supported in.

Changes to cross-Region aggregation aren't implemented for an opt-in Region until you enable the Region in your AWS account. Regions that AWS introduced on or after to March 20, 2019 are opt-in Regions.

When you stop aggregating data from a linked Region, AWS Security Hub CSPM doesn't remove any existing aggregated data from that Region that is accessible in the home Region.

You can't use the update procedures in this section to change the home Region. To change the home Region, you must do the following:

1. Stop cross-Region aggregation. For instructions, see the section called “Stopping aggregation”.
2. Change to the Region that you want to be the new home Region.
3. Enable cross-Region aggregation. For instructions, see the section called “Enabling aggregation”.

You must update the cross-Region aggregation configuration from the current home Region.

Security Hub CSPM console

To change the linked Regions

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in to the current aggregation Region.
2. In the Security Hub CSPM navigation menu, choose **Settings**, then choose **Regions**.
3. For **Finding aggregation**, choose **Edit**.
4. For **Linked Regions**, update the selected linked Regions.
5. If needed, change whether **Link future Regions** is selected. This setting determines whether Security Hub CSPM automatically links new Regions as it adds support for them and you opt into them.
6. Choose **Save**.

Security Hub CSPM API

Use the `UpdateFindingAggregator` operation. If you use the AWS CLI, run the update-finding-aggregator command. To identify the finding aggregator, you must provide the finding aggregator ARN. To obtain the finding aggregator ARN, use the `ListFindingAggregators` operation or list-finding-aggregators command..

If the linking mode is `ALL_REGIONS_EXCEPT_SPECIFIED` or `SPECIFIED_REGIONS`, you can change the list of excluded or included Regions. If you want to change the Region linking mode to `NO_REGIONS`, you shouldn't provide a Regions list.

When you change the list of excluded or included Regions, you must provide the full list with the updates. For example, suppose you currently aggregate findings from US East (Ohio), and want to also aggregate findings from US West (Oregon). You must provide a `Regions` list that contains both US East (Ohio) and US West (Oregon).

The following example updates cross-Region aggregation to selected Regions. The command is run from the current home Region, which is US East (N. Virginia). The linked Regions are US West (N. California) and US West (Oregon). This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
aws securityhub update-finding-aggregator --region us-east-1 --finding-
aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000 --region-linking-mode
 SPECIFIED_REGIONS --regions us-west-1 us-west-2
```

Stopping cross-Region aggregation

> **Note**
>
> The *aggregation Region* is now called the *home Region*. Some Security Hub CSPM API operations still use the older term aggregation Region.

If you don't want AWS Security Hub CSPM to aggregate data, you can delete your finding aggregator. Alternatively, you can keep your finding aggregator but not link any AWS Regions to the home Region by updating the existing aggregator to the `NO_REGIONS` linking mode.

To change your home Region, you must delete your current finding aggregator and create a new one.

When you delete your finding aggregator, Security Hub CSPM stops aggregating data. It doesn't remove any existing aggregated data from the home Region.

Deleting the finding aggregator (console)

You can delete your finding aggregator from the current home Region only.

In Regions other than the home Region, the **Finding aggregation** panel on the Security Hub CSPM console displays a message that you must edit the configuration in the home Region. Choose this message to display a link to switch to the home Region.

Security Hub CSPM console

To stop cross-Region aggregation (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Ensure that you're signed in to your current home Region.
3. In the Security Hub CSPM navigation menu, choose **Settings**, then choose **Regions**.
4. Under **Finding aggregation**, choose **Edit**.
5. Under **Aggregation Region**, choose **No aggregation Region**.
6. Choose **Save**.
7. On the confirmation dialog, in the confirmation field, type **`Confirm`**.
8. Choose **Confirm**.

Security Hub CSPM API

Use the `DeleteFindingAggregator` operation of the Security Hub CSPM API. If you're using the AWS CLI, run the `delete-finding-aggregator` command.

To identify the finding aggregator to delete, provide the finding aggregator ARN. To obtain the finding aggregator ARN, use the `ListFindingAggregators` operation or `list-finding-aggregators` command.

The following example deletes the finding aggregator. The command is run from the current home Region, which is US East (N. Virginia). This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$aws securityhub delete-finding-aggregator arn:aws:securityhub:us-east-1:222222222222:finding-aggregator/123e4567-e89b-12d3-a456-426652340000 --region us-east-1
```

Understanding security standards in Security Hub CSPM

In AWS Security Hub CSPM, a *security standard* is a set of requirements that's based on regulatory frameworks, industry best practices, or company policies. For details about the standards that Security Hub CSPM currently supports, including the security controls that apply to each one, see the Standards reference for Security Hub CSPM.

When you enable a standard, Security Hub CSPM automatically enables all the controls that apply to the standard. Security Hub CSPM then runs security checks on the controls, which generates Security Hub CSPM findings. You can disable and later re-enable individual controls as necessary. You can also disable a standard completely. If you disable a standard, Security Hub CSPM stops running security checks on controls that apply to the standard. Findings are no longer generated for the controls.

In addition to findings, Security Hub CSPM generates a security score for each standard that you enable. The score is based on the status of the controls that apply to the standard. If you set an aggregation Region, the security score for a standard reflects the status of the controls across all linked Regions. If you're the Security Hub CSPM administrator for an organization, the score reflects the status of the controls for all the accounts in your organization. For more information, see Calculating security scores.

To review and manage standards, you can use the Security Hub CSPM console or the Security Hub CSPM API. On the console, the **Security standards** page shows all the security standards that Security Hub CSPM currently supports. This includes a description of each standard and the current status of the standard. If you enable a standard, you can also use this page to access additional details for the standard. For example, you can review:

- The current security score for the standard.
- Aggregated statistics for controls that apply to the standard.
- A list of controls that apply to the standard and are currently enabled, including the compliance status of each one.
- A list of controls that apply to the standard but are currently disabled.

For deeper analysis, you can filter and sort the data, and drill down to review the details of individual controls that apply to the standard.

You can enable standards individually for a single account and AWS Region. However, to save time and reduce configuration drift in multi-account and multi-Region environments, we recommend

using central configuration to enable and manage standards. With central configuration, the delegated Security Hub CSPM administrator can create policies that specify how to configure a standard across multiple accounts and Regions.

Topics

Standards reference for Security Hub CSPM

In AWS Security Hub CSPM, a *security standard* is a set of requirements that's based on regulatory frameworks, industry best practices, or company policies. Security Hub CSPM maps these requirements to controls, and runs security checks on the controls to assess whether the requirements of a standard are being met. Each standard includes multiple controls.

Security Hub CSPM currently supports the following standards:

- **AWS Foundational Security Best Practices** – Developed by AWS and industry professionals, this standard is a compilation of security best practices for organizations, regardless of sector or size. It provides a set of controls that detect when your AWS accounts and resources deviate from security best practices. It also provides prescriptive guidance about how to improve and maintain your security posture.
- **AWS Resource Tagging** – Developed by Security Hub CSPM, this standard can help you determine whether your AWS resources have tags. A *tag* is a key-value pair that acts as metadata for an AWS resource. Tags can help you identify, categorize, manage, and search for AWS resources. For example, you can use tags to categorize resources by purpose, owner, or environment.
- **CIS AWS Foundations Benchmark** – Developed by the Center for Internet Security (CIS), this standard provides secure configuration guidelines for AWS. It specifies a set of security configuration guidelines and best practices for a subset of AWS services and resources, with an emphasis on foundational, testable, and architecture agnostic settings. The guidelines include clear, step-by-step implementation and assessment procedures.

- **NIST SP 800-53 Revision 5** – This standard aligns with National Institute of Standards and Technology (NIST) requirements for protecting the confidentiality, integrity, and availability of information systems and critical resources. The associated framework generally applies to U.S. federal agencies or organizations that work with U.S. federal agencies or information systems. However, private organizations can also use the requirements as a guiding framework.
- **NIST SP 800-171 Revision 2** – This standard aligns with NIST security recommendations and requirements for protecting the confidentiality of Controlled Unclassified Information (CUI) in systems and organizations that aren't part of the U.S. federal government. *CUI* is information that doesn't meet government criteria for classification, but is considered sensitive and is created or possessed by the U.S. federal government or other entities on behalf of the U.S. federal government.
- **PCI DSS** – This standard aligns with the Payment Card Industry Data Security Standard (PCI DSS) compliance framework defined by the PCI Security Standards Council (SSC). The framework provides a set of rules and guidelines for safely handling credit and debit card information. The framework generally applies to organizations that store, process, or transmit cardholder data.
- **Service-managed standard, AWS Control Tower** – This standard helps you configure the proactive controls provided by AWS Control Tower alongside the detective controls provided by Security Hub CSPM. AWS Control Tower offers a straightforward way to set up and govern an AWS multi-account environment, following prescriptive best practices. By enabling both proactive and detective controls for your AWS environment, you can enhance your security posture at different development stages.

Security Hub CSPM standards and controls don't guarantee compliance with any regulatory frameworks or audits. Instead, they provide a way to evaluate and monitor the state of your AWS accounts and resources. We recommend enabling each standard that's relevant to your business needs, industry, or use case.

Individual controls can apply to more than one standard. If you enable multiple standards, we recommend that you also enable consolidated control findings. If you do this, Security Hub CSPM generates a single finding for each control, even if the control applies to more than one standard. If you don't turn on consolidated control findings, Security Hub CSPM generates a separate finding for each enabled standard that a control applies to. For example, if you enable two standards and a control applies to both of them, you receive two separate findings for the control, one for each standard. If you enable consolidated control findings, you receive only one finding for the control. For more information, see Consolidated control findings.

Detailed reference by standard

- AWS Foundational Security Best Practices standard in Security Hub CSPM
- AWS Resource Tagging standard in Security Hub CSPM
- CIS AWS Foundations Benchmark in Security Hub CSPM
- NIST SP 800-53 Revision 5 in Security Hub CSPM
- NIST SP 800-171 Revision 2 in Security Hub CSPM
- PCI DSS in Security Hub CSPM
- Service-managed standards in Security Hub CSPM

AWS Foundational Security Best Practices standard in Security Hub CSPM

Developed by AWS and industry professionals, the AWS Foundational Security Best Practices (FSBP) standard is a compilation of security best practices for organizations, regardless of organization sector or size. It provides a set of controls that detect when AWS accounts and resources deviate from security best practices. It also provides prescriptive guidance about how to improve and maintain your organization's security posture.

In AWS Security Hub CSPM, the AWS Foundational Security Best Practices standard includes controls that continuously evaluate your AWS accounts and workloads, and help you identify areas that deviate from security best practices. The controls include security best practices for resources from multiple AWS services. Each control is assigned a category that reflects the security function that the control applies to. For a list of categories and additional details, see Control categories.

Controls that apply to the standard

The following list specifies which AWS Security Hub CSPM controls apply to the AWS Foundational Security Best Practices standard (v1.0.0). To review the details of a control, choose the control.

[Account.1] Security contact information should be provided for an AWS account

[ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period

[ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits

[APIGateway.1] API Gateway REST and WebSocket API execution logging should be enabled

[APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication

[APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled

[APIGateway.4] API Gateway should be associated with a WAF Web ACL

[APIGateway.5] API Gateway REST API cache data should be encrypted at rest

[APIGateway.8] API Gateway routes should specify an authorization type

[APIGateway.9] Access logging should be configured for API Gateway V2 Stages

[AppSync.1] AWS AppSync API caches should be encrypted at rest

[AppSync.2] AWS AppSync should have field-level logging enabled

[AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys

[AppSync.6] AWS AppSync API caches should be encrypted in transit

[Athena.4] Athena workgroups should have logging enabled

[AutoScaling.1] Auto Scaling groups associated with a load balancer should use ELB health checks

[AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones

[AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)

[Autoscaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses

[AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones

[AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates

[Backup.1] AWS Backup recovery points should be encrypted at rest

[CloudFront.1] CloudFront distributions should have a default root object configured

[CloudFront.3] CloudFront distributions should require encryption in transit

[CloudFront.4] CloudFront distributions should have origin failover configured

[CloudFront.5] CloudFront distributions should have logging enabled

[CloudFront.6] CloudFront distributions should have WAF enabled

[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates

[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests

[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins

[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins

[CloudFront.12] CloudFront distributions should not point to non-existent S3 origins

[CloudFront.13] CloudFront distributions should use origin access control

[CloudFront.15] CloudFront distributions should use the recommended TLS security policy

[CloudFront.16] CloudFront distributions should use origin access control for Lambda function URL origins

[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events

[CloudTrail.2] CloudTrail should have encryption at-rest enabled

[CloudTrail.4] CloudTrail log file validation should be enabled

[CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs

[CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials

[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials

[CodeBuild.3] CodeBuild S3 logs should be encrypted

[CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration

[CodeBuild.7] CodeBuild report group exports should be encrypted at rest

[Cognito.2] Cognito identity pools should not allow unauthenticated identities

[Config.1] AWS Config should be enabled and use the service-linked role for resource recording

[Connect.2] Amazon Connect instances should have CloudWatch logging enabled

[DataFirehose.1] Firehose delivery streams should be encrypted at rest

[DataSync.1] DataSync tasks should have logging enabled

[DMS.1] Database Migration Service replication instances should not be public

[DMS.6] DMS replication instances should have automatic minor version upgrade enabled

[DMS.7] DMS replication tasks for the target database should have logging enabled

[DMS.8] DMS replication tasks for the source database should have logging enabled

[DMS.9] DMS endpoints should use SSL

[DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled

[DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled

[DMS.12] DMS endpoints for Redis OSS should have TLS enabled

[DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest

[DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period

[DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public

[DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs

[DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled

[DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit

[DynamoDB.1] DynamoDB tables should automatically scale capacity with demand

[DynamoDB.2] DynamoDB tables should have point-in-time recovery enabled

[DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest

[DynamoDB.6] DynamoDB tables should have deletion protection enabled

[DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit

[EC2.1] Amazon EBS snapshots should not be publicly restorable

[EC2.2] VPC default security groups should not allow inbound or outbound traffic

[EC2.3] Attached Amazon EBS volumes should be encrypted at-rest

[EC2.4] Stopped EC2 instances should be removed after a specified time period

[EC2.6] VPC flow logging should be enabled in all VPCs

[EC2.7] EBS default encryption should be enabled

[EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)

[EC2.9] Amazon EC2 instances should not have a public IPv4 address

[EC2.10] Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service

[EC2.15] Amazon EC2 subnets should not automatically assign public IP addresses

[EC2.16] Unused Network Access Control Lists should be removed

[EC2.17] Amazon EC2 instances should not use multiple ENIs

[EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports

[EC2.19] Security groups should not allow unrestricted access to ports with high risk

[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up

[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389

[EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests

[EC2.24] Amazon EC2 paravirtual instance types should not be used

[EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces

[EC2.51] EC2 Client VPN endpoints should have client connection logging enabled

[EC2.55] VPCs should be configured with an interface endpoint for ECR API

[EC2.56] VPCs should be configured with an interface endpoint for Docker Registry

[EC2.57] VPCs should be configured with an interface endpoint for Systems Manager

[EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts

[EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager

[EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)

[EC2.171] EC2 VPN connections should have logging enabled

[EC2.172] EC2 VPC Block Public Access settings should block internet gateway traffic

[EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes

[EC2.180] EC2 network interfaces should have source/destination checking enabled

[ECR.1] ECR private repositories should have image scanning configured

[ECR.2] ECR private repositories should have tag immutability configured

[ECR.3] ECR repositories should have at least one lifecycle policy configured

[ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions

[ECS.2] ECS services should not have public IP addresses assigned to them automatically

[ECS.3] ECS task definitions should not share the host's process namespace

[ECS.4] ECS containers should run as non-privileged

[ECS.5] ECS containers should be limited to read-only access to root filesystems

[ECS.8] Secrets should not be passed as container environment variables

[ECS.9] ECS task definitions should have a logging configuration

[ECS.10] ECS Fargate services should run on the latest Fargate platform version

[ECS.12] ECS clusters should use Container Insights

[ECS.16] ECS task sets should not automatically assign public IP addresses

[EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS

[EFS.2] Amazon EFS volumes should be in backup plans

[EFS.3] EFS access points should enforce a root directory

[EFS.4] EFS access points should enforce a user identity

[EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch

[EFS.7] EFS file systems should have automatic backups enabled

[EFS.8] EFS file systems should be encrypted at rest

[EKS.1] EKS cluster endpoints should not be publicly accessible

[EKS.2] EKS clusters should run on a supported Kubernetes version

[EKS.3] EKS clusters should use encrypted Kubernetes secrets

[EKS.8] EKS clusters should have audit logging enabled

[ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled

[ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled

[ElastiCache.3] ElastiCache replication groups should have automatic failover enabled

[ElastiCache.4] ElastiCache replication groups should be encrypted at rest

[ElastiCache.5] ElastiCache replication groups should be encrypted in transit

[ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled

[ElastiCache.7] ElastiCache clusters should not use the default subnet group

[ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled

[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled

[ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch

[ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS

[ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager

[ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination

[ELB.4] Application Load Balancer should be configured to drop invalid http headers

[ELB.5] Application and Classic Load Balancers logging should be enabled

[ELB.6] Application, Gateway, and Network Load Balancers should have deletion protection enabled

[ELB.7] Classic Load Balancers should have connection draining enabled

[ELB.8] Classic Load Balancers with SSL listeners should use a predefined security policy that has strong AWS Configuration

[ELB.9] Classic Load Balancers should have cross-zone load balancing enabled

[ELB.10] Classic Load Balancer should span multiple Availability Zones

[ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode

[ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones

[ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode

[ELB.17] Application and Network Load Balancers with listeners should use recommended security policies

[ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit

[EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses

[EMR.2] Amazon EMR block public access setting should be enabled

[EMR.3] Amazon EMR security configurations should be encrypted at rest

[EMR.4] Amazon EMR security configurations should be encrypted in transit

[ES.1] Elasticsearch domains should have encryption at-rest enabled

[ES.2] Elasticsearch domains should not be publicly accessible

[ES.3] Elasticsearch domains should encrypt data sent between nodes

[ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled

[ES.5] Elasticsearch domains should have audit logging enabled

[ES.6] Elasticsearch domains should have at least three data nodes

[ES.7] Elasticsearch domains should be configured with at least three dedicated master nodes

[ES.8] Connections to Elasticsearch domains should be encrypted using the latest TLS security policy

[EventBridge.3] EventBridge custom event buses should have a resource-based policy attached

[FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes

[FSx.2] FSx for Lustre file systems should be configured to copy tags to backups

[FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment

[FSx.4] FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment

[FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment

[Glue.3] AWS Glue machine learning transforms should be encrypted at rest

[Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue

[GuardDuty.1] GuardDuty should be enabled

[GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled

[GuardDuty.6] GuardDuty Lambda Protection should be enabled

[GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled

[GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled

[GuardDuty.9] GuardDuty RDS Protection should be enabled

[GuardDuty.10] GuardDuty S3 Protection should be enabled

[GuardDuty.11] GuardDuty Runtime Monitoring should be enabled

[GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled

[GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled

[IAM.1] IAM policies should not allow full "*" administrative privileges

[IAM.2] IAM users should not have IAM policies attached

[IAM.3] IAM users' access keys should be rotated every 90 days or less

[IAM.4] IAM root user access key should not exist

[IAM.5] MFA should be enabled for all IAM users that have a console password

[IAM.6] Hardware MFA should be enabled for the root user

[IAM.7] Password policies for IAM users should have strong configurations

[IAM.8] Unused IAM user credentials should be removed

[IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services

[Inspector.1] Amazon Inspector EC2 scanning should be enabled

[Inspector.2] Amazon Inspector ECR scanning should be enabled

[Inspector.3] Amazon Inspector Lambda code scanning should be enabled

[Inspector.4] Amazon Inspector Lambda standard scanning should be enabled

[Kinesis.1] Kinesis streams should be encrypted at rest

[Kinesis.3] Kinesis streams should have an adequate data retention period

[KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys

[KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys

[KMS.3] AWS KMS keys should not be deleted unintentionally

[KMS.5] KMS keys should not be publicly accessible

[Lambda.1] Lambda function policies should prohibit public access

[Lambda.2] Lambda functions should use supported runtimes

[Lambda.5] VPC Lambda functions should operate in multiple Availability Zones

[Macie.1] Amazon Macie should be enabled

[Macie.2] Macie automated sensitive data discovery should be enabled

[MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch

[MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled

[MSK.1] MSK clusters should be encrypted in transit among broker nodes

[MSK.3] MSK Connect connectors should be encrypted in transit

[MSK.4] MSK clusters should have public access disabled

[MSK.5] MSK connectors should have logging enabled

[MSK.6] MSK clusters should disable unauthenticated access

[Neptune.1] Neptune DB clusters should be encrypted at rest

[Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs

[Neptune.3] Neptune DB cluster snapshots should not be public

[Neptune.4] Neptune DB clusters should have deletion protection enabled

[Neptune.5] Neptune DB clusters should have automated backups enabled

[Neptune.6] Neptune DB cluster snapshots should be encrypted at rest

[Neptune.7] Neptune DB clusters should have IAM database authentication enabled

[Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots

[NetworkFirewall.2] Network Firewall logging should be enabled

[NetworkFirewall.3] Network Firewall policies should have at least one rule group associated

[NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets

[NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets

[NetworkFirewall.6] Stateless Network Firewall rule group should not be empty

[NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled

[NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled

[Opensearch.1] OpenSearch domains should have encryption at rest enabled

[Opensearch.2] OpenSearch domains should not be publicly accessible

[Opensearch.3] OpenSearch domains should encrypt data sent between nodes

[Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled

[Opensearch.5] OpenSearch domains should have audit logging enabled

[Opensearch.6] OpenSearch domains should have at least three data nodes

[Opensearch.7] OpenSearch domains should have fine-grained access control enabled

[Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy

[Opensearch.10] OpenSearch domains should have the latest software update installed

[PCA.1] AWS Private CA root certificate authority should be disabled

[Route53.2] Route 53 public hosted zones should log DNS queries

[RDS.1] RDS snapshot should be private

[RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration

[RDS.3] RDS DB instances should have encryption at-rest enabled

[RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest

[RDS.5] RDS DB instances should be configured with multiple Availability Zones

[RDS.6] Enhanced monitoring should be configured for RDS DB instances

[RDS.7] RDS clusters should have deletion protection enabled

[RDS.8] RDS DB instances should have deletion protection enabled

[RDS.9] RDS DB instances should publish logs to CloudWatch Logs

[RDS.10] IAM authentication should be configured for RDS instances

[RDS.11] RDS instances should have automatic backups enabled

[RDS.12] IAM authentication should be configured for RDS clusters

[RDS.13] RDS automatic minor version upgrades should be enabled

[RDS.14] Amazon Aurora clusters should have backtracking enabled

[RDS.15] RDS DB clusters should be configured for multiple Availability Zones

[RDS.16] Aurora DB clusters should be configured to copy tags to DB snapshots

[RDS.17] RDS DB instances should be configured to copy tags to snapshots

[RDS.19] Existing RDS event notification subscriptions should be configured for critical cluster events

[RDS.20] Existing RDS event notification subscriptions should be configured for critical database instance events

[RDS.21] An RDS event notifications subscription should be configured for critical database parameter group events

[RDS.22] An RDS event notifications subscription should be configured for critical database security group events

[RDS.23] RDS instances should not use a database engine default port

[RDS.24] RDS Database clusters should use a custom administrator username

[RDS.25] RDS database instances should use a custom administrator username

[RDS.27] RDS DB clusters should be encrypted at rest

[RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs

[RDS.35] RDS DB clusters should have automatic minor version upgrade enabled

[RDS.36] RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs

[RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs

[RDS.40] RDS for SQL Server DB instances should publish logs to CloudWatch Logs

[RDS.41] RDS for SQL Server DB instances should be encrypted in transit

[RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs

[RDS.44] RDS for MariaDB DB instances should be encrypted in transit

[RDS.45] Aurora MySQL DB clusters should have audit logging enabled

[RDS.46] RDS DB instances should not be deployed in public subnets with routes to internet gateways

[Redshift.1] Amazon Redshift clusters should prohibit public access

[Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit

[Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled

[Redshift.4] Amazon Redshift clusters should have audit logging enabled

[Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled

[Redshift.7] Redshift clusters should use enhanced VPC routing

[Redshift.8] Amazon Redshift clusters should not use the default Admin username

[Redshift.9] Redshift clusters should not use the default database name

[Redshift.10] Redshift clusters should be encrypted at rest

[Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins

[Redshift.18] Redshift clusters should have Multi-AZ deployments enabled

[RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing

[RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL

[RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access

[RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username

[RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs

[RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name

[S3.1] S3 general purpose buckets should have block public access settings enabled

[S3.2] S3 general purpose buckets should block public read access

[S3.3] S3 general purpose buckets should block public write access

[S3.5] S3 general purpose buckets should require requests to use SSL

[S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts

[S3.8] S3 general purpose buckets should block public access

[S3.9] S3 general purpose buckets should have server access logging enabled

[S3.12] ACLs should not be used to manage user access to S3 general purpose buckets

[S3.13] S3 general purpose buckets should have Lifecycle configurations

[S3.19] S3 access points should have block public access settings enabled

[S3.24] S3 Multi-Region Access Points should have block public access settings enabled

[S3.25] S3 directory buckets should have lifecycle configurations

[SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access

[SageMaker.2] SageMaker notebook instances should be launched in a custom VPC

[SageMaker.3] Users should not have root access to SageMaker notebook instances

[SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1

[SageMaker.5] SageMaker models should have network isolation enabled

[SageMaker.8] SageMaker notebook instances should run on supported platforms

[SecretsManager.1] Secrets Manager secrets should have automatic rotation enabled

[SecretsManager.2] Secrets Manager secrets configured with automatic rotation should rotate successfully

[SecretsManager.3] Remove unused Secrets Manager secrets

[SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days

[ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only

[SNS.4] SNS topic access policies should not allow public access

[SQS.1] Amazon SQS queues should be encrypted at rest

[SQS.3] SQS queue access policies should not allow public access

[SSM.1] Amazon EC2 instances should be managed by AWS Systems Manager

[SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation

[SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT

[SSM.4] SSM documents should not be public

[SSM.6] SSM Automation should have CloudWatch logging enabled

[SSM.7] SSM documents should have the block public sharing setting enabled

[StepFunctions.1] Step Functions state machines should have logging turned on

[Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection

[Transfer.3] Transfer Family connectors should have logging enabled

[WAF.1] AWS WAF Classic Global Web ACL logging should be enabled

[WAF.2] AWS WAF Classic Regional rules should have at least one condition

[WAF.3] AWS WAF Classic Regional rule groups should have at least one rule

[WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group

[WAF.6] AWS WAF Classic global rules should have at least one condition

[WAF.7] AWS WAF Classic global rule groups should have at least one rule

[WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

[WAF.10] AWS WAF web ACLs should have at least one rule or rule group

[WAF.12] AWS WAF rules should have CloudWatch metrics enabled

[WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest

[WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

AWS Resource Tagging standard in Security Hub CSPM

The AWS Resource Tagging standard, developed by AWS Security Hub CSPM, helps you determine whether your AWS resources are missing tags. *Tags* are key-value pairs that act as metadata for organizing AWS resources. With most AWS resources, you have the option of adding tags to a resource when you create the resource or after you create the resource. Examples of resources include Amazon CloudFront distributions, Amazon Elastic Compute Cloud (Amazon EC2) instances, and secrets in AWS Secrets Manager. Tags can help you manage, identify, organize, search for, and filter AWS resources.

Each tag has two parts:

- A tag key—for example, `CostCenter`, `Environment`, or `Project`. Tag keys are case sensitive.
- A tag value—for example, `111122223333` or `Production`. Like tag keys, tag values are case sensitive.

You can use tags to categorize resources by purpose, owner, environment, or other criteria. For information about adding tags to AWS resources, see the Tagging AWS Resources and Tag Editor User Guide.

For each control that applies to the AWS Resource Tagging standard in Security Hub CSPM, you can optionally use the supported parameter to specify tag keys that you want the control to check for. If you don't specify any tag keys, the control checks only for the existence of at least one tag key, and fails if a resource doesn't have any tag keys.

Before you enable the AWS Resource Tagging standard, it's important to enable and configure resource recording in AWS Config. When you configure resource recording, also be sure to enable it for all the types of AWS resources that are checked by controls that apply to the standard. Otherwise, Security Hub CSPM might not be able to evaluate the appropriate resources, and generate accurate findings for controls that apply to the standard. For more information, including a list of the types of resources to record, see Required AWS Config resources for control findings.

After you enable the AWS Resource Tagging standard, you begin receiving findings for controls that apply to the standard. Note that it can take up to 18 hours for Security Hub CSPM to generate findings for controls that use the same AWS Config service-linked rule as controls that apply to other enabled standards. For more information, see Schedule for running security checks.

The AWS Resource Tagging standard has the following Amazon Resource Name (ARN): `arn:aws:securityhub:region::standards/aws-resource-tagging-standard/v/1.0.0`, where *region* is the Region code for the applicable AWS Region. You can also use the GetEnabledStandards operation of the Security Hub CSPM API to retrieve the ARN of a standard that's currently enabled.

> **Note**
>
> The AWS Resource Tagging standard isn't available in the Asia Pacific (Taipei) Region.

Controls that apply to the standard

The following list specifies which AWS Security Hub CSPM controls apply to the AWS Resource Tagging standard (v1.0.0). To review the details of a control, choose the control.

- [ACM.3] ACM certificates should be tagged
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged

- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.4] AWS AppSync GraphQL APIs should be tagged
- [Athena.2] Athena data catalogs should be tagged
- [Athena.3] Athena workgroups should be tagged
- [AutoScaling.10] EC2 Auto Scaling groups should be tagged
- [Backup.2] AWS Backup recovery points should be tagged
- [Backup.3] AWS Backup vaults should be tagged
- [Backup.4] AWS Backup report plans should be tagged
- [Backup.5] AWS Backup backup plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.2] Batch scheduling policies should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFormation.2] CloudFormation stacks should be tagged
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudTrail.9] CloudTrail trails should be tagged
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [DataSync.2] DataSync tasks should be tagged
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DynamoDB.5] DynamoDB tables should be tagged
- [EC2.33] EC2 transit gateway attachments should be tagged
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.35] EC2 network interfaces should be tagged

- [EC2.36] EC2 customer gateways should be tagged
- [EC2.37] EC2 Elastic IP addresses should be tagged
- [EC2.38] EC2 instances should be tagged
- [EC2.39] EC2 internet gateways should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.41] EC2 network ACLs should be tagged
- [EC2.42] EC2 route tables should be tagged
- [EC2.43] EC2 security groups should be tagged
- [EC2.44] EC2 subnets should be tagged
- [EC2.45] EC2 volumes should be tagged
- [EC2.46] Amazon VPCs should be tagged
- [EC2.47] Amazon VPC endpoint services should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged
- [EC2.49] Amazon VPC peering connections should be tagged
- [EC2.50] EC2 VPN gateways should be tagged
- [EC2.52] EC2 transit gateways should be tagged
- [EC2.174] EC2 DHCP option sets should be tagged
- [EC2.175] EC2 launch templates should be tagged
- [EC2.176] EC2 prefix lists should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.178] EC2 traffic mirror filters should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [ECR.4] ECR public repositories should be tagged
- [ECS.13] ECS services should be tagged
- [ECS.14] ECS clusters should be tagged
- [ECS.15] ECS task definitions should be tagged
- [EFS.5] EFS access points should be tagged
- [EKS.6] EKS clusters should be tagged
- [EKS.7] EKS identity provider configurations should be tagged
- [ES.9] Elasticsearch domains should be tagged

- [EventBridge.2] EventBridge event buses should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.1] AWS Glue jobs should be tagged
- [GuardDuty.2] GuardDuty filters should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [GuardDuty.4] GuardDuty detectors should be tagged
- [IAM.23] IAM Access Analyzer analyzers should be tagged
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged

- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.2] Kinesis streams should be tagged
- [Lambda.6] Lambda functions should be tagged
- [MQ.4] Amazon MQ brokers should be tagged
- [NetworkFirewall.7] Network Firewall firewalls should be tagged
- [NetworkFirewall.8] Network Firewall firewall policies should be tagged
- [Opensearch.9] OpenSearch domains should be tagged
- [PCA.2] AWS Private CA certificate authorities should be tagged
- [RDS.28] RDS DB clusters should be tagged
- [RDS.29] RDS DB cluster snapshots should be tagged
- [RDS.30] RDS DB instances should be tagged
- [RDS.31] RDS DB security groups should be tagged
- [RDS.32] RDS DB snapshots should be tagged
- [RDS.33] RDS DB subnet groups should be tagged
- [Redshift.11] Redshift clusters should be tagged
- [Redshift.12] Redshift event notification subscriptions should be tagged
- [Redshift.13] Redshift cluster snapshots should be tagged
- [Redshift.14] Redshift cluster subnet groups should be tagged
- [Redshift.17] Redshift cluster parameter groups should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SecretsManager.5] Secrets Manager secrets should be tagged

- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [SNS.3] SNS topics should be tagged
- [SQS.2] SQS queues should be tagged
- [SSM.5] SSM documents should be tagged
- [StepFunctions.2] Step Functions activities should be tagged
- [Transfer.1] AWS Transfer Family workflows should be tagged
- [Transfer.4] Transfer Family agreements should be tagged
- [Transfer.5] Transfer Family certificates should be tagged
- [Transfer.6] Transfer Family connectors should be tagged
- [Transfer.7] Transfer Family profiles should be tagged

CIS AWS Foundations Benchmark in Security Hub CSPM

The Center for Internet Security (CIS) AWS Foundations Benchmark serves as a set of security configuration best practices for AWS. These industry-accepted best practices provide you with clear, step-by-step implementation and assessment procedures. Ranging from operating systems to cloud services and network devices, the controls in this benchmark help you protect the specific systems that your organization uses.

AWS Security Hub CSPM supports CIS AWS Foundations Benchmark versions 3.0.0, 1.4.0, and 1.2.0. This page lists the security controls that each version supports. It also provides a comparison of the versions.

CIS AWS Foundations Benchmark version 3.0.0

Security Hub CSPM supports version 3.0.0 (v3.0.0) of the CIS AWS Foundations Benchmark. Security Hub CSPM has satisfied the requirements of CIS Security Software Certification and has been awarded CIS Security Software Certification for the following CIS Benchmarks:

- CIS Benchmark for CIS AWS Foundations Benchmark, v3.0.0, Level 1
- CIS Benchmark for CIS AWS Foundations Benchmark, v3.0.0, Level 2

Controls that apply to CIS AWS Foundations Benchmark version 3.0.0

[Account.1] Security contact information should be provided for an AWS account

[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events

[CloudTrail.2] CloudTrail should have encryption at-rest enabled

[CloudTrail.4] CloudTrail log file validation should be enabled

[CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket

[Config.1] AWS Config should be enabled and use the service-linked role for resource recording

[EC2.2] VPC default security groups should not allow inbound or outbound traffic

[EC2.6] VPC flow logging should be enabled in all VPCs

[EC2.7] EBS default encryption should be enabled

[EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)

[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389

[EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports

[EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports

[EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS

[IAM.2] IAM users should not have IAM policies attached

[IAM.3] IAM users' access keys should be rotated every 90 days or less

[IAM.4] IAM root user access key should not exist

[IAM.5] MFA should be enabled for all IAM users that have a console password

[IAM.6] Hardware MFA should be enabled for the root user

[IAM.9] MFA should be enabled for the root user

[IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater

[IAM.16] Ensure IAM password policy prevents password reuse

[IAM.18] Ensure a support role has been created to manage incidents with AWS Support

[IAM.22] IAM user credentials unused for 45 days should be removed

[IAM.26] Expired SSL/TLS certificates managed in IAM should be removed

[IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached

[IAM.28] IAM Access Analyzer external access analyzer should be enabled

[KMS.4] AWS KMS key rotation should be enabled

[RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration

[RDS.3] RDS DB instances should have encryption at-rest enabled

[RDS.13] RDS automatic minor version upgrades should be enabled

[S3.1] S3 general purpose buckets should have block public access settings enabled

[S3.5] S3 general purpose buckets should require requests to use SSL

[S3.8] S3 general purpose buckets should block public access

[S3.20] S3 general purpose buckets should have MFA delete enabled

[S3.22] S3 general purpose buckets should log object-level write events

[S3.23] S3 general purpose buckets should log object-level read events

CIS AWS Foundations Benchmark version 1.4.0

Security Hub CSPM supports version 1.4.0 (v1.4.0) of the CIS AWS Foundations Benchmark.

Controls that apply to CIS AWS Foundations Benchmark version 1.4.0

[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events

[CloudTrail.2] CloudTrail should have encryption at-rest enabled

[CloudTrail.4] CloudTrail log file validation should be enabled

[CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs

[CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible

[CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket

[CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user

[CloudWatch.4] Ensure a log metric filter and alarm exist for IAM policy changes

[CloudWatch.5] Ensure a log metric filter and alarm exist for CloudTrail configuration changes

[CloudWatch.6] Ensure a log metric filter and alarm exist for AWS Management Console authentication failures

[CloudWatch.7] Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys

[CloudWatch.8] Ensure a log metric filter and alarm exist for S3 bucket policy changes

[CloudWatch.9] Ensure a log metric filter and alarm exist for AWS Config configuration changes

[CloudWatch.10] Ensure a log metric filter and alarm exist for security group changes

[CloudWatch.11] Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)

[CloudWatch.12] Ensure a log metric filter and alarm exist for changes to network gateways

[CloudWatch.13] Ensure a log metric filter and alarm exist for route table changes

[CloudWatch.14] Ensure a log metric filter and alarm exist for VPC changes

[Config.1] AWS Config should be enabled and use the service-linked role for resource recording

[EC2.2] VPC default security groups should not allow inbound or outbound traffic

[EC2.6] VPC flow logging should be enabled in all VPCs

[EC2.7] EBS default encryption should be enabled

[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389

[IAM.1] IAM policies should not allow full "*" administrative privileges

[IAM.3] IAM users' access keys should be rotated every 90 days or less

[IAM.4] IAM root user access key should not exist

[IAM.5] MFA should be enabled for all IAM users that have a console password

[IAM.6] Hardware MFA should be enabled for the root user

[IAM.9] MFA should be enabled for the root user

[IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater

[IAM.16] Ensure IAM password policy prevents password reuse

[IAM.18] Ensure a support role has been created to manage incidents with AWS Support

[IAM.22] IAM user credentials unused for 45 days should be removed

[KMS.4] AWS KMS key rotation should be enabled

[RDS.3] RDS DB instances should have encryption at-rest enabled

[S3.1] S3 general purpose buckets should have block public access settings enabled

[S3.5] S3 general purpose buckets should require requests to use SSL

[S3.8] S3 general purpose buckets should block public access

[S3.20] S3 general purpose buckets should have MFA delete enabled

CIS AWS Foundations Benchmark version 1.2.0

Security Hub CSPM supports version 1.2.0 (v1.2.0) of the CIS AWS Foundations Benchmark. Security Hub CSPM has satisfied the requirements of CIS Security Software Certification and has been awarded CIS Security Software Certification for the following CIS Benchmarks:

- CIS Benchmark for CIS AWS Foundations Benchmark, v1.2.0, Level 1
- CIS Benchmark for CIS AWS Foundations Benchmark, v1.2.0, Level 2

Controls that apply to CIS AWS Foundations Benchmark version 1.2.0

[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events

[CloudTrail.2] CloudTrail should have encryption at-rest enabled

[CloudTrail.4] CloudTrail log file validation should be enabled

[CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs

[CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible

[CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket

[CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user

[CloudWatch.2] Ensure a log metric filter and alarm exist for unauthorized API calls

[CloudWatch.3] Ensure a log metric filter and alarm exist for Management Console sign-in without MFA

[CloudWatch.4] Ensure a log metric filter and alarm exist for IAM policy changes

[CloudWatch.5] Ensure a log metric filter and alarm exist for CloudTrail configuration changes

[CloudWatch.6] Ensure a log metric filter and alarm exist for AWS Management Console authentication failures

[CloudWatch.7] Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys

[CloudWatch.8] Ensure a log metric filter and alarm exist for S3 bucket policy changes

[CloudWatch.9] Ensure a log metric filter and alarm exist for AWS Config configuration changes

[CloudWatch.10] Ensure a log metric filter and alarm exist for security group changes

[CloudWatch.11] Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)

[CloudWatch.12] Ensure a log metric filter and alarm exist for changes to network gateways

[CloudWatch.13] Ensure a log metric filter and alarm exist for route table changes

[CloudWatch.14] Ensure a log metric filter and alarm exist for VPC changes

[Config.1] AWS Config should be enabled and use the service-linked role for resource recording

[EC2.2] VPC default security groups should not allow inbound or outbound traffic

[EC2.6] VPC flow logging should be enabled in all VPCs

[EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22

[EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389

[IAM.1] IAM policies should not allow full "*" administrative privileges

[IAM.2] IAM users should not have IAM policies attached

[IAM.3] IAM users' access keys should be rotated every 90 days or less

[IAM.4] IAM root user access key should not exist

[IAM.5] MFA should be enabled for all IAM users that have a console password

[IAM.6] Hardware MFA should be enabled for the root user

[IAM.8] Unused IAM user credentials should be removed

[IAM.9] MFA should be enabled for the root user

[IAM.11] Ensure IAM password policy requires at least one uppercase letter

[IAM.12] Ensure IAM password policy requires at least one lowercase letter

[IAM.13] Ensure IAM password policy requires at least one symbol

[IAM.14] Ensure IAM password policy requires at least one number

[IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater

[IAM.16] Ensure IAM password policy prevents password reuse

[IAM.17] Ensure IAM password policy expires passwords within 90 days or less

[IAM.18] Ensure a support role has been created to manage incidents with AWS Support

[KMS.4] AWS KMS key rotation should be enabled

Version comparison for CIS AWS Foundations Benchmark

This section summarizes the differences between specific versions of the Center for Internet Security (CIS) AWS Foundations Benchmark—v3.0.0, v1.4.0, and v1.2.0. AWS Security Hub CSPM supports each of these versions of the CIS AWS Foundations Benchmark. However, we recommend using v3.0.0 to stay current with security best practices. You can have multiple versions of the

standard enabled at the same time. For information about enabling standards, see Enabling a security standard. If you want to upgrade to v3.0.0, enable it before you disable an older version. This prevents gaps in your security checks. If you use the Security Hub CSPM integration with AWS Organizations and want to batch enable v3.0.0 in multiple accounts, we recommend using central configuration.

Mapping of controls to CIS requirements in each version

Understand which controls each version of the CIS AWS Foundations Benchmark supports.

Control ID and title	CIS v3.0.0 requirement	CIS v1.4.0 requirement	CIS v1.2.0 requirement
[Account.1] Security contact information should be provided for an AWS account	1.2	1.2	1.18
[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events	3.1	3.1	2.1
[CloudTrail.2] CloudTrail should have encryption at-rest enabled	3.5	3.7	2.7
[CloudTrail.4] CloudTrail log file validation should be enabled	3.2	3.2	2.2
[CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs	Not supported – CIS removed this requirement	3.4	2.4
[CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible	Not supported – CIS removed this requirement	3.3	2.3
[CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket	3.4	3.6	2.6

Control ID and title	CIS v3.0.0 requirement	CIS v1.4.0 requirement	CIS v1.2.0 requirement
[CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user	Not supported – manual check	4.3	3.3
[CloudWatch.2] Ensure a log metric filter and alarm exist for unauthorized API calls	Not supported – manual check	Not supported – manual check	3.1
[CloudWatch.3] Ensure a log metric filter and alarm exist for Management Console sign-in without MFA	Not supported – manual check	Not supported – manual check	3.2
[CloudWatch.4] Ensure a log metric filter and alarm exist for IAM policy changes	Not supported – manual check	4.4	3.4
[CloudWatch.5] Ensure a log metric filter and alarm exist for CloudTrail configuration changes	Not supported – manual check	4.5	3.5
[CloudWatch.6] Ensure a log metric filter and alarm exist for AWS Management Console authentication failures	Not supported – manual check	4.6	3.6
[CloudWatch.7] Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys	Not supported – manual check	4.7	3.7
[CloudWatch.8] Ensure a log metric filter and alarm exist for S3 bucket policy changes	Not supported – manual check	4.8	3.8

Control ID and title	CIS v3.0.0 requirement	CIS v1.4.0 requirement	CIS v1.2.0 requirement
[CloudWatch.9] Ensure a log metric filter and alarm exist for AWS Config configuration changes	Not supported – manual check	4.9	3.9
[CloudWatch.10] Ensure a log metric filter and alarm exist for security group changes	Not supported – manual check	4.10	3.10
[CloudWatch.11] Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)	Not supported – manual check	4.11	3.11
[CloudWatch.12] Ensure a log metric filter and alarm exist for changes to network gateways	Not supported – manual check	4.12	3.12
[CloudWatch.13] Ensure a log metric filter and alarm exist for route table changes	Not supported – manual check	4.13	3.13
[CloudWatch.14] Ensure a log metric filter and alarm exist for VPC changes	Not supported – manual check	4.14	3.14
[Config.1] AWS Config should be enabled and use the service-linked role for resource recording	3.3	3.5	2.5
[EC2.2] VPC default security groups should not allow inbound or outbound traffic	5.4	5.3	4.3
[EC2.6] VPC flow logging should be enabled in all VPCs	3.7	3.9	2.9

Control ID and title	CIS v3.0.0 requirement	CIS v1.4.0 requirement	CIS v1.2.0 requirement
[EC2.7] EBS default encryption should be enabled	2.2.1	2.2.1	Not supported
[EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)	5.6	Not supported	Not supported
[EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22	Not supported – replaced by requirements 5.2 and 5.3	Not supported – replaced by requirements 5.2 and 5.3	4.1
[EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389	Not supported – replaced by requirements 5.2 and 5.3	Not supported – replaced by requirements 5.2 and 5.3	4.2
[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389	5.1	5.1	Not supported
[EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports	5.2	Not supported	Not supported
[EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports	5.3	Not supported	Not supported
[EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS	2.4.1	Not supported	Not supported
[IAM.1] IAM policies should not allow full "*" administrative privileges	Not supported	1.16	1.22

Control ID and title	CIS v3.0.0 requirement	CIS v1.4.0 requirement	CIS v1.2.0 requirement
[IAM.2] IAM users should not have IAM policies attached	1.15	Not supported	1.16
[IAM.3] IAM users' access keys should be rotated every 90 days or less	1.14	1.14	1.4
[IAM.4] IAM root user access key should not exist	1.4	1.4	1.12
[IAM.5] MFA should be enabled for all IAM users that have a console password	1.10	1.10	1.2
[IAM.6] Hardware MFA should be enabled for the root user	1.6	1.6	1.14
[IAM.8] Unused IAM user credentials should be removed	Not supported – see [IAM.22] IAM user credentials unused for 45 days should be removed instead	Not supported – see [IAM.22] IAM user credentials unused for 45 days should be removed instead	1.3
[IAM.9] MFA should be enabled for the root user	1.5	1.5	1.13
[IAM.11] Ensure IAM password policy requires at least one uppercase letter	Not supported – CIS removed this requirement	Not supported – CIS removed this requirement	1.5
[IAM.12] Ensure IAM password policy requires at least one lowercase letter	Not supported – CIS removed this requirement	Not supported – CIS removed this requirement	1.6

Control ID and title	CIS v3.0.0 requirement	CIS v1.4.0 requirement	CIS v1.2.0 requirement
[IAM.13] Ensure IAM password policy requires at least one symbol	Not supported – CIS removed this requirement	Not supported – CIS removed this requirement	1.7
[IAM.14] Ensure IAM password policy requires at least one number	Not supported – CIS removed this requirement	Not supported – CIS removed this requirement	1.8
[IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater	1.8	1.8	1.9
[IAM.16] Ensure IAM password policy prevents password reuse	1.9	1.9	1.10
[IAM.17] Ensure IAM password policy expires passwords within 90 days or less	Not supported – CIS removed this requirement	Not supported – CIS removed this requirement	1.11
[IAM.18] Ensure a support role has been created to manage incidents with AWS Support	1.17	1.17	1.2
[IAM.20] Avoid the use of the root user	Not supported – CIS removed this requirement	Not supported – CIS removed this requirement	1.1
[IAM.22] IAM user credentials unused for 45 days should be removed	1.12	1.12	Not supported – CIS added this requirement in later versions
[IAM.26] Expired SSL/TLS certificates managed in IAM should be removed	1.19	Not supported – CIS added this requirement in later versions	Not supported – CIS added this requirement in later versions

Control ID and title	CIS v3.0.0 requirement	CIS v1.4.0 requirement	CIS v1.2.0 requirement
[IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached	1.22	Not supported – CIS added this requirement in later versions	Not supported – CIS added this requirement in later versions
[IAM.28] IAM Access Analyzer external access analyzer should be enabled	1.20	Not supported – CIS added this requirement in later versions	Not supported – CIS added this requirement in later versions
[KMS.4] AWS KMS key rotation should be enabled	3.6	3.8	2.8
[Macie.1] Amazon Macie should be enabled	Not supported – manual check	Not supported – manual check	Not supported – manual check
[RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration	2.3.3	Not supported – CIS added this requirement in later versions	Not supported – CIS added this requirement in later versions
[RDS.3] RDS DB instances should have encryption at-rest enabled	2.3.1	2.3.1	Not supported – CIS added this requirement in later versions
[RDS.13] RDS automatic minor version upgrades should be enabled	2.3.2	Not supported – CIS added this requirement in later versions	Not supported – CIS added this requirement in later versions
[S3.1] S3 general purpose buckets should have block public access settings enabled	2.1.4	2.1.5	Not supported – CIS added this requirement in later versions

Control ID and title	CIS v3.0.0 requirement	CIS v1.4.0 requirement	CIS v1.2.0 requirement
[S3.5] S3 general purpose buckets should require requests to use SSL	2.1.1	2.1.2	Not supported – CIS added this requirement in later versions
[S3.8] S3 general purpose buckets should block public access	2.1.4	2.1.5	Not supported – CIS added this requirement in later versions
[S3.20] S3 general purpose buckets should have MFA delete enabled	2.1.2	2.1.3	Not supported – CIS added this requirement in later versions

ARNs for CIS AWS Foundations Benchmarks

When you enable one or more versions of the CIS AWS Foundations Benchmark, you begin receiving findings in the AWS Security Finding Format (ASFF). In ASFF, each version uses the following Amazon Resource Name (ARN):

CIS AWS Foundations Benchmark v3.0.0

```
arn:aws:securityhub:region::standards/cis-aws-foundations-benchmark/
v/3.0.0
```

CIS AWS Foundations Benchmark v1.4.0

```
arn:aws:securityhub:region::standards/cis-aws-foundations-benchmark/
v/1.4.0
```

CIS AWS Foundations Benchmark v1.2.0

```
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0
```

You can use the GetEnabledStandards operation of the Security Hub CSPM API to find the ARN of an enabled standard.

The preceding values are for `StandardsArn`. However, `StandardsSubscriptionArn` refers to the standard subscription resource that Security Hub CSPM creates when you subscribe to a standard by calling BatchEnableStandards in a Region.

> **Note**
>
> When you enable a version of the CIS AWS Foundations Benchmark, it can take up to 18 hours for Security Hub CSPM to generate findings for controls that use the same AWS Config service-linked rule as enabled controls in other enabled standards. For more information about the schedule for generating control findings, see Schedule for running security checks.

Finding fields differ if you turn on consolidated control findings. For information about these differences, see Impact of consolidation on ASFF fields and values. For sample control findings, see Samples of control findings.

CIS requirements that aren't supported in Security Hub CSPM

As noted in the preceding table, Security Hub CSPM doesn't support every CIS requirement in every version of the CIS AWS Foundations Benchmark. Many of the unsupported requirements can be evaluated only by manually reviewing the state of your AWS resources.

NIST SP 800-53 Revision 5 in Security Hub CSPM

NIST Special Publication 800-53 Revision 5 (NIST SP 800-53 Rev. 5) is a cybersecurity and compliance framework developed by the National Institute of Standards and Technology (NIST), an agency that's part of the U.S. Department of Commerce. This compliance framework provides a catalog of security and privacy requirements for protecting the confidentiality, integrity, and availability of information systems and critical resources. U.S. federal government agencies and contractors must comply with these requirements to protect their systems and organizations. Private organizations can also voluntarily use the requirements as a guiding framework for reducing cybersecurity risk. For more information about the framework and its requirements, see NIST SP 800-53 Rev. 5 in the *NIST Computer Security Resource Center*.

AWS Security Hub CSPM provides security controls that support a subset of NIST SP 800-53 Revision 5 requirements. The controls perform automated security checks for certain AWS services and resources. To enable and manage these controls, you can enable the NIST SP 800-53 Revision

5 framework as a standard in Security Hub CSPM. Note that the controls don't support NIST SP 800-53 Revision 5 requirements that require manual checks.

Unlike other frameworks, the NIST SP 800-53 Revision 5 framework isn't prescriptive about how its requirements should be evaluated. Instead, the framework provides guidelines. In Security Hub CSPM, the NIST SP 800-53 Revision 5 standard and controls represent the service's understanding of these guidelines.

Topics

Configuring resource recording for controls that apply to the standard

To optimize coverage and the accuracy of findings, it's important to enable and configure resource recording in AWS Config before you enable the NIST SP 800-53 Revision 5 standard in AWS Security Hub CSPM. When you configure resource recording, also be sure to enable it for all the types of AWS resources that are checked by controls that apply to the standard. This is primarily for controls that have a *change triggered* schedule type. However, some controls with a *periodic* schedule type also require resource recording. If resource recording isn't enabled or configured correctly, Security Hub CSPM might not be able to evaluate the appropriate resources, and generate accurate findings for controls that apply to the standard.

For information about how Security Hub CSPM uses resource recording in AWS Config, see Enabling and configuring AWS Config for Security Hub CSPM. For information about configuring resource recording in AWS Config, see Working with the configuration recorder in the *AWS Config Developer Guide*.

The following table specifies the types of resources to record for controls that apply to the NIST SP 800-53 Revision 5 standard in Security Hub CSPM.

AWS service	Resource types
Amazon API Gateway	`AWS::ApiGateway::Stage` , `AWS::ApiGatewayV2::Stage`
AWS AppSync	`AWS::AppSync::GraphQLApi`

AWS service	Resource types
AWS Backup	`AWS::Backup::RecoveryPoint`
AWS Certificate Manager (ACM)	`AWS::ACM::Certificate`
AWS CloudFormation	`AWS::CloudFormation::Stack`
Amazon CloudFront	`AWS::CloudFront::Distribution`
Amazon CloudWatch	`AWS::CloudWatch::Alarm`
AWS CodeBuild	`AWS::CodeBuild::Project`
AWS Database Migration Service (AWS DMS)	`AWS::DMS::Endpoint` , `AWS::DMS::ReplicationInstance` , `AWS::DMS::ReplicationTask`
Amazon DynamoDB	`AWS::DynamoDB::Table`
Amazon Elastic Compute Cloud (Amazon EC2)	`AWS::EC2::ClientVpnEndpoint` , `AWS::EC2::EIP` , `AWS::EC2::Instance` , `AWS::EC2::LaunchTemplate` , `AWS::EC2::NetworkAcl` , `AWS::EC2::NetworkInterface` , `AWS::EC2::SecurityGroup` , `AWS::EC2::Subnet` , `AWS::EC2::TransitGateway` , `AWS::EC2::VPNConnection` , `AWS::EC2::Volume`
Amazon EC2 Auto Scaling	`AWS::AutoScaling::AutoScalingGroup` , `AWS::AutoScaling::LaunchConfiguration`
Amazon Elastic Container Registry (Amazon ECR)	`AWS::ECR::Repository`

AWS service	Resource types
Amazon Elastic Container Service (Amazon ECS)	`AWS::ECS::Cluster` , `AWS::ECS::Service` , `AWS::ECS::TaskDefinition`
Amazon Elastic File System (Amazon EFS)	`AWS::EFS::AccessPoint`
Amazon Elastic Kubernetes Service (Amazon EKS)	`AWS::EKS::Cluster`
AWS Elastic Beanstalk	`AWS::ElasticBeanstalk::Environment`
Elastic Load Balancing	`AWS::ElasticLoadBalancing::LoadBalancer` , `AWS::ElasticLoadBalancingV2::Listener` , `AWS::ElasticLoadBalancingV2::LoadBalancer`
Amazon ElasticSearch	`AWS::Elasticsearch::Domain`
Amazon EMR	`AWS::EMR::SecurityConfiguration`
Amazon EventBridge	`AWS::Events::Endpoint` , `AWS::Events::EventBus`
AWS Glue	`AWS::Glue::Job`
AWS Identity and Access Management (IAM)	`AWS::IAM::Group` , `AWS::IAM::Policy` , `AWS::IAM::Role` , `AWS::IAM::User`
AWS Key Management Service (AWS KMS)	`AWS::KMS::Alias` , `AWS::KMS::Key`
Amazon Kinesis	`AWS::Kinesis::Stream`
AWS Lambda	`AWS::Lambda::Function`
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	`AWS::MSK::Cluster`

AWS service	Resource types
Amazon MQ	AWS::AmazonMQ::Broker
AWS Network Firewall	AWS::NetworkFirewall::Firewall , AWS::NetworkFirewall::FirewallPolicy , AWS::NetworkFirewall::RuleGroup
Amazon OpenSearch Service	AWS::OpenSearch::Domain
Amazon Relational Database Service (Amazon RDS)	AWS::RDS::DBCluster , AWS::RDS::DBClusterSnapshot , AWS::RDS::DBInstance , AWS::RDS::DBSnapshot , AWS::RDS::EventSubscription
Amazon Redshift	AWS::Redshift::Cluster , AWS::Redshift::ClusterSubnetGroup
Amazon Route 53	AWS::Route53::HostedZone
Amazon Simple Storage Service (Amazon S3)	AWS::S3::AccessPoint , AWS::S3::AccountPublicAccessBlock , AWS::S3::Bucket
AWS Service Catalog	AWS::ServiceCatalog::Portfolio
Amazon Simple Notification Service (Amazon SNS)	AWS::SNS::Topic
Amazon Simple Queue Service (Amazon SQS)	AWS::SQS::Queue
Amazon EC2 Systems Manager (SSM)	AWS::SSM::AssociationCompliance , AWS::SSM::ManagedInstanceInventory , AWS::SSM::PatchCompliance
Amazon SageMaker AI	AWS::SageMaker::NotebookInstance

AWS service	Resource types
AWS Secrets Manager	`AWS::SecretsManager::Secret`
AWS Transfer Family	`AWS::Transfer::Connector`
AWS WAF	`AWS::WAF::Rule` , `AWS::WAF::RuleGroup` , `AWS::WAF::WebACL` , `AWS::WAFRegional::Rule` , `AWS::WAFRegional::RuleGroup` , `AWS::WAFRegional::WebACL` , `AWS::WAFv2::RuleGroup` , `AWS::WAFv2::WebACL`

Determining which controls apply to the standard

The following list specifies the controls that support NIST SP 800-53 Revision 5 requirements and apply to the NIST SP 800-53 Revision 5 standard in AWS Security Hub CSPM. For details about specific requirements that a control supports, choose the control. Then refer to the **Related requirements** field in the details for the control. This field specifies each NIST requirement that the control supports. If the field doesn't specify a particular NIST requirement, the control doesn't support the requirement.

- [Account.1] Security contact information should be provided for an AWS account
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [APIGateway.1] API Gateway REST and WebSocket API execution logging should be enabled
- [APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication
- [APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled
- [APIGateway.4] API Gateway should be associated with a WAF Web ACL
- [APIGateway.5] API Gateway REST API cache data should be encrypted at rest
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys

- [AutoScaling.1] Auto Scaling groups associated with a load balancer should use ELB health checks
- [AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones
- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [Autoscaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses
- [AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones
- [AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events
- [CloudTrail.2] CloudTrail should have encryption at-rest enabled
- [CloudTrail.4] CloudTrail log file validation should be enabled
- [CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs
- [CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys
- [CloudWatch.15] CloudWatch alarms should have specified actions configured
- [CloudWatch.16] CloudWatch log groups should be retained for a specified time period

- [CloudWatch.17] CloudWatch alarm actions should be activated
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [CodeBuild.3] CodeBuild S3 logs should be encrypted
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [Config.1] AWS Config should be enabled and use the service-linked role for resource recording
- [DataFirehose.1] Firehose delivery streams should be encrypted at rest
- [DMS.1] Database Migration Service replication instances should not be public
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DynamoDB.1] DynamoDB tables should automatically scale capacity with demand
- [DynamoDB.2] DynamoDB tables should have point-in-time recovery enabled
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [DynamoDB.6] DynamoDB tables should have deletion protection enabled
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.1] Amazon EBS snapshots should not be publicly restorable
- [EC2.2] VPC default security groups should not allow inbound or outbound traffic

- [EC2.3] Attached Amazon EBS volumes should be encrypted at-rest
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.6] VPC flow logging should be enabled in all VPCs
- [EC2.7] EBS default encryption should be enabled
- [EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.9] Amazon EC2 instances should not have a public IPv4 address
- [EC2.10] Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service
- [EC2.12] Unused Amazon EC2 EIPs should be removed
- [EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22
- [EC2.15] Amazon EC2 subnets should not automatically assign public IP addresses
- [EC2.16] Unused Network Access Control Lists should be removed
- [EC2.17] Amazon EC2 instances should not use multiple ENIs
- [EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports
- [EC2.19] Security groups should not allow unrestricted access to ports with high risk
- [EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up
- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.55] VPCs should be configured with an interface endpoint for ECR API
- [EC2.56] VPCs should be configured with an interface endpoint for Docker Registry
- [EC2.57] VPCs should be configured with an interface endpoint for Systems Manager
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [ECR.1] ECR private repositories should have image scanning configured

- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys
- [ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions
- [ECS.2] ECS services should not have public IP addresses assigned to them automatically
- [ECS.3] ECS task definitions should not share the host's process namespace
- [ECS.4] ECS containers should run as non-privileged
- [ECS.5] ECS containers should be limited to read-only access to root filesystems
- [ECS.8] Secrets should not be passed as container environment variables
- [ECS.9] ECS task definitions should have a logging configuration
- [ECS.10] ECS Fargate services should run on the latest Fargate platform version
- [ECS.12] ECS clusters should use Container Insights
- [ECS.17] ECS task definitions should not use host network mode
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch
- [EKS.1] EKS cluster endpoints should not be publicly accessible
- [EKS.2] EKS clusters should run on a supported Kubernetes version
- [EKS.3] EKS clusters should use encrypted Kubernetes secrets
- [EKS.8] EKS clusters should have audit logging enabled
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled

- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination
- [ELB.4] Application Load Balancer should be configured to drop invalid http headers
- [ELB.5] Application and Classic Load Balancers logging should be enabled
- [ELB.6] Application, Gateway, and Network Load Balancers should have deletion protection enabled
- [ELB.7] Classic Load Balancers should have connection draining enabled
- [ELB.8] Classic Load Balancers with SSL listeners should use a predefined security policy that has strong AWS Configuration
- [ELB.9] Classic Load Balancers should have cross-zone load balancing enabled
- [ELB.10] Classic Load Balancer should span multiple Availability Zones
- [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [EMR.2] Amazon EMR block public access setting should be enabled
- [EMR.3] Amazon EMR security configurations should be encrypted at rest
- [EMR.4] Amazon EMR security configurations should be encrypted in transit
- [ES.1] Elasticsearch domains should have encryption at-rest enabled

- [ES.2] Elasticsearch domains should not be publicly accessible
- [ES.3] Elasticsearch domains should encrypt data sent between nodes
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [ES.5] Elasticsearch domains should have audit logging enabled
- [ES.6] Elasticsearch domains should have at least three data nodes
- [ES.7] Elasticsearch domains should be configured with at least three dedicated master nodes
- [ES.8] Connections to Elasticsearch domains should be encrypted using the latest TLS security policy
- [EventBridge.3] EventBridge custom event buses should have a resource-based policy attached
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.1] GuardDuty should be enabled
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.9] MFA should be enabled for the root user
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys

- [KMS.3] AWS KMS keys should not be deleted unintentionally
- [KMS.4] AWS KMS key rotation should be enabled
- [Lambda.1] Lambda function policies should prohibit public access
- [Lambda.2] Lambda functions should use supported runtimes
- [Lambda.3] Lambda functions should be in a VPC
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones
- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets

- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [RDS.1] RDS snapshot should be private
- [RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration
- [RDS.3] RDS DB instances should have encryption at-rest enabled
- [RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest
- [RDS.5] RDS DB instances should be configured with multiple Availability Zones
- [RDS.6] Enhanced monitoring should be configured for RDS DB instances
- [RDS.7] RDS clusters should have deletion protection enabled
- [RDS.8] RDS DB instances should have deletion protection enabled
- [RDS.9] RDS DB instances should publish logs to CloudWatch Logs
- [RDS.10] IAM authentication should be configured for RDS instances
- [RDS.11] RDS instances should have automatic backups enabled
- [RDS.12] IAM authentication should be configured for RDS clusters

- [RDS.13] RDS automatic minor version upgrades should be enabled
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.16] Aurora DB clusters should be configured to copy tags to DB snapshots
- [RDS.17] RDS DB instances should be configured to copy tags to snapshots
- [RDS.19] Existing RDS event notification subscriptions should be configured for critical cluster events
- [RDS.20] Existing RDS event notification subscriptions should be configured for critical database instance events
- [RDS.21] An RDS event notifications subscription should be configured for critical database parameter group events
- [RDS.22] An RDS event notifications subscription should be configured for critical database security group events
- [RDS.23] RDS instances should not use a database engine default port
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.27] RDS DB clusters should be encrypted at rest
- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.40] RDS for SQL Server DB instances should publish logs to CloudWatch Logs
- [RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.1] Amazon Redshift clusters should prohibit public access
- [Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Redshift.4] Amazon Redshift clusters should have audit logging enabled
- [Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled
- [Redshift.7] Redshift clusters should use enhanced VPC routing
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name

- [Redshift.10] Redshift clusters should be encrypted at rest
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.1] S3 general purpose buckets should have block public access settings enabled
- [S3.2] S3 general purpose buckets should block public read access
- [S3.3] S3 general purpose buckets should block public write access
- [S3.5] S3 general purpose buckets should require requests to use SSL
- [S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts
- [S3.7] S3 general purpose buckets should use cross-Region replication
- [S3.8] S3 general purpose buckets should block public access
- [S3.9] S3 general purpose buckets should have server access logging enabled
- [S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.12] ACLs should not be used to manage user access to S3 general purpose buckets
- [S3.13] S3 general purpose buckets should have Lifecycle configurations
- [S3.14] S3 general purpose buckets should have versioning enabled
- [S3.15] S3 general purpose buckets should have Object Lock enabled
- [S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys
- [S3.19] S3 access points should have block public access settings enabled
- [S3.20] S3 general purpose buckets should have MFA delete enabled
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1
- [SecretsManager.1] Secrets Manager secrets should have automatic rotation enabled
- [SecretsManager.2] Secrets Manager secrets configured with automatic rotation should rotate successfully

- [SecretsManager.3] Remove unused Secrets Manager secrets
- [SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SNS.1] SNS topics should be encrypted at-rest using AWS KMS
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SSM.1] Amazon EC2 instances should be managed by AWS Systems Manager
- [SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.4] SSM documents should not be public
- [Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection
- [Transfer.3] Transfer Family connectors should have logging enabled
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.2] AWS WAF Classic Regional rules should have at least one condition
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WAF.11] AWS WAF web ACL logging should be enabled
- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled

NIST SP 800-171 Revision 2 in Security Hub CSPM

NIST Special Publication 800-171 Revision 2 (NIST SP 800-171 Rev. 2) is a cybersecurity and compliance framework developed by the National Institute of Standards and Technology (NIST), an agency that's part of the U.S. Department of Commerce. This compliance framework provides recommended security requirements for protecting the confidentiality of Controlled Unclassified Information in systems and organizations that aren't part of the U.S. federal government.

Controlled Unclassified Information, also referred to as *CUI*, is sensitive information that doesn't meet government criteria for classification but must be protected. It's information that is considered sensitive and is created or possessed by the U.S. federal government or other entities on behalf of the U.S. federal government.

NIST SP 800-171 Rev. 2 provides recommended security requirements for protecting the confidentiality of CUI when:

- The information resides in non-federal systems and organizations,
- The non-federal organization is not collecting or maintaining information on behalf of a federal agency or using or operating a system on behalf of an agency, and
- There are no specific safeguarding requirements for protecting the confidentiality of CUI prescribed by the authorizing law, regulation, or government-wide policy for the CUI category listed in the CUI Registry.

The requirements apply to all components of non-federal systems and organizations that process, store, or transmit CUI, or provide security protection for the components. For more information, see NIST SP 800-171 Rev. 2 in the *NIST Computer Security Resource Center*.

AWS Security Hub CSPM provides security controls that support a subset of NIST SP 800-171 Revision 2 requirements. The controls perform automated security checks for certain AWS services and resources. To enable and manage these controls, you can enable the NIST SP 800-171 Revision 2 framework as a standard in Security Hub CSPM. Note that the controls don't support NIST SP 800-171 Revision 2 requirements that require manual checks.

Topics

Configuring resource recording for controls that apply to the standard

To optimize coverage and the accuracy of findings, it's important to enable and configure resource recording in AWS Config before you enable the NIST SP 800-171 Revision 2 standard in AWS Security Hub CSPM. When you configure resource recording, also be sure to enable it for all the types of AWS resources that are checked by controls that apply to the standard. Otherwise, Security Hub CSPM might not be able to evaluate the appropriate resources, and generate accurate findings for controls that apply to the standard.

For information about how Security Hub CSPM uses resource recording in AWS Config, see Enabling and configuring AWS Config for Security Hub CSPM. For information about configuring resource recording in AWS Config, see Working with the configuration recorder in the *AWS Config Developer Guide*.

The following table specifies the types of resources to record for controls that apply to the NIST SP 800-171 Revision 2 standard in Security Hub CSPM.

AWS service	Resource types
AWS Certificate Manager (ACM)	`AWS::ACM::Certificate`
Amazon API Gateway	`AWS::ApiGateway::Stage`
Amazon CloudFront	`AWS::CloudFront::Distribution`
Amazon CloudWatch	`AWS::CloudWatch::Alarm`
Amazon Elastic Compute Cloud (Amazon EC2)	`AWS::EC2::ClientVpnEndpoint` , `AWS::EC2::NetworkAcl` , `AWS::EC2::SecurityGroup` , `AWS::EC2::VPC` , `AWS::EC2::VPNConnection`
Elastic Load Balancing	`AWS::ElasticLoadBalancing::LoadBalancer`
AWS Identity and Access Management (IAM)	`AWS::IAM::Policy` , `AWS::IAM::User`
AWS Key Management Service (AWS KMS)	`AWS::KMS::Alias` , `AWS::KMS::Key`
AWS Network Firewall	`AWS::NetworkFirewall::FirewallPolicy` , `AWS::NetworkFirewall::RuleGroup`
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::Bucket`
Amazon Simple Notification Service (Amazon SNS)	`AWS::SNS::Topic`
AWS Systems Manager (SSM)	`AWS::SSM::PatchCompliance`

AWS service	Resource types
AWS WAF	`AWS::WAFv2::RuleGroup`

Determining which controls apply to the standard

The following list specifies the controls that support NIST SP 800-171 Revision 2 requirements and apply to the NIST SP 800-171 Revision 2 standard in AWS Security Hub CSPM. For details about specific requirements that a control supports, choose the control. Then refer to the **Related requirements** field in the details for the control. This field specifies each NIST requirement that the control supports. If the field doesn't specify a particular NIST requirement, the control doesn't support the requirement.

- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudTrail.2] CloudTrail should have encryption at-rest enabled
- [CloudTrail.3] At least one CloudTrail trail should be enabled
- [CloudTrail.4] CloudTrail log file validation should be enabled
- [CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user
- [CloudWatch.2] Ensure a log metric filter and alarm exist for unauthorized API calls
- [CloudWatch.4] Ensure a log metric filter and alarm exist for IAM policy changes
- [CloudWatch.5] Ensure a log metric filter and alarm exist for CloudTrail configuration changes
- [CloudWatch.6] Ensure a log metric filter and alarm exist for AWS Management Console authentication failures
- [CloudWatch.7] Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys
- [CloudWatch.8] Ensure a log metric filter and alarm exist for S3 bucket policy changes
- [CloudWatch.9] Ensure a log metric filter and alarm exist for AWS Config configuration changes
- [CloudWatch.10] Ensure a log metric filter and alarm exist for security group changes

- [CloudWatch.11] Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)
- [CloudWatch.12] Ensure a log metric filter and alarm exist for changes to network gateways
- [CloudWatch.13] Ensure a log metric filter and alarm exist for route table changes
- [CloudWatch.14] Ensure a log metric filter and alarm exist for VPC changes
- [CloudWatch.15] CloudWatch alarms should have specified actions configured
- [EC2.6] VPC flow logging should be enabled in all VPCs
- [EC2.10] Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service
- [EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22
- [EC2.16] Unused Network Access Control Lists should be removed
- [EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports
- [EC2.19] Security groups should not allow unrestricted access to ports with high risk
- [EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up
- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination
- [ELB.8] Classic Load Balancers with SSL listeners should use a predefined security policy that has strong AWS Configuration
- [GuardDuty.1] GuardDuty should be enabled
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.10] Password policies for IAM users should have strong configurations
- [IAM.11] Ensure IAM password policy requires at least one uppercase letter
- [IAM.12] Ensure IAM password policy requires at least one lowercase letter
- [IAM.13] Ensure IAM password policy requires at least one symbol
- [IAM.14] Ensure IAM password policy requires at least one number

- [IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
- [IAM.16] Ensure IAM password policy prevents password reuse
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [S3.5] S3 general purpose buckets should require requests to use SSL
- [S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts
- [S3.9] S3 general purpose buckets should have server access logging enabled
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.14] S3 general purpose buckets should have versioning enabled
- [S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys
- [SNS.1] SNS topics should be encrypted at-rest using AWS KMS
- [SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation
- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled

PCI DSS in Security Hub CSPM

The Payment Card Industry Data Security Standard (PCI DSS) is a third-party compliance framework that provides a set of rules and guidelines for safely handling credit and debit card information. The PCI Security Standards Council (SSC) creates and updates this framework.

AWS Security Hub CSPM provides a PCI DSS standard that can help you stay compliant with this third-party framework. You can use this standard to discover security vulnerabilities in AWS resources that handle cardholder data. We recommend enabling this standard in AWS accounts

that have resources that store, process, or transmit cardholder data or sensitive authentication data. Assessments by the PCI SSC validated this standard.

Security Hub CSPM offers support for both PCI DSS v3.2.1 and PCI DSS v4.0.1. We recommend using v4.0.1 to stay current with security best practices. You can have both versions of the standard enabled at the same time. For information about enabling standards, see Enabling a security standard. If you currently use v3.2.1 but want to use only v4.0.1, enable the newer version before disabling the older version. This prevents gaps in your security checks. If you use the Security Hub CSPM integration with AWS Organizations and want to batch enable v4.0.1 in multiple accounts, we recommend using central configuration to do so.

The following sections specify which controls apply to PCI DSS v3.2.1 and PCI DSS v4.0.1.

Controls that apply to PCI DSS v3.2.1

The following list specifies which Security Hub CSPM controls apply to PCI DSS v3.2.1. To review the details of a control, choose the control.

[AutoScaling.1] Auto Scaling groups associated with a load balancer should use ELB health checks

[CloudTrail.2] CloudTrail should have encryption at-rest enabled

[CloudTrail.3] At least one CloudTrail trail should be enabled

[CloudTrail.4] CloudTrail log file validation should be enabled

[CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs

[CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user

[CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials

[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials

[Config.1] AWS Config should be enabled and use the service-linked role for resource recording

[DMS.1] Database Migration Service replication instances should not be public

[EC2.1] Amazon EBS snapshots should not be publicly restorable

[EC2.2] VPC default security groups should not allow inbound or outbound traffic

[EC2.6] VPC flow logging should be enabled in all VPCs

[EC2.12] Unused Amazon EC2 EIPs should be removed

[EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22

[ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS

[ES.1] Elasticsearch domains should have encryption at-rest enabled

[ES.2] Elasticsearch domains should not be publicly accessible

[GuardDuty.1] GuardDuty should be enabled

[IAM.1] IAM policies should not allow full "*" administrative privileges

[IAM.2] IAM users should not have IAM policies attached

[IAM.4] IAM root user access key should not exist

[IAM.6] Hardware MFA should be enabled for the root user

[IAM.8] Unused IAM user credentials should be removed

[IAM.9] MFA should be enabled for the root user

[IAM.10] Password policies for IAM users should have strong configurations

[IAM.19] MFA should be enabled for all IAM users

[KMS.4] AWS KMS key rotation should be enabled

[Lambda.1] Lambda function policies should prohibit public access

[Lambda.3] Lambda functions should be in a VPC

[Opensearch.1] OpenSearch domains should have encryption at rest enabled

[Opensearch.2] OpenSearch domains should not be publicly accessible

[RDS.1] RDS snapshot should be private

[RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration

[Redshift.1] Amazon Redshift clusters should prohibit public access

[S3.1] S3 general purpose buckets should have block public access settings enabled

[S3.2] S3 general purpose buckets should block public read access

[S3.3] S3 general purpose buckets should block public write access

[S3.5] S3 general purpose buckets should require requests to use SSL

[S3.7] S3 general purpose buckets should use cross-Region replication

[SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access

[SSM.1] Amazon EC2 instances should be managed by AWS Systems Manager

[SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation

[SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT

Controls that apply to PCI DSS v4.0.1

The following list specifies which Security Hub CSPM controls apply to PCI DSS v4.0.1. To review the details of a control, choose the control.

[ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period

[ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits

[APIGateway.9] Access logging should be configured for API Gateway V2 Stages

[AppSync.2] AWS AppSync should have field-level logging enabled

[AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)

[Autoscaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses

[CloudFront.1] CloudFront distributions should have a default root object configured

[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins

[CloudFront.12] CloudFront distributions should not point to non-existent S3 origins

[CloudFront.3] CloudFront distributions should require encryption in transit

[CloudFront.5] CloudFront distributions should have logging enabled

[CloudFront.6] CloudFront distributions should have WAF enabled

[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins

[CloudTrail.2] CloudTrail should have encryption at-rest enabled

[CloudTrail.3] At least one CloudTrail trail should be enabled

[CloudTrail.4] CloudTrail log file validation should be enabled

[CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible

[CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket

[CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials

[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials

[CodeBuild.3] CodeBuild S3 logs should be encrypted

[DMS.1] Database Migration Service replication instances should not be public

[DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled

[DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled

[DMS.12] DMS endpoints for Redis OSS should have TLS enabled

[DMS.6] DMS replication instances should have automatic minor version upgrade enabled

[DMS.7] DMS replication tasks for the target database should have logging enabled

[DMS.8] DMS replication tasks for the source database should have logging enabled

[DMS.9] DMS endpoints should use SSL

[DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period

[DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public

[DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs

[DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit

[EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22

[EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389

[EC2.15] Amazon EC2 subnets should not automatically assign public IP addresses

[EC2.16] Unused Network Access Control Lists should be removed

[EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)

[EC2.171] EC2 VPN connections should have logging enabled

[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389

[EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces

[EC2.51] EC2 Client VPN endpoints should have client connection logging enabled

[EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports

[EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports

[EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)

[ECR.1] ECR private repositories should have image scanning configured

[ECS.10] ECS Fargate services should run on the latest Fargate platform version

[ECS.16] ECS task sets should not automatically assign public IP addresses

[ECS.2] ECS services should not have public IP addresses assigned to them automatically

[ECS.8] Secrets should not be passed as container environment variables

[EFS.4] EFS access points should enforce a user identity

[EKS.1] EKS cluster endpoints should not be publicly accessible

[EKS.2] EKS clusters should run on a supported Kubernetes version

[EKS.3] EKS clusters should use encrypted Kubernetes secrets

[EKS.8] EKS clusters should have audit logging enabled

[ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled

[ElastiCache.5] ElastiCache replication groups should be encrypted in transit

[ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled

[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled

[ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch

[ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode

[ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode

[ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination

[ELB.4] Application Load Balancer should be configured to drop invalid http headers

[ELB.8] Classic Load Balancers with SSL listeners should use a predefined security policy that has strong AWS Configuration

[EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses

[EMR.2] Amazon EMR block public access setting should be enabled

[ES.2] Elasticsearch domains should not be publicly accessible

[ES.3] Elasticsearch domains should encrypt data sent between nodes

[ES.5] Elasticsearch domains should have audit logging enabled

[ES.8] Connections to Elasticsearch domains should be encrypted using the latest TLS security policy

[EventBridge.3] EventBridge custom event buses should have a resource-based policy attached

[GuardDuty.1] GuardDuty should be enabled

[GuardDuty.10] GuardDuty S3 Protection should be enabled

[GuardDuty.6] GuardDuty Lambda Protection should be enabled

[GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled

[GuardDuty.9] GuardDuty RDS Protection should be enabled

[IAM.3] IAM users' access keys should be rotated every 90 days or less

[IAM.5] MFA should be enabled for all IAM users that have a console password

[IAM.6] Hardware MFA should be enabled for the root user

[IAM.7] Password policies for IAM users should have strong configurations

[IAM.8] Unused IAM user credentials should be removed

[IAM.9] MFA should be enabled for the root user

[IAM.11] Ensure IAM password policy requires at least one uppercase letter

[IAM.12] Ensure IAM password policy requires at least one lowercase letter

[IAM.14] Ensure IAM password policy requires at least one number

[IAM.16] Ensure IAM password policy prevents password reuse

[IAM.17] Ensure IAM password policy expires passwords within 90 days or less

[IAM.18] Ensure a support role has been created to manage incidents with AWS Support

[IAM.19] MFA should be enabled for all IAM users

[Inspector.1] Amazon Inspector EC2 scanning should be enabled

[Inspector.2] Amazon Inspector ECR scanning should be enabled

[Inspector.3] Amazon Inspector Lambda code scanning should be enabled

[Inspector.4] Amazon Inspector Lambda standard scanning should be enabled

[KMS.4] AWS KMS key rotation should be enabled

[Lambda.1] Lambda function policies should prohibit public access

[Lambda.2] Lambda functions should use supported runtimes

[MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch

[MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled

[MSK.1] MSK clusters should be encrypted in transit among broker nodes

[MSK.3] MSK Connect connectors should be encrypted in transit

[Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs

[Neptune.3] Neptune DB cluster snapshots should not be public

[Opensearch.10] OpenSearch domains should have the latest software update installed

[Opensearch.5] OpenSearch domains should have audit logging enabled

[RDS.13] RDS automatic minor version upgrades should be enabled

[RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration

[RDS.20] Existing RDS event notification subscriptions should be configured for critical database instance events

[RDS.21] An RDS event notifications subscription should be configured for critical database parameter group events

[RDS.22] An RDS event notifications subscription should be configured for critical database security group events

[RDS.24] RDS Database clusters should use a custom administrator username

[RDS.25] RDS database instances should use a custom administrator username

[RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs

[RDS.35] RDS DB clusters should have automatic minor version upgrade enabled

[RDS.36] RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs

[RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs

[RDS.9] RDS DB instances should publish logs to CloudWatch Logs

[Redshift.1] Amazon Redshift clusters should prohibit public access

[Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins

[Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit

[Redshift.4] Amazon Redshift clusters should have audit logging enabled

[Route53.2] Route 53 public hosted zones should log DNS queries

[S3.1] S3 general purpose buckets should have block public access settings enabled

[S3.15] S3 general purpose buckets should have Object Lock enabled

[S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys

[S3.19] S3 access points should have block public access settings enabled

[S3.22] S3 general purpose buckets should log object-level write events

[S3.23] S3 general purpose buckets should log object-level read events

[S3.24] S3 Multi-Region Access Points should have block public access settings enabled

[S3.5] S3 general purpose buckets should require requests to use SSL

[S3.8] S3 general purpose buckets should block public access

[S3.9] S3 general purpose buckets should have server access logging enabled

[SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access

[SecretsManager.1] Secrets Manager secrets should have automatic rotation enabled

[SecretsManager.2] Secrets Manager secrets configured with automatic rotation should rotate successfully

[SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days

[SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation

[SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT

[StepFunctions.1] Step Functions state machines should have logging turned on

[Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection

[WAF.1] AWS WAF Classic Global Web ACL logging should be enabled

[WAF.11] AWS WAF web ACL logging should be enabled

Service-managed standards in Security Hub CSPM

A service-managed standard is a security standard that another AWS service manages but that you can view in Security Hub CSPM. For example, Service-Managed Standard: AWS Control Tower is a

service-managed standard that AWS Control Tower manages. A service-managed standard differs from a security standard that AWS Security Hub CSPM manages in the following ways:

- **Standard creation and deletion** – You create and delete a service-managed standard with the managing service's console or API, or with the AWS CLI. Until you create the standard in the managing service in one of those ways, the standard doesn't appear in the Security Hub CSPM console and isn't accessible by the Security Hub CSPM API or AWS CLI.
- **No automatic enablement of controls** – When you create a service-managed standard, Security Hub CSPM and the managing service don't automatically enable the controls that apply to the standard. In addition, when Security Hub CSPM releases new controls for the standard, they're not automatically enabled. This is a departure from standards that Security Hub CSPM manages. For more information about the usual way of configuring controls in Security Hub CSPM, see Understanding security controls in Security Hub CSPM.
- **Enabling and disabling controls** – We recommend enabling and disabling controls in the managing service to avoid drift.
- **Availability of controls** – The managing service chooses which controls are available as part of the service-managed standard. Available controls may include all, or a subset of, the existing Security Hub CSPM controls.

After the managing service creates the service-managed standard and makes controls available for it, you can access your control findings, control statuses, and standard security score in the Security Hub CSPM console, Security Hub CSPM API, or AWS CLI. Some or all of this information may also be available in the managing service.

Select a service-managed standard from the following list to view more details about it.

Service-managed standards

- Service-Managed Standard: AWS Control Tower

Service-Managed Standard: AWS Control Tower

This section provides information about Service-Managed Standard: AWS Control Tower.

What is Service-Managed Standard: AWS Control Tower?

This standard is designed for users of AWS Security Hub CSPM and AWS Control Tower. It lets you configure the proactive controls of AWS Control Tower alongside the detective controls of Security Hub CSPM in the AWS Control Tower service.

Proactive controls help ensure that your AWS accounts maintain compliance because they flag actions that may lead to policy violations or misconfigurations. Detective controls detect noncompliance of resources (for example, misconfigurations) within your AWS accounts. By enabling proactive and detective controls for your AWS environment, you can enhance your security posture at different stages of development.

> **Tip**
>
> Service-managed standards differ from standards that AWS Security Hub CSPM manages. For example, you must create and delete a service-managed standard in the managing service. For more information, see Service-managed standards in Security Hub CSPM.

In the Security Hub CSPM console and API, you can view Service-Managed Standard: AWS Control Tower alongside other Security Hub CSPM standards.

Creating the standard

This standard is available only if you create the standard in AWS Control Tower. AWS Control Tower creates the standard when you first enable an applicable control by using one of the following methods:

- AWS Control Tower console
- AWS Control Tower API (call the `EnableControl` API)
- AWS CLI (run the `enable-control` command)

Security Hub CSPM controls are identified in the AWS Control Tower console as **SH.*`ControlID`*** (for example, **SH.CodeBuild.1**).

When you create the standard, if you haven't already enabled Security Hub CSPM, AWS Control Tower also enables Security Hub CSPM for you.

If you haven't set up AWS Control Tower, you can't view or access this standard in the Security Hub CSPM console, Security Hub CSPM API, or AWS CLI. Even if you have set up AWS Control Tower, you can't view or access this standard in Security Hub CSPM without first creating the standard in AWS Control Tower using one of the preceding methods.

This standard is only available in the AWS Regions where AWS Control Tower is available, including AWS GovCloud (US).

Enabling and disabling controls in the standard

After you've created the standard in the AWS Control Tower console, you can view the standard and its available controls in both services.

After you first create the standard, it doesn't have any controls that are automatically enabled. In addition, when Security Hub CSPM adds new controls, they aren't automatically enabled for Service-Managed Standard: AWS Control Tower. You should enable and disable controls for the standard in AWS Control Tower by using one of the following methods:

- AWS Control Tower console
- AWS Control Tower API (call the `EnableControl` and `DisableControl` APIs)
- AWS CLI (run the `enable-control` and `disable-control` commands)

When you change the enablement status of a control in AWS Control Tower, the change is also reflected in Security Hub CSPM.

However, disabling a control in Security Hub CSPM that's enabled in AWS Control Tower results in control drift. The control status in AWS Control Tower shows as `Drifted`. You can resolve this drift by selecting Re-register OU in the AWS Control Tower console, or by disabling and re-enabling the control in AWS Control Tower using one of the preceding methods.

Completing enablement and disablement actions in AWS Control Tower helps you avoid control drift.

When you enable or disable controls in AWS Control Tower, the action applies across accounts and Regions. If you enable and disable controls in Security Hub CSPM (not recommended for this standard), the action applies only to the current account and Region.

> **Note**
>
> Central configuration can't be used to manage Service-Managed Standard: AWS Control Tower. If you use central configuration, you can use *only* the AWS Control Tower service to enable and disable controls in this standard for a centrally managed account.

Viewing enablement status and control status

You can view the enablement status of a control by using one of the following methods:

- Security Hub CSPM console, Security Hub CSPM API, or AWS CLI
- AWS Control Tower console
- AWS Control Tower API to see a list of enabled controls (call the `ListEnabledControls` API)
- AWS CLI to see a list of enabled controls (run the `list-enabled-controls` command)

A control that you disable in AWS Control Tower has an enablement status of `Disabled` in Security Hub CSPM unless you explicitly enable that control in Security Hub CSPM.

Security Hub CSPM calculates control status based on the workflow status and compliance status of the control findings. For more information about enablement status and control status, see Reviewing the details of controls in Security Hub CSPM.

Based on control statuses, Security Hub CSPM calculates a security score for Service-Managed Standard: AWS Control Tower. This score is only available in Security Hub CSPM. In addition, you can only view control findings in Security Hub CSPM. The standard security score and control findings aren't available in AWS Control Tower.

> **Note**
>
> When you enable controls for Service-Managed Standard: AWS Control Tower, Security Hub CSPM may take up to 18 hours to generate findings for controls that use an existing AWS Config service-linked rule. You may have existing service-linked rules if you've enabled other standards and controls in Security Hub CSPM. For more information, see Schedule for running security checks.

Deleting the standard

You can delete this standard in AWS Control Tower by disabling all applicable controls using one of the following methods:

- AWS Control Tower console
- AWS Control Tower API (call the `DisableControl` API)
- AWS CLI (run the `disable-control` command)

Disabling all controls deletes the standard in all managed accounts and governed Regions in AWS Control Tower. Deleting the standard in AWS Control Tower removes it from the **Standards** page

of the Security Hub CSPM console, and you can no longer access it by using the Security Hub CSPM API or AWS CLI.

> **Note**
>
> Disabling all controls from the standard in Security Hub CSPM doesn't disable or delete the standard.

Disabling the Security Hub CSPM service removes Service-Managed Standard: AWS Control Tower and any other standards that you've enabled.

Finding field format for Service-Managed Standard: AWS Control Tower

When you create Service-Managed Standard: AWS Control Tower and enable controls for it, you'll start to receive control findings in Security Hub CSPM. Security Hub CSPM reports control findings in the AWS Security Finding Format (ASFF). These are the ASFF values for this standard's Amazon Resource Name (ARN) and `GeneratorId`:

- **Standard ARN** – *`arn:aws:us-east-1`*`:securityhub:::standards/service-managed-aws-control-tower/v/1.0.0`
- **GeneratorId** – `service-managed-aws-control-tower/v/1.0.0/`*`CodeBuild.1`*

For a sample finding for Service-Managed Standard: AWS Control Tower, see Samples of control findings.

Controls that apply to Service-Managed Standard: AWS Control Tower

Service-Managed Standard: AWS Control Tower supports a subset of controls that are part of the AWS Foundational Security Best Practices (FSBP) standard. Choose a control to view information about it, including remediation steps for failed findings.

The following list shows available controls for Service-Managed Standard: AWS Control Tower. Regional limits on controls match Regional limits on the corollary controls in the FSBP standard. This list shows standard-agnostic security control IDs. In the AWS Control Tower console, control IDs are formatted as **SH.*`ControlID`*** (for example **SH.CodeBuild.1**). In Security Hub CSPM, if consolidated control findings is turned off in your account, the `ProductFields.ControlId` field uses the standard-based control ID. The standard-based control ID is formatted as **CT.*`ControlId`*** (for example, **CT.CodeBuild.1**).

- [Account.1] Security contact information should be provided for an AWS account
- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [APIGateway.1] API Gateway REST and WebSocket API execution logging should be enabled
- [APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication
- [APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled
- [APIGateway.4] API Gateway should be associated with a WAF Web ACL
- [APIGateway.5] API Gateway REST API cache data should be encrypted at rest
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [AutoScaling.1] Auto Scaling groups associated with a load balancer should use ELB health checks
- [AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones
- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [Autoscaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses
- [AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones
- [AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates
- [CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events
- [CloudTrail.2] CloudTrail should have encryption at-rest enabled
- [CloudTrail.4] CloudTrail log file validation should be enabled
- [CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [CodeBuild.3] CodeBuild S3 logs should be encrypted

- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [DMS.1] Database Migration Service replication instances should not be public
- [DMS.9] DMS endpoints should use SSL
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DynamoDB.1] DynamoDB tables should automatically scale capacity with demand
- [DynamoDB.2] DynamoDB tables should have point-in-time recovery enabled
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [EC2.1] Amazon EBS snapshots should not be publicly restorable
- [EC2.2] VPC default security groups should not allow inbound or outbound traffic
- [EC2.3] Attached Amazon EBS volumes should be encrypted at-rest
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.6] VPC flow logging should be enabled in all VPCs
- [EC2.7] EBS default encryption should be enabled
- [EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.9] Amazon EC2 instances should not have a public IPv4 address
- [EC2.10] Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service
- [EC2.15] Amazon EC2 subnets should not automatically assign public IP addresses
- [EC2.16] Unused Network Access Control Lists should be removed
- [EC2.17] Amazon EC2 instances should not use multiple ENIs
- [EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports
- [EC2.19] Security groups should not allow unrestricted access to ports with high risk
- [EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up
- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces

- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions
- [ECS.2] ECS services should not have public IP addresses assigned to them automatically
- [ECS.3] ECS task definitions should not share the host's process namespace
- [ECS.4] ECS containers should run as non-privileged
- [ECS.5] ECS containers should be limited to read-only access to root filesystems
- [ECS.8] Secrets should not be passed as container environment variables
- [ECS.10] ECS Fargate services should run on the latest Fargate platform version
- [ECS.12] ECS clusters should use Container Insights
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EKS.1] EKS cluster endpoints should not be publicly accessible
- [EKS.2] EKS clusters should run on a supported Kubernetes version
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination
- [ELB.4] Application Load Balancer should be configured to drop invalid http headers
- [ELB.5] Application and Classic Load Balancers logging should be enabled

- [ELB.6] Application, Gateway, and Network Load Balancers should have deletion protection enabled
- [ELB.7] Classic Load Balancers should have connection draining enabled
- [ELB.8] Classic Load Balancers with SSL listeners should use a predefined security policy that has strong AWS Configuration
- [ELB.9] Classic Load Balancers should have cross-zone load balancing enabled
- [ELB.10] Classic Load Balancer should span multiple Availability Zones
- [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [ES.1] Elasticsearch domains should have encryption at-rest enabled
- [ES.2] Elasticsearch domains should not be publicly accessible
- [ES.3] Elasticsearch domains should encrypt data sent between nodes
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [ES.5] Elasticsearch domains should have audit logging enabled
- [ES.6] Elasticsearch domains should have at least three data nodes
- [ES.7] Elasticsearch domains should be configured with at least three dedicated master nodes
- [ES.8] Connections to Elasticsearch domains should be encrypted using the latest TLS security policy
- [EventBridge.3] EventBridge custom event buses should have a resource-based policy attached
- [GuardDuty.1] GuardDuty should be enabled
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user

- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [KMS.3] AWS KMS keys should not be deleted unintentionally
- [KMS.4] AWS KMS key rotation should be enabled
- [Lambda.1] Lambda function policies should prohibit public access
- [Lambda.2] Lambda functions should use supported runtimes
- [Lambda.3] Lambda functions should be in a VPC
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets
- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty

- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [RDS.1] RDS snapshot should be private
- [RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration
- [RDS.3] RDS DB instances should have encryption at-rest enabled
- [RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest
- [RDS.5] RDS DB instances should be configured with multiple Availability Zones
- [RDS.6] Enhanced monitoring should be configured for RDS DB instances
- [RDS.8] RDS DB instances should have deletion protection enabled
- [RDS.9] RDS DB instances should publish logs to CloudWatch Logs
- [RDS.10] IAM authentication should be configured for RDS instances
- [RDS.11] RDS instances should have automatic backups enabled
- [RDS.12] IAM authentication should be configured for RDS clusters
- [RDS.13] RDS automatic minor version upgrades should be enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.17] RDS DB instances should be configured to copy tags to snapshots
- [RDS.18] RDS instances should be deployed in a VPC
- [RDS.19] Existing RDS event notification subscriptions should be configured for critical cluster events
- [RDS.20] Existing RDS event notification subscriptions should be configured for critical database instance events
- [RDS.21] An RDS event notifications subscription should be configured for critical database parameter group events

- [RDS.22] An RDS event notifications subscription should be configured for critical database security group events
- [RDS.23] RDS instances should not use a database engine default port
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.27] RDS DB clusters should be encrypted at rest
- [Redshift.1] Amazon Redshift clusters should prohibit public access
- [Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit
- [Redshift.4] Amazon Redshift clusters should have audit logging enabled
- [Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled
- [Redshift.7] Redshift clusters should use enhanced VPC routing
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.10] Redshift clusters should be encrypted at rest
- [S3.1] S3 general purpose buckets should have block public access settings enabled
- [S3.2] S3 general purpose buckets should block public read access
- [S3.3] S3 general purpose buckets should block public write access
- [S3.5] S3 general purpose buckets should require requests to use SSL
- [S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts
- [S3.8] S3 general purpose buckets should block public access
- [S3.9] S3 general purpose buckets should have server access logging enabled
- [S3.12] ACLs should not be used to manage user access to S3 general purpose buckets
- [S3.13] S3 general purpose buckets should have Lifecycle configurations
- [S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SecretsManager.1] Secrets Manager secrets should have automatic rotation enabled
- [SecretsManager.2] Secrets Manager secrets configured with automatic rotation should rotate successfully
- [SecretsManager.3] Remove unused Secrets Manager secrets
- [SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days

- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SSM.1] Amazon EC2 instances should be managed by AWS Systems Manager
- [SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.4] SSM documents should not be public
- [WAF.2] AWS WAF Classic Regional rules should have at least one condition
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group

For more information about this standard, see Security Hub CSPM controls in the *AWS Control Tower User Guide*.

Enabling a security standard

When you enable a security standard in AWS Security Hub CSPM, Security Hub CSPM automatically creates and enables all the controls that apply to the standard. Security Hub CSPM also starts running security checks and generating findings for the controls.

To optimize coverage and the accuracy of findings, enable and configure resource recording in AWS Config before you enable a standard. When you configure resource recording, also be sure to enable it for all the types of resources that are checked by controls that apply to the standard. Otherwise, Security Hub CSPM might not be able to evaluate the appropriate resources, and generate accurate findings for controls that apply to the standard. For more information, see Enabling and configuring AWS Config for Security Hub CSPM.

After you enable a standard, you can disable or later re-enable individual controls that apply to the standard. If you disable a control for a standard, Security Hub CSPM stops generating findings for the control. In addition, Security Hub CSPM ignores the control when it calculates the security score for the standard. The security score is the percentage of controls that passed evaluation, relative to the total number of controls that apply to the standard, are enabled, and have evaluation data.

When you enable a standard, Security Hub CSPM generates a preliminary security score for the standard, typically within 30 minutes of your first visit to the **Summary** or **Security standards**

page on the Security Hub CSPM console. Security scores are generated only for standards that are enabled when you visit those pages on the console. In addition, resource recording must be configured in AWS Config for the scores to appear. In the China Regions and AWS GovCloud (US) Regions, it can take up to 24 hours for Security Hub CSPM to generate a preliminary security score for a standard. After Security Hub CSPM generates a preliminary score, it updates the score every 24 hours. To determine when a security score was last updated, you can refer to a timestamp that Security Hub CSPM provides for the score. For more information, see Calculating security scores.

How you enable a standard depends on whether you use central configuration to manage Security Hub CSPM for multiple accounts and AWS Regions. We recommend using central configuration if you want to enable standards in multi-account, multi-Region environments. You can use central configuration if you integrate Security Hub CSPM with AWS Organizations. If you don't use central configuration, you must enable each standard separately in each account and each Region.

Topics

Enabling a standard in multiple accounts and AWS Regions

To enable and configure a security standard across multiple accounts and AWS Regions, use central configuration. With central configuration, the delegated Security Hub CSPM administrator can create Security Hub CSPM configuration policies that enable one or more standards. The administrator can then associate a configuration policy with individual accounts, organizational units (OUs), or the root. A configuration policy affects the home Region, also referred to as an *aggregation Region*, and all linked Regions.

Configuration policies offer customization options. For example, you might choose to enable only the AWS Foundational Security Best Practices (FSBP) standard for one OU. For another OU, you might choose to enable both the FSBP standard and the Center for Internet Security (CIS) AWS Foundations Benchmark v1.4.0 standard. For information about creating a configuration policy that enables particular standards that you specify, see Creating and associating configuration policies.

If you use central configuration, Security Hub CSPM doesn't automatically enable any standards in new or existing accounts. Instead, the Security Hub CSPM administrator specifies which standards to enable in different accounts when they create Security Hub CSPM configuration policies for their

organization. Security Hub CSPM offers a recommended configuration policy in which only the FSBP standard is enabled. For more information, see Types of configuration policies.

> **Note**
>
> The Security Hub CSPM administrator can use configuration policies to enable any standard except the AWS Control Tower service-managed standard. To enable this standard, the administrator must use AWS Control Tower directly. They must also use AWS Control Tower to enable or disable individual controls in this standard for a centrally managed account.

If you want some accounts to enable and configure standards for their own accounts, the Security Hub CSPM administrator can designate those accounts as *self-managed accounts*. Self-managed accounts must enable and configure standards separately in each Region.

Enabling a standard in a single account and AWS Region

If you don't use central configuration or you have a self-managed account, you can't use configuration policies to centrally enable security standards in multiple accounts or AWS Regions. However, you can enable a standard in a single account and Region. You can do this by using the Security Hub CSPM console or the Security Hub CSPM API.

Security Hub CSPM console

Follow these steps to enable a standard in one account and Region by using the Security Hub CSPM console.

To enable a standard in one account and Region

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. By using the AWS Region selector in the upper-right corner of the page, choose the Region in which you want to enable the standard.
3. In the navigation pane, choose **Security standards**. The **Security standards** page lists all the standards that Security Hub CSPM currently supports. If you already enabled a standard, the section for the standard includes the current security score and additional details for the standard.
4. In the section for the standard that you want to enable, choose **Enable standard**.

To enable the standard in additional Regions, repeat the preceding steps in each additional Region.

Security Hub CSPM API

To enable a standard programmatically in a single account and Region, use the BatchEnableStandards operation. Or, if you're using the AWS Command Line Interface (AWS CLI), run the batch-enable-standards command.

In your request, use the `StandardsArn` parameter to specify the Amazon Resource Name (ARN) of the standard that you want to enable. Also specify the Region that your request applies to. For example, the following command enables the AWS Foundational Security Best Practices (FSBP) standard:

```
$ aws securityhub batch-enable-standards \
--standards-subscription-requests '{"StandardsArn":"arn:aws:securityhub:us-east-1::standards/aws-foundational-security-best-practices/v/1.0.0"}' \
--region us-east-1
```

Where *arn:aws:securityhub:us-east-1::standards/aws-foundational-security-best-practices/v/1.0.0* is the ARN of the FSBP standard in the US East (N. Virginia) Region, and *us-east-1* is the Region in which to enable it.

To obtain the ARN for a standard, use the DescribeStandards operation or, if you're using the AWS CLI, run the describe-standards command.

To first review a list of standards that are currently enabled in your account, you can use the GetEnabledStandards operation. If you're using the AWS CLI, you can run the get-enabled-standards command to retrieve this list.

After you enable a standard, Security Hub CSPM begins performing tasks to enable the standard in the account and the specified Region. This includes creating all the controls that apply to the standard. To monitor the status of these tasks, you can check the status of the standard for the account and Region.

Checking the status of a standard

When you enable a security standard for an account, Security Hub CSPM begins creating all the controls that apply to the standard in the account. Security Hub CSPM also performs additional tasks to enable the standard for the account, such as generating a preliminary security score for

the standard. While Security Hub CSPM performs these tasks, the status of the standard is *Pending* for the account. The status of the standard then passes through additional states, which you can monitor and check.

> **Note**
>
> Changes to individual controls for a standard don't affect the overall status of the standard. For example, if you enable a control that you previously disabled, your change doesn't affect the status of the standard. Similarly, if you change a parameter value for an enabled control, your change doesn't affect the status of the standard.

To check the status of a standard by using the Security Hub CSPM console, choose **Security standards** in the navigation pane. The **Security standards** page lists all the standards that Security Hub CSPM currently supports. If Security Hub CSPM is currently performing tasks to enable the standard, the section for the standard indicates that Security Hub CSPM is still generating a security score for the standard. If a standard is enabled, the section for the standard includes the current score. Choose **View results** to review additional details, including the status of individual controls that apply to the standard. For more information, see Schedule for running security checks.

To check the status of a standard programmatically with the Security Hub CSPM API, use the GetEnabledStandards operation. In your request, optionally use the `StandardsSubscriptionArns` parameter to specify the Amazon Resource Name (ARN) of the standard whose status you want to check. If you're using the AWS Command Line Interface (AWS CLI), you can run the get-enabled-standards command to check the status of a standard. To specify the ARN of the standard to check, use the `standards-subscription-arns` parameter. To determine which ARN to specify, you can use the DescribeStandards operation or, for the AWS CLI, run the describe-standards command.

If your request succeeds, Security Hub CSPM responds with an array of `StandardsSubscription` objects. A *standard subscription* is an AWS resource that Security Hub CSPM creates in an account when a standard is enabled for the account. Each `StandardsSubscription` object provides details about a standard that is currently enabled or is being enabled or disabled for the account. Within each object, the `StandardsStatus` field specifies the current status of the standard for the account.

The status of a standard (`StandardsStatus`) can be one of the following.

PENDING

Security Hub CSPM is currently performing tasks to enable the standard for the account. This includes creating the controls that apply to the standard, and generating a preliminary security score for the standard. It can take several minutes for Security Hub CSPM to complete all the tasks. A standard can also have this status if it's already enabled for the account and Security Hub CSPM is currently adding new controls to the standard.

If a standard has this status, you might not be able to retrieve the details of individual controls that apply to the standard. In addition, you might not be able to configure or disable individual controls for the standard. For example, if you try to disable a control by using the UpdateStandardsControl operation, an error occurs.

To determine whether you can configure or otherwise manage individual controls for the standard, refer to the value for the `StandardsControlsUpdatable` field. If the value for this field is `READY_FOR_UPDATES`, you can start managing individual controls for the standard. Otherwise, wait until Security Hub CSPM completes additional processing tasks to enable the standard.

READY

The standard is currently enabled for the account. Security Hub CSPM can run security checks and generate findings for all the controls that apply to the standard and are currently enabled. Security Hub CSPM can also calculate a security score for the standard.

If a standard has this status, you can retrieve the details of individual controls that apply to the standard. In addition, you can configure, disable, or re-enable the controls. You can also disable the standard.

INCOMPLETE

Security Hub CSPM wasn't able to enable the standard completely for the account. Security Hub CSPM can't run security checks and generate findings for all the controls that apply to the standard and are currently enabled. In addition, Security Hub CSPM can't calculate a security score for the standard.

To determine why the standard wasn't enabled completely, refer to the information in the `StandardsStatusReason` array. This array specifies issues that prevented Security Hub CSPM from enabling the standard. If an internal error occurred, try enabling the standard for the account again. For other types of issues, check your AWS Config settings. You can also disable individual controls that you don't want to check, or disable the standard completely.

DELETING

Security Hub CSPM is currently processing a request to disable the standard for the account. This includes disabling the controls that apply to the standard, and removing the associated security score. It can take several minutes for Security Hub CSPM to finish processing the request.

If a standard has this status, you can't re-enable the standard or try to disable it again for the account. Security Hub CSPM must finish processing the current request first. In addition, you can't retrieve the details of individual controls that apply to the standard or manage the controls.

FAILED

Security Hub CSPM wasn't able to disable the standard for the account. One or more errors occurred when Security Hub CSPM attempted to disable the standard. In addition, Security Hub CSPM can't calculate a security score for the standard.

To determine why the standard wasn't disabled completely, refer to the information in the `StandardsStatusReason` array. This array specifies issues that prevented Security Hub CSPM from disabling the standard.

If a standard has this status, you can't retrieve the details of individual controls that apply to the standard or manage the controls. You can, however, re-enable the standard for the account. If you address the issues that prevented Security Hub CSPM from disabling the standard, you can also try to disable the standard again.

If the status of a standard is `READY`, Security Hub CSPM runs security checks and generates findings for all the controls that apply to the standard and are currently enabled. For other statuses, Security Hub CSPM might run checks and generate findings for some, but not all, enabled controls. It can take up to 24 hours to generate or update control findings. For more information, see Schedule for running security checks.

Reviewing the details of a security standard

After you enable a security standard in AWS Security Hub CSPM, you can use the console to review the details of the standard. On the console, the details page for a standard includes the following information:

- The current security score for the standard.

- A table of controls that apply to the standard.
- Aggregated statistics for controls that apply to the standard.
- A visual summary of the status of the controls that apply to the standard.
- A visual summary of security checks for controls that are enabled and apply to the standard. If you integrate with AWS Organizations, controls that are enabled in at least one organization account are considered enabled.

To review these details, choose **Security standards** in the navigation pane on the console. Then, in the section for the standard, choose **View results**. For deeper analysis, you can filter and sort the data, and drill down to review the details of individual controls that apply to the standard.

Topics

Understanding the standard security score

On the AWS Security Hub CSPM console, the details page for a standard displays the security score for the standard. The score is the percentage of controls that passed evaluation, relative to the total number of controls that apply to the standard, are enabled, and have evaluation data. Under the score is a chart that summarizes security checks for controls that are enabled for the standard. This includes the number of passed and failed security checks. For administrator accounts, the standard score and chart are aggregated across the administrator account and all member accounts. To review failed security checks for controls that have a specific severity, choose the severity.

When you enable a standard, Security Hub CSPM generates a preliminary security score for the standard, typically within 30 minutes of your first visit to the **Summary** page or the **Security standards** page on the Security Hub CSPM console. Scores are generated only for standards that are enabled when you visit those pages. In addition, AWS Config resource recording must be configured for the scores to appear. In the China Regions and AWS GovCloud (US) Regions, it can take up to 24 hours for Security Hub CSPM to generate a preliminary score. After Security Hub CSPM generates a preliminary score for a standard, it updates the score every 24 hours. For more information, see Calculating security scores.

All the data on **Security standards** detail pages is specific to the current AWS Region unless you set an aggregation Region. If you set an aggregation Region, security scores apply across Regions and include findings for all linked Regions. In addition, the compliance status of controls reflects findings from linked Regions, and the number of security checks includes findings from linked Regions.

Reviewing the controls for a standard

When you use the AWS Security Hub CSPM console to review the details of a standard that you enabled, you can review a table of security controls that apply to the standard. For each control, the table includes the following information:

- The control ID and title.
- The status of the control. For more information, see Evaluating compliance status and control status.
- The severity assigned to the control.
- The number of failed checks and the total number of checks. If applicable, the **Failed checks** field also specifies the number of findings with a status of **Unknown**.
- Whether the control supports custom parameters. For more information, see Understanding control parameters in Security Hub CSPM.

Security Hub CSPM updates control statuses and the count of security checks every 24 hours. A timestamp at the top of the page indicates when Security Hub CSPM most recently updated this data.

For administrator accounts, control statuses and the number of security checks are aggregated across the administrator account and all member accounts. The count of enabled controls includes controls that are enabled for the standard in the administrator account or at least one member account. The count of disabled controls includes controls that are disabled for the standard in the administrator account and all member accounts.

You can filter the table of controls that apply to the standard. Using the **Filter by** options next to the table, you can choose to view only enabled or only disabled controls for the standard. If you display only enabled controls, you can further filter the table by control status. You can then focus on controls that have a specific control status. In addition to the **Filter by** options, you can enter filter criteria in the **Filter controls** box. For example, you can filter by control ID or title.

Choose your preferred access method. Then follow the steps to review the controls that apply to a standard that you enabled.

Security Hub CSPM console

To review the controls for an enabled standard

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Choose **Security standards** in the navigation pane.
3. In the section for the standard, choose **View results**.

The table at the bottom of the page lists all the controls that apply to the standard. You can filter and sort the table. You can also download the current page of the table as a CSV file. To do this, choose **Download** above the table. If you filter the table, the downloaded file includes only the controls that match your current filter settings.

Security Hub CSPM API

To review the controls for an enabled standard

1. Use the ListSecurityControlDefinitions operation of the Security Hub CSPM API. If you're using the AWS CLI, run the list-security-control-definitions command.

   Specify the Amazon Resource Name (ARN) of the standard that you want to review controls for. To obtain ARNs for standards, use the DescribeStandards operation or run the describe-standards command. If you don't specify the ARN for a standard, Security Hub CSPM returns all security control IDs.
2. Use the ListStandardsControlAssociations operation of the Security Hub CSPM API, or run the list-standards-control-associations command. This operation tells you which standards a control is enabled in.

   Identify the control by providing the security control ID or ARN. Pagination parameters are optional.

The following example tells you which standards the Config.1 control is enabled in.

```
$ aws securityhub list-standards-control-associations --region us-east-1 --security-control-id Config.1
```

Turning off automatically enabled security standards

If your organization doesn't use central configuration, it uses a configuration type called *local configuration*. With local configuration, AWS Security Hub CSPM can automatically enable default security standards for new member accounts when the accounts join your organization. All the controls that apply to these default standards are also enabled automatically.

Currently, the default security standards are the AWS Foundational Security Best Practices standard and the Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0 standard. For information about these standards, see the Standards reference for Security Hub CSPM.

If you prefer to manually enable security standards for new member accounts, you can turn off automatic enablement of the default standards. You can do this only if you integrate with AWS Organizations and use local configuration. If you use central configuration, you can instead create a configuration policy that enables the default standards and associate the policy with the root. All of your organization accounts and OUs then inherit this configuration policy unless they are associated with a different policy or are self-managed. If you don't integrate with AWS Organizations, you can disable a default standard when you initially enable Security Hub CSPM or later. To learn how, see Disabling a standard.

To turn off automatic enablement of the default standards for new member accounts, you can use the Security Hub CSPM console or the Security Hub CSPM API.

Security Hub CSPM console

Follow these steps to turn off automatic enablement of the default standards by using the Security Hub CSPM console.

To turn off automatic enablement of default standards

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the administrator account.
2. In the navigation pane, under **Settings**, choose **Configuration**.

3. In the **Overview** section, choose **Edit**.
4. Under **New account settings**, clear the **Enable the default security standards** checkbox.
5. Choose **Confirm**.

Security Hub CSPM API

To turn off automatic enablement of the default standards programmatically, from the Security Hub CSPM administrator account, use the UpdateOrganizationConfiguration operation of the Security Hub CSPM API. In your request, specify `NONE` for the `AutoEnableStandards` parameter.

If you're using the AWS CLI, run the update-organization-configuration command to turn off automatic enablement of the default standards. For the `auto-enable-standards` parameter, specify `NONE`. For example, the following command automatically enables Security Hub CSPM for new member accounts, and turns off automatic enablement of the default standards for the accounts.

```
$ aws securityhub update-organization-configuration --auto-enable --auto-enable-
standards NONE
```

Disabling a security standard

When you disable a security standard in AWS Security Hub CSPM, the following occurs:

- All the controls that apply to the standard are disabled, unless they're associated with another standard that's currently enabled.
- Security checks for the disabled controls are no longer performed, and no additional findings are generated for the disabled controls.
- Existing findings for the disabled controls are archived automatically after approximately 3-5 days.
- AWS Config rules that Security Hub CSPM created for the disabled controls are deleted.

Deletion of the appropriate AWS Config rules typically occurs within a few minutes of disabling a standard. However, it might take longer. If the first request fails to delete the rules, Security Hub CSPM tries again every 12 hours. However, if you disabled Security Hub CSPM or don't have any

other standards enabled, Security Hub CSPM can't try again, which means that it can't delete the rules. If this occurs and you need to delete the rules, contact AWS Support.

Topics

Disabling a standard in multiple accounts and AWS Regions

To disable a security standard across multiple accounts and AWS Regions, use central configuration. With central configuration, the delegated Security Hub CSPM administrator can create Security Hub CSPM configuration policies that disable one or more standards. The administrator can then associate a configuration policy with individual accounts, organizational units (OUs), or the root. A configuration policy affects the home Region, also referred to as an *aggregation Region*, and all linked Regions.

Configuration policies offer customization options. For example, you might choose to disable the Payment Card Industry Data Security Standard (PCI DSS) in one OU. For another OU, you might choose to disable both the PCI DSS and the National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5 standard. For information about creating a configuration policy that enables or disables individual standards that you specify, see Creating and associating configuration policies.

> **Note**
>
> The Security Hub CSPM administrator can use configuration policies to disable any standard except the AWS Control Tower service-managed standard. To disable this standard, the administrator must use AWS Control Tower directly. They must also use AWS Control Tower to disable or enable individual controls in this standard for a centrally managed account.

If you want some accounts to configure or disable standards for their own accounts, the Security Hub CSPM administrator can designate those accounts as *self-managed accounts*. Self-managed accounts must disable standards separately in each Region.

Disabling a standard in a single account and AWS Region

If you don't use central configuration or you have a self-managed account, you can't use configuration policies to centrally disable security standards in multiple accounts or AWS Regions. However, you can disable a standard in a single account and Region. You can do this by using the Security Hub CSPM console or the Security Hub CSPM API.

Security Hub CSPM console

Follow these steps to disable a standard in one account and Region by using the Security Hub CSPM console.

To disable a standard in one account and Region

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. By using the AWS Region selector in the upper-right corner of the page, choose the Region in which you want to disable the standard.
3. In the navigation pane, choose **Security standards**.
4. In the section for the standard that you want to disable, choose **Disable standard**.

To disable the standard in additional Regions, repeat the preceding steps in each additional Region.

Security Hub CSPM API

To disable a standard programmatically in a single account and Region, use the BatchDisableStandards operation. Or, if you're using the AWS Command Line Interface (AWS CLI), run the batch-disable-standards command.

In your request, use the `StandardsSubscriptionArns` parameter to specify the Amazon Resource Name (ARN) of the standard that you want to disable. If you're using the AWS CLI, use the `standards-subscription-arns` parameter to specify the ARN. Also specify the Region that your request applies to. For example, the following command disables the AWS Foundational Security Best Practices (FSBP) standard for an account (*123456789012*):

```
$ aws securityhub batch-disable-standards \
--standards-subscription-arns "arn:aws:securityhub:us-
east-1:123456789012:subscription/aws-foundational-security-best-practices/v/1.0.0" \
```

```
--region us-east-1
```

Where *arn:aws:securityhub:us-east-1:123456789012:subscription/aws-foundational-security-best-practices/v/1.0.0* is the ARN of the FSBP standard for the account in the US East (N. Virginia) Region, and *us-east-1* is the Region in which to disable it.

To obtain the ARN for a standard, you can use the GetEnabledStandards operation. This operation retrieves information about the standards that are currently enabled in your account. If you're using the AWS CLI, you can run the get-enabled-standards command to retrieve this information.

After you disable a standard, Security Hub CSPM begins performing tasks to disable the standard in the account and the specified Region. This includes disabling all the controls that apply to the standard. To monitor the status of these tasks, you can check the status of the standard for the account and Region.

Understanding security controls in Security Hub CSPM

In AWS Security Hub CSPM, a *security control*, also referred to as a *control*, is a safeguard within a security standard that helps an organization protect the confidentiality, integrity, and availability of its information. In Security Hub CSPM, a control is related to a specific AWS resource.

When you enable a control in one or more standards, Security Hub CSPM begins running security checks on it. The security checks result in Security Hub CSPM findings. When you disable a control, Security Hub CSPM stops running security checks on it, and findings are no longer generated.

You can enable or disable controls individually for a single account and AWS Region. To save time and reduce configuration drift in multi-account environments, we recommend using central configuration to enable or disable controls. With central configuration, the delegated Security Hub CSPM administrator can create policies that specify how a control should be configured across multiple accounts and Regions. For more information about enabling and disabling controls, see Enabling controls in Security Hub CSPM.

Consolidated controls view

The **Controls** page of the Security Hub CSPM console displays all of the controls available in the current AWS Region (you can view controls in the context of a standard by visiting the **Security**

standards page and choosing an enabled standard). Security Hub CSPM assigns controls a consistent security control ID, title, and description across standards. Controls IDs include the relevant AWS service and a unique number (for example, CodeBuild.3).

The following information is available on the **Controls** page of the Security Hub CSPM console:

- An overall security score based on the proportion of passed controls compared to the total number of enabled controls with data
- Breakdown of control statuses across all supported Security Hub CSPM controls
- The number of total passed and failed security checks.
- The number of failed security checks for controls of varying severity, and links to view more details about those failed checks.
- A list of Security Hub CSPM controls, with filters to view specific subsets of controls.

From the **Controls** page, you can choose a control to view its details and take action on the findings generated by the control. From this page, you can also enable or disable a security control in your current AWS account and AWS Region. Enablement and disablement actions from the **Controls** page apply across standards. For more information, see Enabling controls in Security Hub CSPM.

For administrator accounts, the **Controls** page reflects the status of controls across the member accounts. If a control check fails in at least one member account, the control status is **Failed**. If you have set an aggregation Region, the **Controls** page reflects the status of controls across all linked Regions. If a control check fails in at least one linked Region, the control status is **Failed**.

Consolidated controls view causes changes to control finding fields in the AWS Security Finding Format (ASFF) that may affect workflows. For more information, see Consolidated controls view – ASFF changes.

Summary security score for controls

The **Controls** page displays a summary security score from 0–100 percent. The summary security score is calculated based on the proportion of passed controls compared to the total number of enabled controls with data across standards.

> **Note**
>
> To view the overall security score for controls, you must add permission to call **`BatchGetControlEvaluations`** to the IAM role that you use to access Security Hub CSPM. This permission isn't required to view security scores for specific standards.

When you enable Security Hub CSPM, Security Hub CSPM calculates the initial security score within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub CSPM console. It can take up to 24 hours for first-time security scores to be generated in the China Regions and AWS GovCloud (US) Regions.

In addition to the overall security score, Security Hub CSPM calculates a standard security score for each enabled standard within 30 minutes after your first visit to the **Summary** page or **Security standards** page. To view a list of standards that are currently enabled, use the `GetEnabledStandards` API operation.

AWS Config must be enabled with resource recording for scores to appear. For information about how Security Hub CSPM calculates security scores, see Calculating security scores.

After first-time score generation, Security Hub CSPM updates security scores every 24 hours. Security Hub CSPM displays a timestamp to indicate when a security score was last updated.

If you have set an aggregation Region, the overall security score reflects control findings across linked Regions.

Control reference for Security Hub CSPM

This control reference provides a table of available AWS Security Hub CSPM controls with links to more information about each control. In the table, controls are listed in alphabetical order by control ID. Only controls in active use by Security Hub CSPM are included here. Retired controls are excluded from the table.

The table provides the following information for each control:

- **Security control ID** – This ID applies across standards and indicates the AWS service and resource that the control relates to. The Security Hub CSPM console displays security control IDs, regardless of whether consolidated control findings is turned on or off in your account. However, Security Hub CSPM findings reference security control IDs only if consolidated control findings

is turned on in your account. If consolidated control findings is turned off in your account, some control IDs vary by standard in your control findings. For a mapping of standard-specific control IDs to security control IDs, see How consolidation impacts control IDs and titles.

If you want to set up automations for security controls, we recommend filtering based on control ID rather than title or description. Whereas Security Hub CSPM may occasionally update control titles or descriptions, control IDs stay the same.

Control IDs may skip numbers. These are placeholders for future controls.

- **Security control title** – This title applies across standards. The Security Hub CSPM console displays security control titles, regardless of whether consolidated control findings is turned on or off in your account. However, Security Hub CSPM findings reference security control titles only if consolidated control findings is turned on in your account. If consolidated control findings is turned off in your account, some control titles vary by standard in your control findings. For a mapping of standard-specific control IDs to security control IDs, see How consolidation impacts control IDs and titles.
- **Applicable standards** – Indicates which standards a control applies to. Choose a control to review specific requirements from third-party compliance frameworks.
- **Severity** – The severity of a control identifies its importance from a security standpoint. For information about how Security Hub CSPM determines control severity, see Severity levels for control findings.
- **Supports custom parameters** – Indicates whether the control supports custom values for one or more parameters. Choose a control to review the parameter details. For more information, see Understanding control parameters in Security Hub CSPM.
- **Schedule type** – Indicates when the control is evaluated. For more information, see Schedule for running security checks.

Choose a control to review additional details. Controls are listed in alphabetical order by security control ID.

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Account.1	Security contact information should be provided for an AWS account	CIS AWS Foundations Benchmark v3.0.0, AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	No	Periodic
Account.2	AWS account should be part of an AWS Organizations organization	NIST SP 800-53 Rev. 5	HIGH	No	Periodic
ACM.1	Imported and ACM-issued certificates should be renewed after a specified time period	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1	MEDIUM	Yes	Change triggered and periodic
ACM.2	RSA certificates managed by ACM should use a key length of at least 2,048 bits	AWS Foundational Security Best Practices v1.0.0, PCI DSS v4.0.1	HIGH	No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ACM.3	ACM certificates should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Amplify.1	Amplify apps should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Amplify.2	Amplify branches should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
APIGateway.1	API Gateway REST and WebSocket API execution logging should be enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered
APIGateway.2	API Gateway REST API stages should be configured to use SSL certificates for backend authentication	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
APIGateway.3	API Gateway REST API stages should have AWS X-Ray tracing enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
APIGateway.4	API Gateway should be associated with a WAF Web ACL	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
APIGateway.5	API Gateway REST API cache data should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
APIGateway.8	API Gateway routes should specify an authorization type	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
APIGateway.9	Access logging should be configured for API Gateway V2 Stages	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
AppConfig.1	AWS AppConfig applications should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
AppConfig.2	AWS AppConfig configuration profiles should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
AppConfig.3	AWS AppConfig environments should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
AppConfig.4	AWS AppConfig extension associations should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
AppFlow.1	Amazon AppFlow flows should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
AppRunner.1	App Runner services should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
AppRunner.2	App Runner VPC connectors should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
AppSync.1	AWS AppSync API caches should be encrypted at rest	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
AppSync.2	AWS AppSync should have field-level logging enabled	AWS Foundational Security Best Practices v1.0.0, PCI DSS v4.0.1	MEDIUM	 Yes	Change triggered
AppSync.4	AWS AppSync GraphQL APIs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
AppSync.5	AWS AppSync GraphQL APIs should not be authenticated with API keys	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
AppSync.6	AWS AppSync API caches should be encrypted in transit	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
Athena.2	Athena data catalogs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Athena.3	Athena workgroups should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Athena.4	Athena workgroups should have logging enabled	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
AutoScaling.1	Auto Scaling groups associated with a load balancer should use ELB health checks	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
AutoScaling.2	Amazon EC2 Auto Scaling group should cover multiple Availability Zones	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered
AutoScaling.3	Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	HIGH	 No	Change triggered
Autoscaling.5	Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
AutoScaling.6	Auto Scaling groups should use multiple instance types in multiple Availability Zones	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
AutoScaling.9	EC2 Auto Scaling groups should use EC2 launch templates	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
AutoScaling.10	EC2 Auto Scaling groups should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Backup.1	AWS Backup recovery points should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Backup.2	AWS Backup recovery points should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Backup.3	AWS Backup vaults should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Backup.4	AWS Backup report plans should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Backup.5	AWS Backup backup plans should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Batch.1	Batch job queues should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Batch.2	Batch scheduling policies should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Batch.3	Batch compute environments should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Batch.4	Compute resources properties in managed Batch compute environments should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
CloudFormation.2	CloudFormation stacks should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
CloudFront.1	CloudFront distributions should have a default root object configured	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	HIGH	 No	Change triggered
CloudFront.3	CloudFront distributions should require encryption in transit	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
CloudFront.4	CloudFront distributions should have origin failover configured	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudFront.5	CloudFront distributions should have logging enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
CloudFront.6	CloudFront distributions should have WAF enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
CloudFront.7	CloudFront distributions should use custom SSL/TLS certificates	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Change triggered
CloudFront.8	CloudFront distributions should use SNI to serve HTTPS requests	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
CloudFront.9	CloudFront distributions should encrypt traffic to custom origins	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudFront.10	CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
CloudFront.12	CloudFront distributions should not point to non-existent S3 origins	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	HIGH	 No	Periodic
CloudFront.13	CloudFront distributions should use origin access control	AWS Foundational Security Best Practices v1.0.0	MEDIUM	 No	Change triggered
CloudFront.14	CloudFront distributions should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
CloudFront.15	CloudFront distributions should use the recommended TLS security policy	AWS Foundational Security Best Practices v1.0.0	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudFront.16	CloudFront distributions should use origin access control for Lambda function URL origins	AWS Foundational Security Best Practices v1.0.0	MEDIUM	 No	Change triggered
CloudTrail.1	CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudTrail.2	CloudTrail should have encryption at-rest enabled	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0 AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Periodic
CloudTrail.3	At least one CloudTrail trail should be enabled	NIST SP 800-171 Rev. 2, PCI DSS v4.0.1, PCI DSS v3.2.1	HIGH	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudTrail.4	CloudTrail log file validation should be enabled	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1, PCI DSS v3.2.1, Service-Managed Standard: AWS Control Tower	LOW	 No	Periodic
CloudTrail.5	CloudTrail trails should be integrated with Amazon CloudWatch Logs	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0, AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	LOW	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudTrail.6	Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0, PCI DSS v4.0.1	CRITICAL	 No	Change triggered and periodic
CloudTrail.7	Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v3.0.0, PCI DSS v4.0.1	LOW	 No	Periodic
CloudTrail.9	CloudTrail trails should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
CloudTrail.10	CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
CloudWatch.1	A log metric filter and alarm should exist for usage of the "root" user	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2, PCI DSS v3.2.1	LOW	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudWatch.2	Ensure a log metric filter and alarm exist for unauthorized API calls	CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	No	Periodic
CloudWatch.3	Ensure a log metric filter and alarm exist for Management Console sign-in without MFA	CIS AWS Foundations Benchmark v1.2.0	LOW	No	Periodic
CloudWatch.4	Ensure a log metric filter and alarm exist for IAM policy changes	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	No	Periodic
CloudWatch.5	Ensure a log metric filter and alarm exist for CloudTrail configuration changes	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	No	Periodic
CloudWatch.6	Ensure a log metric filter and alarm exist for AWS Management Console authentication failures	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudWatch.7	Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	 No	Periodic
CloudWatch.8	Ensure a log metric filter and alarm exist for S3 bucket policy changes	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	 No	Periodic
CloudWatch.9	Ensure a log metric filter and alarm exist for AWS Config configuration changes	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	 No	Periodic
CloudWatch.10	Ensure a log metric filter and alarm exist for security group changes	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudWatch.11	Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	 No	Periodic
CloudWatch.12	Ensure a log metric filter and alarm exist for changes to network gateways	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	 No	Periodic
CloudWatch.13	Ensure a log metric filter and alarm exist for route table changes	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	 No	Periodic
CloudWatch.14	Ensure a log metric filter and alarm exist for VPC changes	CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	LOW	 No	Periodic
CloudWatch.15	CloudWatch alarms should have specified actions configured	NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	HIGH	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudWatch.16	CloudWatch log groups should be retained for a specified time period	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
CloudWatch.17	CloudWatch alarm actions should be enabled	NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
CodeArtifact.1	CodeArtifact repositories should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
CodeBuild.1	CodeBuild Bitbucket source repository URLs should not contain sensitive credentials	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	CRITICAL	 No	Change triggered
CodeBuild.2	CodeBuild project environment variables should not contain clear text credentials	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	CRITICAL	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CodeBuild.3	CodeBuild S3 logs should be encrypted	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower,	LOW	 No	Change triggered
CodeBuild.4	CodeBuild project environments should have a logging configuration	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
CodeBuild.7	CodeBuild report group exports should be encrypted at rest	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
CodeGuruProfiler.1	CodeGuru Profiler profiling groups should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
CodeGuruReviewer.1	CodeGuru Reviewer repository associations should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Cognito.1	Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication	AWS Foundational Security Best Practices	MEDIUM	 Yes	Change triggered
Cognito.2	Cognito identity pools should not allow unauthenticated identities	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
Config.1	AWS Config should be enabled and use the service-linked role for resource recording	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1	CRITICAL	 Yes	Periodic
Connect.1	Amazon Connect Customer Profiles object types should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Connect.2	Amazon Connect instances should have CloudWatch logging enabled	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
DataFirehose.1	Firehose delivery streams should be encrypted at rest	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
DataSync.1	DataSync tasks should have logging enabled	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
DataSync.2	DataSync tasks should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Detective.1	Detective behavior graphs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
DMS.1	Database Migration Service replication instances should not be public	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	CRITICAL	 No	Periodic
DMS.2	DMS certificates should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
DMS.3	DMS event subscriptions should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
DMS.4	DMS replication instances should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
DMS.5	DMS replication subnet groups should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
DMS.6	DMS replication instances should have automatic minor version upgrade enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
DMS.7	DMS replication tasks for the target database should have logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
DMS.8	DMS replication tasks for the source database should have logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
DMS.9	DMS endpoints should use SSL	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
DMS.10	DMS endpoints for Neptune databases should have IAM authorization enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
DMS.11	DMS endpoints for MongoDB should have an authentication mechanism enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
DMS.12	DMS endpoints for Redis OSS should have TLS enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
DocumentDB.1	Amazon DocumentDB clusters should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
DocumentDB.2	Amazon DocumentDB clusters should have an adequate backup retention period	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
DocumentDB.3	Amazon DocumentDB manual cluster snapshots should not be public	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	CRITICAL	 No	Change triggered
DocumentDB.4	Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
DocumentDB.5	Amazon DocumentDB clusters should have deletion protection enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
DocumentDB.6	Amazon DocumentDB clusters should be encrypted in transit	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
DynamoDB1	DynamoDB tables should automatically scale capacity with demand	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
DynamoDB 2	DynamoDB tables should have point-in-time recovery enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
DynamoDB 3	DynamoDB Accelerator (DAX) clusters should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
DynamoDB 4	DynamoDB tables should be present in a backup plan	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
DynamoDB 5	DynamoDB tables should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
DynamoDB 6	DynamoDB tables should have deletion protection enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
DynamoDB 7	DynamoDB Accelerator clusters should be encrypted in transit	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Periodic
EC2.1	EBS snapshots should not be publicly restorable	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	CRITICAL	 No	Periodic
EC2.2	VPC default security groups should not allow inbound or outbound traffic	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.3	Attached EBS volumes should be encrypted at-rest	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
EC2.4	Stopped EC2 instances should be removed after a specified time period	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
EC2.6	VPC flow logging should be enabled in all VPCs	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.7	EBS default encryption should be enabled	CIS AWS Foundations Benchmark v3.0.0, AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
EC2.8	EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)	CIS AWS Foundations Benchmark v3.0.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Change triggered
EC2.9	EC2 instances should not have a public IPv4 address	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.10	Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Periodic
EC2.12	Unused EC2 EIPs should be removed	PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
EC2.13	Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22	CIS AWS Foundations Benchmark v1.2.0, PCI DSS v3.2.1, PCI DSS v4.0.1, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	HIGH	 No	Change triggered and periodic
EC2.14	Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389	CIS AWS Foundations Benchmark v1.2.0, PCI DSS v4.0.1	HIGH	 No	Change triggered and periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.15	EC2 subnets should not automatically assign public IP addresses	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower,	MEDIUM	 No	Change triggered
EC2.16	Unused Network Access Control Lists should be removed	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower,	LOW	 No	Change triggered
EC2.17	EC2 instances should not use multiple ENIs	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.18	Security groups should only allow unrestricted incoming traffic for authorized ports	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	HIGH	 Yes	Change triggered
EC2.19	Security groups should not allow unrestricted access to ports with high risk	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	CRITICAL	 No	Change triggered and periodic
EC2.20	Both VPN tunnels for an AWS Site-to-Site VPN connection should be up	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.21	Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
EC2.22	Unused EC2 security groups should be removed	Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Periodic
EC2.23	EC2 Transit Gateways should not automatically accept VPC attachment requests	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
EC2.24	EC2 paravirtual instance types should not be used	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.25	EC2 launch templates should not assign public IPs to network interfaces	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Change triggered
EC2.28	EBS volumes should be in a backup plan	NIST SP 800-53 Rev. 5	LOW	 Yes	Periodic
EC2.33	EC2 transit gateway attachments should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.34	EC2 transit gateway route tables should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.35	EC2 network interfaces should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.36	EC2 customer gateways should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.37	EC2 Elastic IP addresses should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.38	EC2 instances should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.39	EC2 internet gateways should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.40	EC2 NAT gateways should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.41	EC2 network ACLs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.42	EC2 route tables should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.43	EC2 security groups should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.44	EC2 subnets should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.45	EC2 volumes should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.46	Amazon VPCs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.47	Amazon VPC endpoint services should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.48	Amazon VPC flow logs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.49	Amazon VPC peering connections should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.50	EC2 VPN gateways should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.51	EC2 Client VPN endpoints should have client connection logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1	LOW	 No	Change triggered
EC2.52	EC2 transit gateways should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.53	EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports	CIS AWS Foundations Benchmark v3.0.0, PCI DSS v4.0.1	HIGH	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.54	EC2 security groups should not allow ingress from ::/0 to remote server administration ports	CIS AWS Foundations Benchmark v3.0.0, PCI DSS v4.0.1	HIGH	 No	Periodic
EC2.55	VPCs should be configured with an interface endpoint for ECR API	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
EC2.56	VPCs should be configured with an interface endpoint for Docker Registry	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
EC2.57	VPCs should be configured with an interface endpoint for Systems Manager	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
EC2.58	VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.60	VPCs should be configured with an interface endpoint for Systems Manager Incident Manager	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
EC2.170	EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)	AWS Foundational Security Best Practices, PCI DSS v4.0.1	LOW	 No	Change triggered
EC2.171	EC2 VPN connections should have logging enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
EC2.172	EC2 VPC Block Public Access settings should block internet gateway traffic	AWS Foundational Security Best Practices	MEDIUM	 Yes	Change triggered
EC2.173	EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
EC2.174	EC2 DHCP option sets should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.175	EC2 launch templates should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.176	EC2 prefix lists should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.177	EC2 traffic mirror sessions should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.178	EC2 traffic mirror filters should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.179	EC2 traffic mirror targets should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EC2.180	EC2 network interfaces should have source/destination checking enabled	AWS Foundational Security Best Practices v1.0.0	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ECR.1	ECR private repositories should have image scanning configured	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Periodic
ECR.2	ECR private repositories should have tag immutability configured	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ECR.3	ECR repositories should have at least one lifecycle policy configured	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ECR.4	ECR public repositories should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ECR.5	ECR repositories should be encrypted with customer managed AWS KMS keys	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered
ECS.1	Amazon ECS task definitions should have secure networking modes and user definitions.	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
ECS.2	ECS services should not have public IP addresses assigned to them automatically	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Change triggered
ECS.3	ECS task definitions should not share the host's process namespace	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ECS.4	ECS containers should run as non-privileged	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
ECS.5	ECS containers should be limited to read-only access to root filesystems	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
ECS.8	Secrets should not be passed as container environment variables	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Change triggered
ECS.9	ECS task definitions should have a logging configuration	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ECS.10	ECS Fargate services should run on the latest Fargate platform version	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
ECS.12	ECS clusters should use Container Insights	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ECS.13	ECS services should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
ECS.14	ECS clusters should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
ECS.15	ECS task definitions should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ECS.16	ECS task sets should not automatically assign public IP addresses	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 No	Change triggered
ECS.17	ECS task definitions should not use host network mode	NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
EFS.1	Elastic File System should be configured to encrypt file data at-rest using AWS KMS	CIS AWS Foundations Benchmark v3.0.0, AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
EFS.2	Amazon EFS volumes should be in backup plans	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EFS.3	EFS access points should enforce a root directory	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
EFS.4	EFS access points should enforce a user identity	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
EFS.5	EFS access points should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EFS.6	EFS mount targets should not be associated with subnets that assign public IP addresses on launch	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
EFS.7	EFS file systems should have automatic backups enabled	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EFS.8	EFS file systems should be encrypted at rest	AWS Foundational Security Best Practices	MEDIUM	 Yes	Change triggered
EKS.1	EKS cluster endpoints should not be publicly accessible	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	HIGH	 No	Periodic
EKS.2	EKS clusters should run on a supported Kubernetes version	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Change triggered
EKS.3	EKS clusters should use encrypted Kubernetes secrets	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Periodic
EKS.6	EKS clusters should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EKS.7	EKS identity provider configurations should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
EKS.8	EKS clusters should have audit logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
ElastiCache.1	ElastiCache (Redis OSS) clusters should have automatic backups enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	HIGH	 Yes	Periodic
ElastiCache.2	ElastiCache clusters should have automatic minor version upgrades enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	HIGH	 No	Periodic
ElastiCache.3	ElastiCache replication groups should have automatic failover enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
ElastiCache.4	ElastiCache replication groups should be encrypted-at-rest	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ElastiCache.5	ElastiCache replication groups should be encrypted-in-transit	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Periodic
ElastiCache.6	ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Periodic
ElastiCache.7	ElastiCache clusters should not use the default subnet group	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	HIGH	 No	Periodic
ElasticBeanstalk.1	Elastic Beanstalk environments should have enhanced health reporting enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ElasticBeanstalk.2	Elastic Beanstalk managed platform updates should be enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 Yes	Change triggered
ElasticBeanstalk.3	Elastic Beanstalk should stream logs to CloudWatch	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 Yes	Change triggered
ELB.1	Application Load Balancer should be configured to redirect all HTTP requests to HTTPS	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
ELB.2	Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ELB.3	Classic Load Balancer listeners should be configured with HTTPS or TLS termination	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
ELB.4	Application Load Balancer should be configured to drop http headers	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
ELB.5	Application and Classic Load Balancers logging should be enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ELB.6	Application, Gateway, and Network Load Balancers should have deletion protection enabled	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ELB.7	Classic Load Balancers should have connection draining enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ELB.8	Classic Load Balancers with SSL listeners should use a predefined security policy that has strong configuration	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ELB.9	Classic Load Balancers should have cross-zone load balancing enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ELB.10	Classic Load Balancer should span multiple Availability Zones	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered
ELB.12	Application Load Balancer should be configured with defensive or strictest desync mitigation mode	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
ELB.13	Application, Network and Gateway Load Balancers should span multiple Availability Zones	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ELB.14	Classic Load Balancer should be configured with defensive or strictest desync mitigation mode	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
ELB.16	Application Load Balancers should be associated with an AWS WAF web ACL	NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ELB.17	Application and Network Load Balancers with listeners should use recommended security policies	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ELB.18	Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit	AWS Foundational Security Best Practices v1.0.0	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EMR.1	Amazon EMR cluster primary nodes should not have public IP addresses	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Periodic
EMR.2	Amazon EMR block public access setting should be enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	CRITICAL	 No	Periodic
EMR.3	Amazon EMR security configurations should be encrypted at rest	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
EMR.4	Amazon EMR security configurations should be encrypted in transit	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ES.1	Elasticsearch domains should have encryption at-rest enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ES.2	Elasticsearch domains should not be publicly accessible	AWS Foundational Security Best Practices, PCI DSS v3.2.1, PCI DSS v4.0.1, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	CRITICAL	 No	Periodic
ES.3	Elasticsearch domains should encrypt data sent between nodes	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower,	MEDIUM	 No	Change triggered
ES.4	Elasticsearch domain error logging to CloudWatch Logs should be enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ES.5	Elasticsearch domains should have audit logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
ES.6	Elasticsearch domains should have at least three data nodes	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ES.7	Elasticsearch domains should be configured with at least three dedicated master nodes	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
ES.8	Connections to Elasticsearch domains should be encrypted using the latest TLS security policy	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ES.9	Elasticsearch domains should be tagged	AWS Resource Tagging Standard	LOW	Yes	Change triggered
EventBridge.2	EventBridge event buses should be tagged	AWS Resource Tagging Standard	LOW	Yes	Change triggered
EventBridge.3	EventBridge custom event buses should have a resource-based policy attached	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	LOW	No	Change triggered
EventBridge.4	EventBridge global endpoints should have event replication enabled	NIST SP 800-53 Rev. 5	MEDIUM	No	Change triggered
FraudDetector.1	Amazon Fraud Detector entity types should be tagged	AWS Resource Tagging Standard	LOW	Yes	Change triggered
FraudDetector.2	Amazon Fraud Detector labels should be tagged	AWS Resource Tagging Standard	LOW	Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
FraudDetector.3	Amazon Fraud Detector outcomes should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
FraudDetector.4	Amazon Fraud Detector variables should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
FSx.1	FSx for OpenZFS file systems should be configured to copy tags to backups and volumes	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	LOW	 No	Periodic
FSx.2	FSx for Lustre file systems should be configured to copy tags to backups	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	LOW	 No	Periodic
FSx.3	FSx for OpenZFS file systems should be configured for Multi-AZ deployment	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
FSx.4	FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment	AWS Foundational Security Best Practices	MEDIUM	 Yes	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
FSx.5	FSx for Windows File Server file systems should be configured for Multi-AZ deployment	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
Glue.1	AWS Glue jobs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Glue.3	AWS Glue machine learning transforms should be encrypted at rest	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
Glue.4	AWS Glue Spark jobs should run on supported versions of AWS Glue	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
GlobalAccelerator.1	Global Accelerator accelerators should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
GuardDuty.1	GuardDuty should be enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Periodic
GuardDuty.2	GuardDuty filters should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
GuardDuty.3	GuardDuty IPSets should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
GuardDuty.4	GuardDuty detectors should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
GuardDuty.5	GuardDuty EKS Audit Log Monitoring should be enabled	AWS Foundational Security Best Practices	HIGH	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
GuardDuty.6	GuardDuty Lambda Protection should be enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 No	Periodic
GuardDuty.7	GuardDuty EKS Runtime Monitoring should be enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	MEDIUM	 No	Periodic
GuardDuty.8	GuardDuty Malware Protection for EC2 should be enabled	AWS Foundational Security Best Practices	HIGH	 No	Periodic
GuardDuty.9	GuardDuty RDS Protection should be enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 No	Periodic
GuardDuty.10	GuardDuty S3 Protection should be enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 No	Periodic
GuardDuty.11	GuardDuty Runtime Monitoring should be enabled	AWS Foundational Security Best Practices	HIGH	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
GuardDuty.12	GuardDuty ECS Runtime Monitoring should be enabled	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
GuardDuty.13	GuardDuty EC2 Runtime Monitoring should be enabled	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
IAM.1	IAM policies should not allow full "*" administrative privileges	CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.2	IAM users should not have IAM policies attached	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	LOW	 No	Change triggered
IAM.3	IAM users' access keys should be rotated every 90 days or less	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.4	IAM root user access key should not exist	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	CRITICAL	 No	Periodic
IAM.5	MFA should be enabled for all IAM users that have a console password	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.6	Hardware MFA should be enabled for the root user	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	CRITICAL	 No	Periodic
IAM.7	Password policies for IAM users should have strong configurations	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 Yes	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.8	Unused IAM user credentials should be removed	CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Periodic
IAM.9	MFA should be enabled for the root user	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1	CRITICAL	 No	Periodic
IAM.10	Password policies for IAM users should have strong configurations	NIST SP 800-171 Rev. 2, PCI DSS v3.2.1	MEDIUM	 No	Periodic
IAM.11	Ensure IAM password policy requires at least one uppercase letter	CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.12	Ensure IAM password policy requires at least one lowercase letter	CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1	MEDIUM	 No	Periodic
IAM.13	Ensure IAM password policy requires at least one symbol	CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	MEDIUM	 No	Periodic
IAM.14	Ensure IAM password policy requires at least one number	CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1	MEDIUM	 No	Periodic
IAM.15	Ensure IAM password policy requires minimum password length of 14 or greater	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2	MEDIUM	 No	Periodic
IAM.16	Ensure IAM password policy prevents password reuse	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1	LOW	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.17	Ensure IAM password policy expires passwords within 90 days or less	CIS AWS Foundations Benchmark v1.2.0, PCI DSS v4.0.1	LOW	 No	Periodic
IAM.18	Ensure a support role has been created to manage incidents with AWS Support	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1	LOW	 No	Periodic
IAM.19	MFA should be enabled for all IAM users	NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v3.2.1, PCI DSS v4.0.1	MEDIUM	 No	Periodic
IAM.21	IAM customer managed policies that you create should not allow wildcard actions for services	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	LOW	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.22	IAM user credentials unused for 45 days should be removed	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-171 Rev. 2	MEDIUM	 No	Periodic
IAM.23	IAM Access Analyzer analyzers should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IAM.24	IAM roles should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IAM.25	IAM users should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IAM.26	Expired SSL/TLS certificates managed in IAM should be removed	CIS AWS Foundations Benchmark v3.0.0	MEDIUM	 No	Periodic
IAM.27	IAM identities should not have the AWSCloudShellFullAccess policy attached	CIS AWS Foundations Benchmark v3.0.0	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.28	IAM Access Analyzer external access analyzer should be enabled	CIS AWS Foundations Benchmark v3.0.0	HIGH	 No	Periodic
Inspector.1	Amazon Inspector EC2 scanning should be enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 No	Periodic
Inspector.2	Amazon Inspector ECR scanning should be enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 No	Periodic
Inspector.3	Amazon Inspector Lambda code scanning should be enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 No	Periodic
Inspector.4	Amazon Inspector Lambda standard scanning should be enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 No	Periodic
IoT.1	AWS IoT Device Defender security profiles should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IoT.2	AWS IoT Core mitigation actions should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoT.3	AWS IoT Core dimensions should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoT.4	AWS IoT Core authorizers should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoT.5	AWS IoT Core role aliases should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoT.6	AWS IoT Core policies should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTEvents.1	AWS IoT Events inputs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IoTEvents.2	AWS IoT Events detector models should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTEvents.3	AWS IoT Events alarm models should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTSiteWise.1	AWS IoT SiteWise asset models should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTSiteWise.2	AWS IoT SiteWise dashboards should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTSiteWise.3	AWS IoT SiteWise gateways should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTSiteWise.4	AWS IoT SiteWise portals should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IoTSiteWise.5	AWS IoT SiteWise projects should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTTwinMaker.1	AWS IoT TwinMaker sync jobs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTTwinMaker.2	AWS IoT TwinMaker workspaces should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTTwinMaker.3	AWS IoT TwinMaker scenes should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTTwinMaker.4	AWS IoT TwinMaker entities should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTWireless.1	AWS IoT Wireless multicast groups should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IoTWireless.2	AWS IoT Wireless service profiles should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IoTWireless.3	AWS IoT Wireless FUOTA tasks should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IVS.1	IVS playback key pairs should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IVS.2	IVS recording configurations should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
IVS.3	IVS channels should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Keyspaces.1	Amazon Keyspaces keyspaces should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Kinesis.1	Kinesis streams should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Kinesis.2	Kinesis streams should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Kinesis.3	Kinesis streams should have an adequate data retention period	AWS Foundational Security Best Practices	MEDIUM	 Yes	Change triggered
KMS.1	IAM customer managed policies should not allow decryption actions on all KMS keys	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
KMS.2	IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
KMS.3	AWS KMS keys should not be deleted unintentionally	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	CRITICAL	 No	Change triggered
KMS.4	AWS KMS key rotation should be enabled	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v1.2.0, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1	MEDIUM	 No	Periodic
KMS.5	KMS keys should not be publicly accessible	AWS Foundational Security Best Practices	CRITICAL	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Lambda.1	Lambda function policies should prohibit public access	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	CRITICAL	 No	Change triggered
Lambda.2	Lambda functions should use supported runtimes	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
Lambda.3	Lambda functions should be in a VPC	PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
Lambda.5	VPC Lambda functions should operate in multiple Availability Zones	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Lambda.6	Lambda functions should be tagged	AWS Resource Tagging Standard	LOW	Yes	Change triggered
Lambda.7	Lambda functions should have AWS X-Ray active tracing enabled	NIST SP 800-53 Rev. 5	LOW	No	Change triggered
Macie.1	Amazon Macie should be enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	No	Periodic
Macie.2	Macie automated sensitive data discovery should be enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	HIGH	No	Periodic
MSK.1	MSK clusters should be encrypted in transit among broker nodes	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	No	Change triggered
MSK.2	MSK clusters should have enhanced monitoring configured	NIST SP 800-53 Rev. 5	LOW	No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
MSK.3	MSK Connect connectors should be encrypted in transit	AWS Foundational Security Best Practices, PCI DSS v4.0.1	MEDIUM	No	Change triggered
MSK.4	MSK clusters should have public access disabled	AWS Foundational Security Best Practices	CRITICAL	No	Change triggered
MSK.5	MSK connectors should have logging enabled	AWS Foundational Security Best Practices	MEDIUM	No	Change triggered
MSK.6	MSK clusters should disable unauthenticated access	AWS Foundational Security Best Practices	MEDIUM	No	Change triggered
MQ.2	ActiveMQ brokers should stream audit logs to CloudWatch	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	No	Change triggered
MQ.3	Amazon MQ brokers should have automatic minor version upgrade enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	LOW	No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
MQ.4	Amazon MQ brokers should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
MQ.5	ActiveMQ brokers should use active/standby deployment mode	NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	LOW	 No	Change triggered
MQ.6	RabbitMQ brokers should use cluster deployment mode	NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	LOW	 No	Change triggered
Neptune.1	Neptune DB clusters should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
Neptune.2	Neptune DB clusters should publish audit logs to CloudWatch Logs	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Neptune.3	Neptune DB cluster snapshots should not be public	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	CRITICAL	 No	Change triggered
Neptune.4	Neptune DB clusters should have deletion protection enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	LOW	 No	Change triggered
Neptune.5	Neptune DB clusters should have automated backups enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 Yes	Change triggered
Neptune.6	Neptune DB cluster snapshots should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Neptune.7	Neptune DB clusters should have IAM database authentication enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
Neptune.8	Neptune DB clusters should be configured to copy tags to snapshots	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	LOW	 No	Change triggered
Neptune.9	Neptune DB clusters should be deployed across multiple Availability Zones	NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
NetworkFirewall.1	Network Firewall firewalls should be deployed across multiple Availability Zones	NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
NetworkFirewall.2	Network Firewall logging should be enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Periodic
NetworkFirewall.3	Network Firewall policies should have at least one rule group associated	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Change triggered
NetworkFirewall.4	The default stateless action for Network Firewall policies should be drop or forward for full packets	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
NetworkFirewall.5	The default stateless action for Network Firewall policies should be drop or forward for fragmented packets	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Change triggered
NetworkFirewall.6	Stateless network firewall rule group should not be empty	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Change triggered
NetworkFirewall.7	Network Firewall firewalls should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
NetworkFirewall.8	Network Firewall firewall policies should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
NetworkFirewall.9	Network Firewall firewalls should have deletion protection enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
NetworkFirewall.10	Network Firewall firewalls should have subnet change protection enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Opensearch.1	OpenSearch domains should have encryption at rest enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Opensearch.2	OpenSearch domains should not be publicly accessible	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	CRITICAL	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Opensearch.3	OpenSearch domains should encrypt data sent between nodes	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Opensearch.4	OpenSearch domain error logging to CloudWatch Logs should be enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Opensearch.5	OpenSearch domains should have audit logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
Opensearch.6	OpenSearch domains should have at least three data nodes	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Opensearch.7	OpenSearch domains should have fine-grained access control enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
Opensearch.8	Connections to OpenSearch domains should be encrypted using the latest TLS security policy	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Opensearch.9	OpenSearch domains should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Opensearch.10	OpenSearch domains should have the latest software update installed	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	LOW	 No	Change triggered
Opensearch.11	OpenSearch domains should have at least three dedicated primary nodes	NIST SP 800-53 Rev. 5	LOW	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
PCA.1	AWS Private CA root certificate authority should be disabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	LOW	 No	Periodic
PCA.2	AWS Private CA certificate authorities should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
RDS.1	RDS snapshot should be private	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	CRITICAL	 No	Change triggered
RDS.2	RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration	CIS AWS Foundations Benchmark v3.0.0, AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1	CRITICAL	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.3	RDS DB instances should have encryption at-rest enabled	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
RDS.4	RDS cluster snapshots and database snapshots should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
RDS.5	RDS DB instances should be configured with multiple Availability Zones	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.6	Enhanced monitoring should be configured for RDS DB instances	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	LOW	 Yes	Change triggered
RDS.7	RDS clusters should have deletion protection enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
RDS.8	RDS DB instances should have deletion protection enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
RDS.9	RDS DB instances should publish logs to CloudWatch Logs	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.10	IAM authentication should be configured for RDS instances	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
RDS.11	RDS instances should have automatic backups enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered
RDS.12	IAM authentication should be configured for RDS clusters	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
RDS.13	RDS automatic minor version upgrades should be enabled	CIS AWS Foundations Benchmark v3.0.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.14	Amazon Aurora clusters should have backtracking enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered
RDS.15	RDS DB clusters should be configured for multiple Availability Zones	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
RDS.16	Aurora DB clusters should be configured to copy tags to DB snapshots	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
RDS.17	RDS DB instances should be configured to copy tags to snapshots	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
RDS.18	RDS instances should be deployed in a VPC	Service-Managed Standard: AWS Control Tower	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.19	Existing RDS event notification subscriptions should be configured for critical cluster events	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
RDS.20	Existing RDS event notification subscriptions should be configured for critical database instance events	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	LOW	 No	Change triggered
RDS.21	An RDS event notifications subscription should be configured for critical database parameter group events	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	LOW	 No	Change triggered
RDS.22	An RDS event notifications subscription should be configured for critical database security group events	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	LOW	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.23	RDS instances should not use a database engine default port	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
RDS.24	RDS Database Clusters should use a custom administrator username	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
RDS.25	RDS database instances should use a custom administrator username	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
RDS.26	RDS DB instances should be protected by a backup plan	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.27	RDS DB clusters should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
RDS.28	RDS DB clusters should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
RDS.29	RDS DB cluster snapshots should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
RDS.30	RDS DB instances should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
RDS.31	RDS DB security groups should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
RDS.32	RDS DB snapshots should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.33	RDS DB subnet groups should be tagged	AWS Resource Tagging Standard	LOW	Yes	Change triggered
RDS.34	Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	No	Change triggered
RDS.35	RDS DB clusters should have automatic minor version upgrade enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	No	Change triggered
RDS.36	RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs	AWS Foundational Security Best Practices, PCI DSS v4.0.1	MEDIUM	Yes	Change triggered
RDS.37	Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs	AWS Foundational Security Best Practices, PCI DSS v4.0.1	MEDIUM	No	Change triggered
RDS.38	RDS for PostgreSQL DB instances should be encrypted in transit	AWS Foundational Security Best Practices	MEDIUM	No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.39	RDS for MySQL DB instances should be encrypted in transit	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
RDS.40	RDS for SQL Server DB instances should publish logs to CloudWatch Logs	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered
RDS.41	RDS for SQL Server DB instances should be encrypted in transit	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
RDS.42	RDS for MariaDB DB instances should publish logs to CloudWatch Logs	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
RDS.44	RDS for MariaDB DB instances should be encrypted in transit	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
RDS.45	Aurora MySQL DB clusters should have audit logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.46	RDS DB instances should not be deployed in public subnets with routes to internet gateways	AWS Foundational Security Best Practices	HIGH	 No	Periodic
Redshift.1	Amazon Redshift clusters should prohibit public access	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	CRITICAL	 No	Change triggered
Redshift.2	Connections to Amazon Redshift clusters should be encrypted in transit	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
Redshift.3	Amazon Redshift clusters should have automatic snapshots enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Redshift.4	Amazon Redshift clusters should have audit logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
Redshift.6	Amazon Redshift should have automatic upgrades to major versions enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Redshift.7	Redshift clusters should use enhanced VPC routing	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Redshift.8	Amazon Redshift clusters should not use the default Admin username	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Redshift.9	Redshift clusters should not use the default database name	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Redshift.10	Redshift clusters should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Redshift.11	Redshift clusters should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Redshift.12	Redshift event subscription notifications should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Redshift.13	Redshift cluster snapshots should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Redshift.14	Redshift cluster subnet groups should be tagged	AWS Resource Tagging Standard	LOW	Yes	Change triggered
Redshift.15	Redshift security groups should allow ingress on the cluster port only from restricted origins	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	No	Periodic
Redshift.16	Redshift cluster subnet groups should have subnets from multiple Availability Zones	NIST SP 800-53 Rev. 5	MEDIUM	No	Change triggered
Redshift.17	Redshift cluster parameter groups should be tagged	AWS Resource Tagging Standard	LOW	Yes	Change triggered
Redshift.18	Redshift clusters should have Multi-AZ deployments enabled	AWS Foundational Security Best Practices	MEDIUM	No	Change triggered
RedshiftServerless.1	Amazon Redshift Serverless workgroups should use enhanced VPC routing	AWS Foundational Security Best Practices	HIGH	No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RedshiftServerless.2	Connections to Redshift Serverless workgroups should be required to use SSL	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
RedshiftServerless.3	Redshift Serverless workgroups should prohibit public access	AWS Foundational Security Best Practices	HIGH	 No	Periodic
RedshiftServerless.4	Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
RedshiftServerless.5	Redshift Serverless namespaces should not use the default admin username	AWS Foundational Security Best Practices	MEDIUM	 Yes	Periodic
RedshiftServerless.6	Redshift Serverless namespaces should export logs to CloudWatch Logs	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
RedshiftServerless.7	Redshift Serverless namespaces should not use the default database name	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Route53.1	Route 53 health checks should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Route53.2	Route 53 public hosted zones should log DNS queries	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Change triggered
S3.1	S3 general purpose buckets should have block public access settings enabled	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Periodic
S3.2	S3 general purpose buckets should block public read access	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	CRITICAL	 No	Change triggered and periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
S3.3	S3 general purpose buckets should block public write access	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	CRITICAL	 No	Change triggered and periodic
S3.5	S3 general purpose buckets should require requests to use SSL	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
S3.6	S3 general purpose bucket policies should restrict access to other AWS accounts	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
S3.7	S3 general purpose buckets should use cross-Region replication	PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
S3.8	S3 general purpose buckets should block public access	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Change triggered
S3.9	S3 general purpose buckets should have server access logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
S3.10	S3 general purpose buckets with versioning enabled should have Lifecycle configurations	NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
S3.11	S3 general purpose buckets should have event notifications enabled	NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 Yes	Change triggered
S3.12	ACLs should not be used to manage user access to S3 general purpose buckets	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
S3.13	S3 general purpose buckets should have Lifecycle configurations	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	LOW	 Yes	Change triggered
S3.14	S3 general purpose buckets should have versioning enabled	NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	LOW	 No	Change triggered
S3.15	S3 general purpose buckets should have Object Lock enabled	NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
S3.17	S3 general purpose buckets should be encrypted at rest with AWS KMS keys	NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
S3.19	S3 access points should have block public access settings enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	CRITICAL	 No	Change triggered
S3.20	S3 general purpose buckets should have MFA delete enabled	CIS AWS Foundations Benchmark v3.0.0, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
S3.22	S3 general purpose buckets should log object-level write events	CIS AWS Foundations Benchmark v3.0.0, PCI DSS v4.0.1	MEDIUM	 No	Periodic
S3.23	S3 general purpose buckets should log object-level read events	CIS AWS Foundations Benchmark v3.0.0, PCI DSS v4.0.1	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
S3.24	S3 Multi-Region Access Points should have block public access settings enabled	AWS Foundational Security Best Practices, PCI DSS v4.0.1	HIGH	 No	Change triggered
S3.25	S3 directory buckets should have lifecycle configurations	AWS Foundational Security Best Practices	LOW	 Yes	Change triggered
SageMaker.1	Amazon SageMaker notebook instances should not have direct internet access	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Periodic
SageMaker.2	SageMaker notebook instances should be launched in a custom VPC	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
SageMaker.3	Users should not have root access to SageMaker notebook instances	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
SageMaker.4	SageMaker endpoint production variants should have an initial instance count greater than 1	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
SageMaker.5	SageMaker models should have network isolation enabled	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
SageMaker.6	SageMaker app image configurations should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
SageMaker.7	SageMaker images should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
SageMaker.8	SageMaker notebook instances should run on supported platforms	AWS Foundational Security Best Practices	MEDIUM	 No	Periodic
SecretsManager.1	Secrets Manager secrets should have automatic rotation enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 Yes	Change triggered
SecretsManager.2	Secrets Manager secrets configured with automatic rotation should rotate successfully	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
SecretsManager.3	Remove unused Secrets Manager secrets	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 Yes	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
SecretsManager.4	Secrets Manager secrets should be rotated within a specified number of days	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	MEDIUM	 Yes	Periodic
SecretsManager.5	Secrets Manager secrets should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
ServiceCatalog.1	Service Catalog portfolios should be shared within an AWS organization only	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	HIGH	 No	Periodic
SES.1	SES contact lists should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
SES.2	SES configuration sets should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
SNS.1	SNS topics should be encrypted at-rest using AWS KMS	NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
SNS.3	SNS topics should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
SNS.4	SNS topic access policies should not allow public access	AWS Foundational Security Best Practices	HIGH	 No	Change triggered
SQS.1	Amazon SQS queues should be encrypted at rest	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
SQS.2	SQS queues should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
SQS.3	SQS queue access policies should not allow public access	AWS Foundational Security Best Practices	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
SSM.1	EC2 instances should be managed by AWS Systems Manager	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
SSM.2	EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	HIGH	 No	Change triggered
SSM.3	EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, PCI DSS v3.2.1, PCI DSS v4.0.1, Service-Managed Standard: AWS Control Tower	LOW	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
SSM.4	SSM documents should not be public	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	CRITICAL	 No	Periodic
SSM.5	SSM documents should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
SSM.6	SSM Automation should have CloudWatch logging enabled	AWS Foundational Security Best Practices v1.0.0	MEDIUM	 No	Periodic
SSM.7	SSM documents should have the block public sharing setting enabled	AWS Foundational Security Best Practices v1.0.0	CRITICAL	 No	Periodic
StepFunctions.1	Step Functions state machines should have logging turned on	AWS Foundational Security Best Practices v1.0.0, PCI DSS v4.0.1	MEDIUM	 Yes	Change triggered
StepFunctions.2	Step Functions activities should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Transfer.1	Transfer Family workflows should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Transfer.2	Transfer Family servers should not use FTP protocol for endpoint connection	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Periodic
Transfer.3	Transfer Family connectors should have logging enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Transfer.4	Transfer Family agreements should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Transfer.5	Transfer Family certificates should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
Transfer.6	Transfer Family connectors should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Transfer.7	Transfer Family profiles should be tagged	AWS Resource Tagging Standard	LOW	 Yes	Change triggered
WAF.1	AWS WAF Classic Global Web ACL logging should be enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	MEDIUM	 No	Periodic
WAF.2	AWS WAF Classic Regional rules should have at least one condition	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
WAF.3	AWS WAF Classic Regional rule groups should have at least one rule	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
WAF.4	AWS WAF Classic Regional web ACLs should have at least one rule or rule group	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
WAF.6	AWS WAF Classic global rules should have at least one condition	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
WAF.7	AWS WAF Classic global rule groups should have at least one rule	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
WAF.8	AWS WAF Classic global web ACLs should have at least one rule or rule group	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
WAF.10	AWS WAF web ACLs should have at least one rule or rule group	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
WAF.11	AWS WAF web ACL logging should be enabled	NIST SP 800-53 Rev. 5, PCI DSS v4.0.1	LOW	 No	Periodic
WAF.12	AWS WAF rules should have CloudWatch metrics enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, NIST SP 800-171 Rev. 2	MEDIUM	 No	Change triggered
WorkSpaces.1	WorkSpaces user volumes should be encrypted at rest	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered
WorkSpaces.2	WorkSpaces root volumes should be encrypted at rest	AWS Foundational Security Best Practices	MEDIUM	 No	Change triggered

Change log for Security Hub CSPM controls

The following change log tracks material changes to existing AWS Security Hub CSPM controls, which can result in changes to the overall status of a control and the compliance status of its findings. For information about how Security Hub CSPM evaluates control status, see Evaluating compliance status and control status. Changes can take a few days after their entry in this log to affect all AWS Regions in which the control is available.

This log tracks changes occurring since April 2023. Choose a control to review additional details about it. Title changes are noted in a control's detailed description for 90 days.

Date of change	Control ID and title	Description of change
August 13, 2025	[SageMaker.5] SageMaker models should have network isolation enabled	Security Hub CSPM changed the title and description of this control. The new title and description more accurately reflect that the control checks the setting for the `EnableNetworkIsolation` parameter of Amazon SageMaker AI hosted models. Previously, the title of this control was: *SageMaker models should block inbound traffic*.
August 13, 2025	[EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch	Security Hub CSPM changed the title and description of this control. The new title and description more precisely reflect the scope and nature of the check that the control performs. Previously, the title of this control was: *EFS mount targets should not be associated with a public subnet*.

Date of change	Control ID and title	Description of change
July 24, 2025	[EKS.2] EKS clusters should run on a supported Kubernetes version	This control checks whether an Amazon EKS cluster runs on a supported Kubernetes version. Security Hub CSPM changed the parameter value for this control from `1.30` to `1.31`. Standard support for Kubernetes version 1.30 in Amazon EKS ended on July 23, 2025.
July 23, 2025	[EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes	Security Hub CSPM changed the title of this control. The new title more accurately reflects that the control only checks Amazon EC2 Spot Fleet requests that specify launch parameters. Previously, the title of this control was: *EC2 Spot Fleet requests should enable encryption for attached EBS volumes*.

Date of change	Control ID and title	Description of change
June 30, 2025	[IAM.13] Ensure IAM password policy requires at least one symbol	Security Hub CSPM removed this control from the PCI DSS v4.0.1 standard. PCI DSS v4.0.1 doesn't explicitly require the use of symbols in passwords.
June 30, 2025	[IAM.17] Ensure IAM password policy expires passwords within 90 days or less	Security Hub CSPM removed this control from the NIST SP 800-171 Revision 2 standard. NIST SP 800-171 Revision 2 doesn't explicitl y require password expiration periods of 90 days or less.
June 30, 2025	[RDS.16] Aurora DB clusters should be configured to copy tags to DB snapshots	Security Hub CSPM changed the title of this control. The new title more accuratel y reflects that the control only checks Amazon Aurora DB clusters. Previousl y, the title of this control was: *RDS DB clusters should be configured to copy tags to snapshots*.

Date of change	Control ID and title	Description of change
June 30, 2025	[SageMaker.8] SageMaker notebook instances should run on supported platforms	This control checks whether an Amazon SageMaker AI notebook instance is configured to run on a supported platform, based on the platform identifier specified for the notebook instance. Security Hub CSPM no longer supports `notebook-al2-v1` and `notebook-al2-v2` as parameter values for this control. Notebook instances that run on these platforms reached end of support on June 30, 2025.

Date of change	Control ID and title	Description of change
May 30, 2025	[IAM.10] Password policies for IAM users should have strong configurations	Security Hub CSPM removed this control from the PCI DSS v4.0.1 standard. This control checks whether account password policies for IAM users meet minimum requirements, including a minimum password length of 7 characters. PCI DSS v4.0.1 now requires passwords to have a minimum of 8 characters. The control continues to apply to the PCI DSS v3.2.1 standard, which has different password requirements. To evaluate account password policies against PCI DSS v4.0.1 requirements, you can use the IAM.7 control. This control requires passwords to have a minimum of 8 characters. It also supports custom values for password

Date of change	Control ID and title	Description of change
		length and other parameters. The IAM.7 control is part of the PCI DSS v4.0.1 standard in Security Hub CSPM.
May 8, 2025	[RDS.46] RDS DB instances should not be deployed in public subnets with routes to internet gateways	Security Hub CSPM rolled back the release of the RDS.46 control in all AWS Regions. Previously, this control supported the AWS Foundational Security Best Practices (FSBP) standard.

Date of change	Control ID and title	Description of change
April 7, 2025	[ELB.17] Application and Network Load Balancers with listeners should use recommended security policies	This control checks whether the HTTPS listener for an Application Load Balancer or the TLS listener for a Network Load Balancer is configured to encrypt data in transit by using a recommend ed security policy. Security Hub CSPM now supports two additional parameter values for this control: `ELBSecuri tyPolicy-TLS13-1-2-Res-2021-06` and `ELBSecuri tyPolicy-TLS13-1-2-Res-FIPS-2023-04` .

Date of change	Control ID and title	Description of change
March 27, 2025	[Lambda.2] Lambda functions should use supported runtimes	This control checks whether the runtime settings for an AWS Lambda function match expected values for supported runtimes in each language. Security Hub CSPM now supports `ruby3.4` as a parameter value for this control. AWS Lambda added support for this runtime.
March 26, 2025	[EKS.2] EKS clusters should run on a supported Kubernetes version	This control checks whether an Amazon Elastic Kubernetes Service (Amazon EKS) cluster runs on a supported Kubernetes version. For the `oldestVersionSupported` parameter, Security Hub CSPM changed the value from `1.29` to `1.30`. The oldest supported Kubernetes version is now `1.30`.

Date of change	Control ID and title	Description of change
March 10, 2025	[Lambda.2] Lambda functions should use supported runtimes	This control checks whether the runtime settings for an AWS Lambda function match expected values for supported runtimes in each language. Security Hub CSPM no longer supports `dotnet6` and `python3.8` as parameter values for this control. AWS Lambda no longer supports these runtimes.

Date of change	Control ID and title	Description of change
March 7, 2025	[RDS.18] RDS instances should be deployed in a VPC	Security Hub CSPM removed this control from the AWS Foundational Security Best Practices standard and automated checks for NIST SP 800-53 Rev. 5 requirements. Since Amazon EC2-Classic networking was retired, Amazon Relational Database Service (Amazon RDS) instances can no longer be deployed outside a VPC. The control continues to be part of the AWS Control Tower service-managed standard.
January 10, 2025	[Glue.2] AWS Glue jobs should have logging enabled	Security Hub CSPM retired this control and removed it from all standards.
December 20, 2024	EC2.61 through EC2.169	Security Hub CSPM rolled back the release of the EC2.61 through EC2.169 controls.

Date of change	Control ID and title	Description of change
December 12, 2024	[RDS.23] RDS instances should not use a database engine default port	RDS.23 checks whether an Amazon Relational Database Service (Amazon RDS) cluster or instance uses a port other than the default port of the database engine. We updated the control so that the underlying AWS Config rule returns a result of `NOT_APPLICABLE` for RDS instances that are part of a cluster.
December 2, 2024	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM now supports `nodejs22.x` as a parameter.

Date of change	Control ID and title	Description of change
November 26, 2024	[EKS.2] EKS clusters should run on a supported Kubernetes version	This control checks whether an Amazon Elastic Kubernetes Service (Amazon EKS) cluster runs on a supported Kubernetes version. The oldest supported version is now 1.29.

Date of change	Control ID and title	Description of change
November 20, 2024	[Config.1] AWS Config should be enabled and use the service-linked role for resource recording	Config.1 checks whether AWS Config is enabled, uses the service-linked role, and records resources for enabled controls. Security Hub CSPM increased the severity of this control from `MEDIUM` to `CRITICAL`. Security Hub CSPM also added new status codes and status reasons for failed Config.1 findings. These changes reflect the importance of Config.1 to the operation of Security Hub CSPM controls. If you have AWS Config or resource recording disabled, you can receive inaccurate control findings. To receive a `PASSED` finding for Config.1, turn on resource recording for resources that correspond to enabled Security

Date of change	Control ID and title	Description of change
		Hub CSPM controls, and disable controls that aren't required in your organizat ion. For instructi ons on configuri ng AWS Config for Security Hub CSPM, see Enabling and configuring AWS Config for Security Hub CSPM. For a list of Security Hub CSPM controls and their correspon ding resources, see Required AWS Config resources for control findings.
November 12, 2024	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM now supports `python3.13` as a parameter.

Date of change	Control ID and title	Description of change
October 11, 2024	ElastiCache controls	Changed control titles for ElastiCache.3, ElastiCache.4, ElastiCache.5, and ElastiCache.7. Titles no longer mention Redis OSS because the controls also apply to ElastiCache for Valkey.
September 27, 2024	[ELB.4] Application Load Balancer should be configured to drop invalid http headers	Changed control title from **Application Load Balancer should be configured to drop http headers** to **Application Load Balancer should be configured to drop invalid http headers**.
August 19, 2024	Title changes to DMS.12 and ElastiCache controls	Changed control titles for DMS.12 and ElastiCache.1 through ElastiCache.7. We changed these titles to reflect a name change in the Amazon ElastiCache (Redis OSS) service.

Date of change	Control ID and title	Description of change
August 15, 2024	[Config.1] AWS Config should be enabled and use the service-linked role for resource recording	Config.1 checks whether AWS Config is enabled, uses the service-linked role, and records resources for enabled controls. Security Hub CSPM added a custom control parameter named `includeConfigServiceLinkedRoleCheck`. By setting this parameter to `false`, you can opt out of checking whether AWS Config uses the service-linked role.
July 31, 2024	[IoT.1] AWS IoT Device Defender security profiles should be tagged	Changed control title from **AWS IoT Core security profiles should be tagged** to **AWS IoT Device Defender security profiles should be tagged**.

Date of change	Control ID and title	Description of change
July 29, 2024	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM no longer supports `nodejs16.x` as a parameter.
July 29, 2024	[EKS.2] EKS clusters should run on a supported Kubernetes version	This control checks whether an Amazon Elastic Kubernetes Service (Amazon EKS) cluster runs on a supported Kubernetes version. The oldest supported version is `1.28`.
June 25, 2024	[Config.1] AWS Config should be enabled and use the service-linked role for resource recording	This control checks whether AWS Config is enabled, uses the service-linked role, and records resources for enabled controls. Security Hub CSPM updated the control title to reflect what the control evaluates.

Date of change	Control ID and title	Description of change
June 14, 2024	[RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs	This control checks whether an Amazon Aurora MySQL DB cluster is configure d to publish audit logs to Amazon CloudWatch Logs. Security Hub CSPM updated the control so that it doesn't generate findings for Aurora Serverless v1 DB clusters.
June 11, 2024	[EKS.2] EKS clusters should run on a supported Kubernetes version	This control checks whether an Amazon Elastic Kubernete s Service (Amazon EKS) cluster runs on a supported Kubernete s version. The oldest supported version is `1.27`.

Date of change	Control ID and title	Description of change
June 10, 2024	[Config.1] AWS Config should be enabled and use the service-linked role for resource recording	This control checks whether AWS Config is enabled and AWS Config resource recording is turned on. Previously, the control produced a PASSED finding only if you configure d recording for all resources. Security Hub CSPM updated the control to produce a PASSED finding when recording is turned on for resources that are required for enabled controls. The control has also been updated to check whether the AWS Config service-linked role is used, which provides permissions to record necessary resources.

Date of change	Control ID and title	Description of change
May 8, 2024	[S3.20] S3 general purpose buckets should have MFA delete enabled	This control checks whether an Amazon S3 general purpose versioned bucket has multi-factor authentication (MFA) delete enabled. Previously, the control produced a `FAILED` finding for buckets that have a Lifecycle configura tion. However, MFA delete with versionin g can't be enabled on a bucket that has a Lifecycle configura tion. Security Hub CSPM updated the control to produce no findings for buckets that have a Lifecycle configuration. The control description has been updated to reflect the current behavior.

Date of change	Control ID and title	Description of change
May 2, 2024	[EKS.2] EKS clusters should run on a supported Kubernetes version	Security Hub CSPM updated the oldest supported version of Kubernetes that the Amazon EKS cluster can run on to produce a passed finding. The current oldest supported version is Kubernetes 1.26.
April 30, 2024	[CloudTrail.3] At least one CloudTrail trail should be enabled	Changed control title from **CloudTrail should be enabled** to **At least one CloudTrail trail should be enabled**. This control currently produces a PASSED finding if an AWS account has at least one CloudTrail trail enabled. The title and description have been changed to accurately reflect the current behavior.

Date of change	Control ID and title	Description of change
April 29, 2024	[AutoScaling.1] Auto Scaling groups associated with a load balancer should use ELB health checks	Changed control title from **Auto Scaling groups associated with a Classic Load Balancer should use load balancer health checks** to **Auto Scaling groups associated with a load balancer should use ELB health checks**. This control currently evaluates Application, Gateway, Network, and Classic Load Balancers. The title and description have been changed to accurately reflect the current behavior.

Date of change	Control ID and title	Description of change
April 19, 2024	[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events	The control checks whether AWS CloudTrail is enabled and configured with at least one multi-Region trail that includes read and write management events. Previously, the control incorrectly generated PASSED findings when an account had CloudTrail enabled and configured with at least one multi-Region trail, even if no trail captured read and write management events. The control now generates a PASSED finding only when CloudTrail is enabled and configured with at least one multi-Region trail that captures read and write management events.

Date of change	Control ID and title	Description of change
April 10, 2024	[Athena.1] Athena workgroups should be encrypted at rest	Security Hub CSPM retired this control and removed it from all standards. Athena workgroups send logs to Amazon Simple Storage Service (Amazon S3) buckets. Amazon S3 now provides default encryption with S3 managed keys (SS3-S3) on new and existing S3 buckets.
April 10, 2024	[AutoScaling.4] Auto Scaling group launch configuration should not have a metadata response hop limit greater than 1	Security Hub CSPM retired this control and removed it from all standards. Metadata response hop limits for Amazon Elastic Compute Cloud (Amazon EC2) instances are workload dependent.

Date of change	Control ID and title	Description of change
April 10, 2024	[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)	Security Hub CSPM retired this control and removed it from all standards . Integrating AWS CloudFormation stacks with Amazon SNS topics is no longer a security best practice. Though integrating important CloudFormation stacks with SNS topics can be useful, it is not required for all stacks.
April 10, 2024	[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled	Security Hub CSPM retired this control and removed it from all standards. Enabling privileged mode in a CodeBuild project does not impose an additiona l risk to the customer environment.

Date of change	Control ID and title	Description of change
April 10, 2024	[IAM.20] Avoid the use of the root user	Security Hub CSPM retired this control and removed it from all standards. The purpose of this control is covered by another control, [CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user.
April 10, 2024	[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic	Security Hub CSPM retired this control and removed it from all standards . Logging delivery status for SNS topics is no longer a security best practice. Though logging delivery status for important SNS topics can be useful, it is not required for all topics.

Date of change	Control ID and title	Description of change
April 10, 2024	[S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations	Security Hub CSPM removed this control from AWS Foundational Security Best Practices and Service-Managed Standard: AWS Control Tower. The purpose of this control is covered by two other controls: [S3.13] S3 general purpose buckets should have Lifecycle configurations and [S3.14] S3 general purpose buckets should have versioning enabled. This control is still part of NIST SP 800-53 Rev. 5.

Date of change	Control ID and title	Description of change
April 10, 2024	[S3.11] S3 general purpose buckets should have event notifications enabled	Security Hub CSPM removed this control from AWS Foundational Security Best Practices and Service-Managed Standard: AWS Control Tower. Though there are some cases where event notifications for S3 buckets are useful, this not a universal security best practice. This control is still part of NIST SP 800-53 Rev. 5.

Date of change	Control ID and title	Description of change
April 10, 2024	[SNS.1] SNS topics should be encrypted at-rest using AWS KMS	Security Hub CSPM removed this control from AWS Foundational Security Best Practices and Service-Managed Standard: AWS Control Tower. By default, SNS encrypts topics at rest with disk encryption. For more information, see Data encryption. Using AWS KMS to encrypt topics is no longer recommended as a security best practice. This control is still part of NIST SP 800-53 Rev. 5.

Date of change	Control ID and title	Description of change
April 8, 2024	[ELB.6] Application, Gateway, and Network Load Balancers should have deletion protection enabled	Changed control title from **Application Load Balancer deletion protection should be enabled** to **Application, Gateway, and Network Load Balancers should have deletion protection enabled**. This control currently evaluates Application, Gateway, and Network Load Balancers. The title and description have been changed to accurately reflect the current behavior.

Date of change	Control ID and title	Description of change
March 22, 2024	[Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy	Changed control title from **Connections to OpenSearch domains should be encrypted using TLS 1.2** to **Connections to OpenSearch domains should be encrypted using the latest TLS security policy**. Previously, the control only checked whether connections to OpenSearch domains used TLS 1.2. The control now produces a PASSED finding if OpenSearch domains are encrypted using the latest TLS security policy. The control title and description have been updated to reflect the current behavior.

Date of change	Control ID and title	Description of change
March 22, 2024	[ES.8] Connections to Elasticsearch domains should be encrypted using the latest TLS security policy	Changed control title from **Connections to Elasticsearch domains should be encrypted using TLS 1.2** to **Connections to Elasticsearch domains should be encrypted using the latest TLS security policy**. Previously, the control only checked whether connections to Elasticsearch domains used TLS 1.2. The control now produces a PASSED finding if Elasticsearch domains are encrypted using the latest TLS security policy. The control title and description have been updated to reflect the current behavior.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.1] S3 general purpose buckets should have block public access settings enabled	Changed title from **S3 Block Public Access setting should be enabled** to **S3 general purpose buckets should have block public access settings enabled**. Security Hub CSPM changed the title to account for a new S3 bucket type.
March 12, 2024	[S3.2] S3 general purpose buckets should block public read access	Changed title from **S3 buckets should prohibit public read access** to **S3 general purpose buckets should block public read access**. Security Hub CSPM changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.3] S3 general purpose buckets should block public write access	Changed title from **S3 buckets should prohibit public write access** to **S3 general purpose buckets should block public write access**. Security Hub CSPM changed the title to account for a new S3 bucket type.
March 12, 2024	[S3.5] S3 general purpose buckets should require requests to use SSL	Changed title from **S3 buckets should require requests to use Secure Socket Layer** to **S3 general purpose buckets should require requests to use SSL**. Security Hub CSPM changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts	Changed title from **S3 permissions granted to other AWS accounts in bucket policies should be restricte d** to **S3 general purpose bucket policies should restrict access to other AWS accounts**. Security Hub CSPM changed the title to account for a new S3 bucket type.
March 12, 2024	[S3.7] S3 general purpose buckets should use cross-Region replication	Changed title from **S3 buckets should have cross-Region replication enabled** to **S3 general purpose buckets should use cross-Region replication**. Security Hub CSPM changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.7] S3 general purpose buckets should use cross-Region replication	Changed title from **S3 buckets should have cross-Region replication enabled** to **S3 general purpose buckets should use cross-Region replication**. Security Hub CSPM changed the title to account for a new S3 bucket type.
March 12, 2024	[S3.8] S3 general purpose buckets should block public access	Changed title from **S3 Block Public Access setting should be enabled at the bucket-le vel** to **S3 general purpose buckets should block public access**. Security Hub CSPM changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.9] S3 general purpose buckets should have server access logging enabled	Changed title from **S3 bucket server access logging should be enabled** to **Server access logging should be enabled for S3 general purpose buckets**. Security Hub CSPM changed the title to account for a new S3 bucket type.
March 12, 2024	[S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations	Changed title from **S3 buckets with versioning enabled should have lifecycle policies configure d** to **S3 general purpose buckets with versioning enabled should have Lifecycle configura tions**. Security Hub CSPM changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.11] S3 general purpose buckets should have event notifications enabled	Changed title from **S3 buckets should have event notifications enabled** to **S3 general purpose buckets should have event notifications enabled**. Security Hub CSPM changed the title to account for a new S3 bucket type.
March 12, 2024	[S3.12] ACLs should not be used to manage user access to S3 general purpose buckets	Changed title from **S3 access control lists (ACLs) should not be used to manage user access to buckets** to **ACLs should not be used to manage user access to S3 general purpose buckets**. Security Hub CSPM changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.13] S3 general purpose buckets should have Lifecycle configurations	Changed title from **S3 buckets should have lifecycle policies configured** to **S3 general purpose buckets should have Lifecycle configurations**. Security Hub CSPM changed the title to account for a new S3 bucket type.
March 12, 2024	[S3.14] S3 general purpose buckets should have versioning enabled	Changed title from **S3 buckets should use versioning** to **S3 general purpose buckets should have versioning enabled**. Security Hub CSPM changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.15] S3 general purpose buckets should have Object Lock enabled	Changed title from **S3 buckets should be configured to use Object Lock** to **S3 general purpose buckets should have Object Lock enabled**. Security Hub CSPM changed the title to account for a new S3 bucket type.
March 12, 2024	[S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys	Changed title from **S3 buckets should be encrypted at rest with AWS KMS keys** to **S3 general purpose buckets should be encrypted at rest with AWS KMS keys**. Security Hub CSPM changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 7, 2024	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM now supports `nodejs20.x` and `ruby3.3` as parameters.
February 22, 2024	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM now supports `dotnet8` as a parameter.

Date of change	Control ID and title	Description of change
February 5, 2024	[EKS.2] EKS clusters should run on a supported Kubernetes version	Security Hub CSPM updated the oldest supported version of Kubernetes that the Amazon EKS cluster can run on to produce a passed finding. The current oldest supported version is Kubernetes 1.25.

Date of change	Control ID and title	Description of change
January 10, 2024	[CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials	Changed title from **CodeBuild GitHub or Bitbucket source repository URLs should use OAuth** to **CodeBuild Bitbucket source repository URLs should not contain sensitive credentials**. Security Hub CSPM removed mention of OAuth because other connection methods can also be secure. Security Hub CSPM removed mention of GitHub because it's no longer possible to have a personal access token or username and password in GitHub source repository URLs.

Date of change	Control ID and title	Description of change
January 8, 2024	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM no longer supports `go1.x` and `java8` as parameters because these are retired runtimes.
December 29, 2023	[RDS.8] RDS DB instances should have deletion protection enabled	RDS.8 checks whether an Amazon RDS DB instance that uses one of the supported database engines has deletion protection enabled. Security Hub CSPM now supports `custom-oracle-ee`, `oracle-ee-cdb`, and `oracle-se2-cdb` as database engines.

Date of change	Control ID and title	Description of change
December 22, 2023	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM now supports `java21` and `python3.12` as parameters. Security Hub CSPM no longer supports `ruby2.7` as a parameter.
December 15, 2023	[CloudFront.1] CloudFront distributions should have a default root object configured	CloudFront.1 checks whether an Amazon CloudFront distribut ion has a default root object configured. Security Hub CSPM lowered the severity of this control from CRITICAL to HIGH because adding the default root object is a recommendation that depends on a user's application and specific requirements.

Date of change	Control ID and title	Description of change
December 5, 2023	[EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22	Changed control title from **Security groups should not allow ingress from 0.0.0.0/0 to port 22** to **Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22**.
December 5, 2023	[EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389	Changed control title from **Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389** to **Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389**.

Date of change	Control ID and title	Description of change
December 5, 2023	[RDS.9] RDS DB instances should publish logs to CloudWatch Logs	Changed control title from **Database logging should be enabled** to **RDS DB instances should publish logs to CloudWatch Logs**. Security Hub CSPM identified that this control only checks whether logs are published to Amazon CloudWatch Logs and doesn't check whether RDS logs are enabled. The control produces a PASSED finding if RDS DB instances are configured to publish logs to CloudWatch Logs. The control title has been updated to reflect the current behavior.

Date of change	Control ID and title	Description of change
December 5, 2023	[EKS.8] EKS clusters should have audit logging enabled	This control checks whether Amazon EKS clusters have audit logging enabled. The AWS Config rule that Security Hub CSPM uses to evaluate this control changed from `eks-cluster-logging-enabled` to `eks-cluster-log-enabled` .
November 17, 2023	[EC2.19] Security groups should not allow unrestricted access to ports with high risk	EC2.19 checks whether unrestricted incoming traffic for a security group is accessible to the specified ports that are considered to be high risk. Security Hub CSPM updated this control to account for managed prefix lists when they are supplied as the source for a security group rule. The control produces a `FAILED` finding if the prefix lists contain the strings '0.0.0.0/0' or '::/0'.

Date of change	Control ID and title	Description of change
November 16, 2023	[CloudWatch.15] CloudWatch alarms should have specified actions configured	Changed control title from **CloudWatch alarms should have an action configured for the ALARM state** to **CloudWatch alarms should have specified actions configured**.
November 16, 2023	[CloudWatch.16] CloudWatch log groups should be retained for a specified time period	Changed control title from **CloudWatch log groups should be retained for at least 1 year** to **CloudWatch log groups should be retained for a specified time period**.
November 16, 2023	[Lambda.5] VPC Lambda functions should operate in multiple Availability Zones	Changed control title from **VPC Lambda functions should operate in more than one Availability Zone** to **VPC Lambda functions should operate in multiple Availability Zones**.

Date of change	Control ID and title	Description of change
November 16, 2023	[AppSync.2] AWS AppSync should have field-level logging enabled	Changed control title from **AWS AppSync should have request-level and field-level logging turned on** to **AWS AppSync should have field-level logging enabled**.
November 16, 2023	[EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses	Changed control title from **Amazon Elastic MapReduce cluster master nodes should not have public IP addresses** to **Amazon EMR cluster primary nodes should not have public IP addresses**.
November 16, 2023	[Opensearch.2] OpenSearch domains should not be publicly accessible	Changed control title from **OpenSearch domains should be in a VPC** to **OpenSearch domains should not be publicly accessible**.

Date of change	Control ID and title	Description of change
November 16, 2023	[ES.2] Elasticsearch domains should not be publicly accessible	Changed control title from **Elasticsearch domains should be in a VPC** to **Elasticsearch domains should not be publicly accessible**.

Date of change	Control ID and title	Description of change
October 31, 2023	[ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled	ES.4 checks whether Elasticsearch domains are configured to send error logs to Amazon CloudWatch Logs. The control previously produced a PASSED finding for an Elasticsearch domain that has any logs configured to send to CloudWatch Logs. Security Hub CSPM updated the control to produce a PASSED finding only for an Elasticsearch domain that is configured to send error logs to CloudWatch Logs. The control was also updated to exclude Elasticsearch versions that don't support error logs from evaluation.

Date of change	Control ID and title	Description of change
October 16, 2023	[EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22	EC2.13 checks whether security groups allow unrestricted ingress access to port 22. Security Hub CSPM updated this control to account for managed prefix lists when they are supplied as the source for a security group rule. The control produces a `FAILED` finding if the prefix lists contain the strings '0.0.0.0/0' or '::/0'.

Date of change	Control ID and title	Description of change
October 16, 2023	[EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389	EC2.14 checks whether security groups allow unrestricted ingress access to port 3389. Security Hub CSPM updated this control to account for managed prefix lists when they are supplied as the source for a security group rule. The control produces a `FAILED` finding if the prefix lists contain the strings '0.0.0.0/0' or '::/0'.

Date of change	Control ID and title	Description of change
October 16, 2023	[EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports	EC2.18 checks whether the security groups that are in use allow unrestric ted incoming traffic. Security Hub CSPM updated this control to account for managed prefix lists when they are supplied as the source for a security group rule. The control produces a `FAILED` finding if the prefix lists contain the strings '0.0.0.0/0' or '::/0'.
October 16, 2023	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM now supports `python3.11` as a parameter.

Date of change	Control ID and title	Description of change
October 4, 2023	[S3.7] S3 general purpose buckets should use cross-Region replication	Security Hub CSPM added the parameter `ReplicationType` with a value of `CROSS-REGION` to ensure that S3 buckets have cross-Region replication enabled rather than same-Region replication.
September 27, 2023	[EKS.2] EKS clusters should run on a supported Kubernetes version	Security Hub CSPM updated the oldest supported version of Kubernetes that the Amazon EKS cluster can run on to produce a passed finding. The current oldest supported version is Kubernetes `1.24`.

Date of change	Control ID and title	Description of change
September 20, 2023	[CloudFront.2] CloudFront distributions should have origin access identity enabled	Security Hub CSPM retired this control and removed it from all standards. Instead, see [CloudFront.13] CloudFront distributions should use origin access control. Origin access control is the current security best practice. This control will be removed from documentation in 90 days.

Date of change	Control ID and title	Description of change
September 20, 2023	[EC2.22] Unused Amazon EC2 security groups should be removed	Security Hub CSPM removed this control from AWS Foundational Security Best Practices (FSBP) and National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5. It is still part of Service-Managed Standard: AWS Control Tower. This control produces a passed finding if security groups are attached to EC2 instances or to an elastic network interface. However, for certain use cases, unattached security groups don't pose a security risk. You can use other EC2 controls—such as EC2.2, EC2.13, EC2.14, EC2.18, and EC2.19—to monitor your security groups.

Date of change	Control ID and title	Description of change
September 20, 2023	[EC2.29] EC2 instances should be launched in a VPC	Security Hub CSPM retired this control and removed it from all standards . Amazon EC2 has migrated EC2-Class ic instances to a VPC. This control will be removed from documentation in 90 days.
September 20, 2023	[S3.4] S3 buckets should have server-side encryption enabled	Security Hub CSPM retired this control and removed it from all standards . Amazon S3 now provides default encryption with S3 managed keys (SS3-S3) on new and existing S3 buckets. The encryptio n settings are unchanged for existing buckets that are encrypted with SS3-S3 or SS3-KMS server-side encryptio n. This control will be removed from documentation in 90 days.

Date of change	Control ID and title	Description of change
September 14, 2023	[EC2.2] VPC default security groups should not allow inbound or outbound traffic	Changed control title from **The VPC default security group should not allow inbound and outbound traffic** to **VPC default security groups should not allow inbound or outbound traffic**.
September 14, 2023	[IAM.9] MFA should be enabled for the root user	Changed control title from **Virtual MFA should be enabled for the root user** to **MFA should be enabled for the root user**.
September 14, 2023	[RDS.19] Existing RDS event notification subscriptions should be configured for critical cluster events	Changed control title from **An RDS event notificat ions subscription should be configure d for critical cluster events** to **Existing RDS event notificat ion subscriptions should be configure d for critical cluster events**.

Date of change	Control ID and title	Description of change
September 14, 2023	[RDS.20] Existing RDS event notification subscriptions should be configured for critical database instance events	Changed control title from **An RDS event notifications subscription should be configured for critical database instance events** to **Existing RDS event notification subscriptions should be configured for critical database instance events**.
September 14, 2023	[WAF.2] AWS WAF Classic Regional rules should have at least one condition	Changed control title from **A WAF Regional rule should have at least one condition** to **AWS WAF Classic Regional rules should have at least one condition**.
September 14, 2023	[WAF.3] AWS WAF Classic Regional rule groups should have at least one rule	Changed control title from **A WAF Regional rule group should have at least one rule** to **AWS WAF Classic Regional rule groups should have at least one rule**.

Date of change	Control ID and title	Description of change
September 14, 2023	[WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group	Changed control title from **A WAF Regional web ACL should have at least one rule or rule group** to **AWS WAF Classic Regional web ACLs should have at least one rule or rule group**.
September 14, 2023	[WAF.6] AWS WAF Classic global rules should have at least one condition	Changed control title from **A WAF global rule should have at least one condition** to **AWS WAF Classic global rules should have at least one condition**.
September 14, 2023	[WAF.7] AWS WAF Classic global rule groups should have at least one rule	Changed control title from **A WAF global rule group should have at least one rule** to **AWS WAF Classic global rule groups should have at least one rule**.

Date of change	Control ID and title	Description of change
September 14, 2023	[WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group	Changed control title from **A WAF global web ACL should have at least one rule or rule group** to **AWS WAF Classic global web ACLs should have at least one rule or rule group**.
September 14, 2023	[WAF.10] AWS WAF web ACLs should have at least one rule or rule group	Changed control title from **A WAFv2 web ACL should have at least one rule or rule group** to **AWS WAF web ACLs should have at least one rule or rule group**.
September 14, 2023	[WAF.11] AWS WAF web ACL logging should be enabled	Changed control title from **AWS WAFv2 web ACL logging should be activated** to **AWS WAF web ACL logging should be enabled**.

Date of change	Control ID and title	Description of change
July 20, 2023	[S3.4] S3 buckets should have server-side encryption enabled	S3.4 checks whether an Amazon S3 bucket either has server-side encryption enabled or that the S3 bucket policy explicitly denies `PutObject` requests without server-side encryption. Security Hub CSPM updated this control to include dual-layer server side encryption with KMS keys (DSSE-KMS). The control produces a passed finding when an S3 bucket is encrypted with SSE-S3, SSE-KMS, or DSSE-KMS.

Date of change	Control ID and title	Description of change
July 17, 2023	[S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys	S3.17 checks whether an Amazon S3 bucket is encrypted with an AWS KMS key. Security Hub CSPM updated this control to include dual-layer server side encryptio n with KMS keys (DSSE-KMS). The control produces a passed finding when an S3 bucket is encrypted with SSE-KMS or DSSE-KMS.
June 9, 2023	[EKS.2] EKS clusters should run on a supported Kubernetes version	EKS.2 checks whether an Amazon EKS cluster is running on a supported Kubernetes version.T he oldest supported version is now 1 . 23.

Date of change	Control ID and title	Description of change
June 9, 2023	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM now supports `ruby3.2` as a parameter.
June 5, 2023	[APIGateway.5] API Gateway REST API cache data should be encrypted at rest	APIGateway.5.checks whether all methods in Amazon API Gateway REST API stages are encrypted at rest. Security Hub CSPM updated the control to evaluate the encryption of a particular method only when caching is enabled for that method.

Date of change	Control ID and title	Description of change
May 18, 2023	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM now supports `java17` as a parameter.
May 18, 2023	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM no longer supports `nodejs12.x` as a parameter.

Date of change	Control ID and title	Description of change
April 23, 2023	[ECS.10] ECS Fargate services should run on the latest Fargate platform version	ECS.10 checks whether Amazon ECS Fargate services are running the latest Fargate platform version. Customers can deploy Amazon ECS through ECS directly, or by using CodeDeploy. Security Hub CSPM updated this control to produce **Passed** findings when you use CodeDeploy to deploy ECS Fargate services.
April 20, 2023	[S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts	S3.6 checks whether an Amazon Simple Storage Service (Amazon S3) bucket policy prevents principals from other AWS accounts from performing denied actions on resources in the S3 bucket. Security Hub CSPM updated the control to account for conditionals in a bucket policy.

Date of change	Control ID and title	Description of change
April 18, 2023	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM now supports `python3.10` as a parameter.
April 18, 2023	[Lambda.2] Lambda functions should use supported runtimes	Lambda.2 checks whether the AWS Lambda function settings for runtimes match the expected values set for the supported runtimes in each language. Security Hub CSPM no longer supports `dotnetcore3.1` as a parameter.

Date of change	Control ID and title	Description of change
April 17, 2023	[RDS.11] RDS instances should have automatic backups enabled	RDS.11 checks whether Amazon RDS instances have automated backups enabled, with a backup retention period that's greater than or equal to seven days. Security Hub CSPM updated this control to exclude read replicas from evaluation, as not all engines support automated backups on read replicas. Additionally, RDS doesn't provide the option to specify a backup retention period when creating read replicas. Read replicas are created with a backup retention period of `0` by default.

Security Hub controls for AWS accounts

These Security Hub controls evaluate AWS accounts.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Account.1] Security contact information should be provided for an AWS account

Related requirements: NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Identify > Resource Configuration

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `security-account-information-provided`

Schedule type: Periodic

Parameters: None

This control checks if an Amazon Web Services (AWS) account has security contact information. The control fails if security contact information is not provided for the account.

Alternate security contacts allow AWS to contact another person about issues with your account in case you're unavailable. Notifications can be from Support, or other AWS service teams about security-related topics associated with your AWS account usage.

Remediation

To add an alternate contact as a security contact to your AWS account, see Update the alternate contacts for your AWS account in the *AWS Account Management Reference Guide*.

[Account.2] AWS accounts should be part of an AWS Organizations organization

Category: Protect > Secure access management > Access control

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: `account-part-of-organizations`

Schedule type: Periodic

Parameters: None

This control checks if an AWS account is part of an organization managed through AWS Organizations. The control fails if the account is not part of an organization.

Organizations helps you centrally manage your environment as you scale your workloads on AWS. You can use multiple AWS accounts to isolate workloads that have specific security requirements, or to comply with frameworks such as HIPAA or PCI. By creating an organization, you can administer multiple accounts as a single unit and centrally manage their access to AWS services, resources, and Regions.

Remediation

To create a new organization and automatically add AWS accounts to it, see Creating an organization in the *AWS Organizations User Guide*. To add accounts to an existing organization, see Inviting an AWS account to join your organization in the *AWS Organizations User Guide*.

Security Hub controls for AWS Amplify

These Security Hub controls evaluate the AWS Amplify service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Amplify.1] Amplify apps should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Amplify::App`

AWS Config rule: amplify-app-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Amplify app has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the app doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the app doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an AWS Amplify app, see Resource tagging support in the *AWS Amplify Hosting User Guide*.

[Amplify.2] Amplify branches should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Amplify::Branch`

AWS Config rule: amplify-branch-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Amplify branch has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the branch doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the branch doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

Note

Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an AWS Amplify branch, see Resource tagging support in the *AWS Amplify Hosting User Guide*.

Security Hub controls for Amazon API Gateway

These AWS Security Hub controls evaluate the Amazon API Gateway service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[APIGateway.1] API Gateway REST and WebSocket API execution logging should be enabled

Related requirements: NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::ApiGateway::Stage`, `AWS::ApiGatewayV2::Stage`

AWS Config rule: api-gw-execution-logging-enabled

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`loggingLevel`	Logging level	Enum	`ERROR`, `INFO`	`No default value`

This control checks whether all stages of an Amazon API Gateway REST or WebSocket API have logging enabled. The control fails if the `loggingLevel` isn't `ERROR` or `INFO` for all stages of the API. Unless you provide custom parameter values to indicate that a specific log type should be enabled, Security Hub produces a passed finding if the logging level is either `ERROR` or `INFO`.

API Gateway REST or WebSocket API stages should have relevant logs enabled. API Gateway REST and WebSocket API execution logging provides detailed records of requests made to API Gateway REST and WebSocket API stages. The stages include API integration backend responses, Lambda authorizer responses, and the `requestId` for AWS integration endpoints.

Remediation

To enable logging for REST and WebSocket API operations, see Set up CloudWatch API logging using the API Gateway console in the *API Gateway Developer Guide*.

[APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), NIST.800-171.r2 3.13.15

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::ApiGateway::Stage`

AWS Config rule: api-gw-ssl-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon API Gateway REST API stages have SSL certificates configured. Backend systems use these certificates to authenticate that incoming requests are from API Gateway.

API Gateway REST API stages should be configured with SSL certificates to allow backend systems to authenticate that requests originate from API Gateway.

Remediation

For detailed instructions on how to generate and configure API Gateway REST API SSL certificates, see Generate and configure an SSL certificate for backend authentication in the *API Gateway Developer Guide*.

[APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled

Related requirements: NIST.800-53.r5 CA-7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::ApiGateway::Stage`

AWS Config rule: api-gw-xray-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether AWS X-Ray active tracing is enabled for your Amazon API Gateway REST API stages.

X-Ray active tracing enables a more rapid response to performance changes in the underlying infrastructure. Changes in performance could result in a lack of availability of the API. X-Ray active tracing provides real-time metrics of user requests that flow through your API Gateway REST API operations and connected services.

Remediation

For detailed instructions on how to enable X-Ray active tracing for API Gateway REST API operations, see Amazon API Gateway active tracing support for AWS X-Ray in the *AWS X-Ray Developer Guide*.

[APIGateway.4] API Gateway should be associated with a WAF Web ACL

Related requirements: NIST.800-53.r5 AC-4(21)

Category: Protect > Protective services

Severity: Medium

Resource type: `AWS::ApiGateway::Stage`

AWS Config rule: api-gw-associated-with-waf

Schedule type: Change triggered

Parameters: None

This control checks whether an API Gateway stage uses an AWS WAF web access control list (ACL). This control fails if an AWS WAF web ACL is not attached to a REST API Gateway stage.

AWS WAF is a web application firewall that helps protect web applications and APIs from attacks. It enables you to configure an ACL, which is a set of rules that allow, block, or count web requests

based on customizable web security rules and conditions that you define. Ensure that your API Gateway stage is associated with an AWS WAF web ACL to help protect it from malicious attacks.

Remediation

For information on how to use the API Gateway console to associate an AWS WAF Regional web ACL with an existing API Gateway API stage, see Using AWS WAF to protect your APIs in the *API Gateway Developer Guide*.

[APIGateway.5] API Gateway REST API cache data should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: `AWS::ApiGateway::Stage`

AWS Config rule: `api-gw-cache-encrypted` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether all methods in API Gateway REST API stages that have cache enabled are encrypted. The control fails if any method in an API Gateway REST API stage is configured to cache and the cache is not encrypted. Security Hub evaluates the encryption of a particular method only when caching is enabled for that method.

Encrypting data at rest reduces the risk of data stored on disk being accessed by a user not authenticated to AWS. It adds another set of access controls to limit unauthorized users ability access the data. For example, API permissions are required to decrypt the data before it can be read.

API Gateway REST API caches should be encrypted at rest for an added layer of security.

Remediation

To configure API caching for a stage, see Enable Amazon API Gateway caching in the *API Gateway Developer Guide*. In **Cache Settings**, choose **Encrypt cache data**.

[APIGateway.8] API Gateway routes should specify an authorization type

Related requirements: NIST.800-53.r5 AC-3, NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Protect > Secure Access Management

Severity: Medium

Resource type: `AWS::ApiGatewayV2::Route`

AWS Config rule: api-gwv2-authorization-type-configured

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`authorizationType`	Authorization type of the API routes	Enum	`AWS_IAM`, `CUSTOM`, `JWT`	No default value

This control checks if Amazon API Gateway routes have an authorization type. The control fails if the API Gateway route doesn't have any authorization type. Optionally, you can provide a custom parameter value if you want the control to pass only if the route uses the authorization type specified in the `authorizationType` parameter.

API Gateway supports multiple mechanisms for controlling and managing access to your API. By specifying an authorization type, you can restrict access to your API to only authorized users or processes.

Remediation

To set an authorization type for HTTP APIs, see Controlling and managing access to an HTTP API in API Gateway in the *API Gateway Developer Guide*. To set an authorization type for WebSocket APIs, see Controlling and managing access to a WebSocket API in API Gateway in the *API Gateway Developer Guide*.

[APIGateway.9] Access logging should be configured for API Gateway V2 Stages

Related requirements: NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::ApiGatewayV2::Stage`

AWS Config rule: api-gwv2-access-logs-enabled

Schedule type: Change triggered

Parameters: None

This control checks if Amazon API Gateway V2 stages have access logging configured. This control fails if access log settings aren't defined.

API Gateway access logs provide detailed information about who has accessed your API and how the caller accessed the API. These logs are useful for applications such as security and access audits and forensics investigation. Enable these access logs to analyze traffic patterns and to troubleshoot issues.

For additional best practices, see Monitoring REST APIs in the *API Gateway Developer Guide*.

Remediation

To set up access logging, see Set up CloudWatch API logging using the API Gateway console in the *API Gateway Developer Guide*.

Security Hub controls for AWS AppConfig

These Security Hub controls evaluate the AWS AppConfig service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[AppConfig.1] AWS AppConfig applications should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AppConfig::Application`

AWS Config rule: `appconfig-application-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS AppConfig application has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the application doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the application isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS AppConfig application, see TagResource in the *AWS AppConfig API Reference*.

[AppConfig.2] AWS AppConfig configuration profiles should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AppConfig::ConfigurationProfile`

AWS Config rule: `appconfig-configuration-profile-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS AppConfig configuration profile has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the configuration profile doesn't

have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the configuration profile isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS AppConfig configuration profile, see TagResource in the *AWS AppConfig API Reference*.

[AppConfig.3] AWS AppConfig environments should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AppConfig::Environment`

AWS Config rule: `appconfig-environment-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS AppConfig environment has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the environment doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the environment isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS AppConfig environment, see TagResource in the *AWS AppConfig API Reference*.

[AppConfig.4] AWS AppConfig extension associations should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AppConfig::ExtensionAssociation`

AWS Config rule: `appconfig-extension-association-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS AppConfig extension association has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the extension association doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the extension association isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other

criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS AppConfig extension association, see TagResource in the *AWS AppConfig API Reference*.

Security Hub controls for Amazon AppFlow

These Security Hub controls evaluate the Amazon AppFlow service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[AppFlow.1] Amazon AppFlow flows should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AppFlow::Flow`

AWS Config rule: `appflow-flow-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon AppFlow flow has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the flow doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the flow isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an Amazon AppFlow flow, see Creating flows in Amazon AppFlow in the *Amazon AppFlow User Guide*.

Security Hub controls for AWS App Runner

These AWS Security Hub controls evaluate the AWS App Runner service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[AppRunner.1] App Runner services should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AppRunner::Service`

AWS Config rule: apprunner-service-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS App Runner service has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the App Runner service doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the

parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the App Runner service isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

For information about adding tags to an AWS App Runner service, see TagResource in the *AWS App Runner API Reference*.

[AppRunner.2] App Runner VPC connectors should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AppRunner::VpcConnector`

AWS Config rule: apprunner-vpc-connector-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS App Runner VPC connector has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the VPC connector doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the VPC connector isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

For information about adding tags to an AWS App Runner VPC connector, see TagResource in the *AWS App Runner API Reference*.

Security Hub controls for AWS AppSync

These Security Hub controls evaluate the AWS AppSync service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[AppSync.1] AWS AppSync API caches should be encrypted at rest

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::AppSync::GraphQLApi`

AWS Config rule: appsync-cache-ct-encryption-at-rest

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS AppSync API cache is encrypted at rest. The control fails if the API cache isn't encrypted at rest.

Data at rest refers to data that's stored in persistent, non-volatile storage for any duration. Encrypting data at rest helps you protect its confidentiality, which reduces the risk that an unauthorized user can access it.

Remediation

You can't change the encryption settings after enabling caching for your AWS AppSync API. Instead, you must delete the cache and and recreate it with encryption enabled. For more information, see Cache encryption in the *AWS AppSync Developer Guide*.

[AppSync.2] AWS AppSync should have field-level logging enabled

Related requirements: PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::AppSync::GraphQLApi`

AWS Config rule: `appsync-logging-enabled`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`fieldLoggingLevel`	Field logging level	Enum	`ERROR`, `ALL`, `INFO`, `DEBUG`	No `default value`

This control checks whether an AWS AppSync API has field-level logging turned on. The control fails if the field resolver log level is set to **None**. Unless you provide custom parameter values to indicate that a specific log type should be enabled, Security Hub produces a passed finding if the field resolver log level is either `ERROR` or `ALL`.

You can use logging and metrics to identify, troubleshoot, and optimize your GraphQL queries. Turning on logging for AWS AppSync GraphQL helps you get detailed information about API requests and responses, identify and respond to issues, and comply with regulatory requirements.

Remediation

To turn on logging for AWS AppSync, see Setup and configuration in the *AWS AppSync Developer Guide*.

[AppSync.4] AWS AppSync GraphQL APIs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AppSync::GraphQLApi`

AWS Config rule: `tagged-appsync-graphqlapi` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS AppSync GraphQL API has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the GraphQL API doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the GraphQL API isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS AppSync GraphQL API, see TagResource in the *AWS AppSync API Reference*.

[AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6

Category: Protect > Secure access management > Passwordless authentication

Severity: High

Resource type: `AWS::AppSync::GraphQLApi`

AWS Config rule: `appsync-authorization-check`

Schedule type: Change triggered

Parameters:

- `AllowedAuthorizationTypes: AWS_LAMBDA, AWS_IAM, OPENID_CONNECT, AMAZON_COGNITO_USER_POOLS` (not customizable)

This control checks whether your application uses an API key to interact with an AWS AppSync GraphQL API. The control fails if an AWS AppSync GraphQL API is authenticated with an API key.

An API key is a hard-coded value in your application that is generated by the AWS AppSync service when you create an unauthenticated GraphQL endpoint. If this API key is compromised, your endpoint is vulnerable to unintended access. Unless you are supporting a publicly accessible application or website, we don't recommend using an API key for authentication.

Remediation

To set an authorization option for your AWS AppSync GraphQL API, see Authorization and authentication in the *AWS AppSync Developer Guide*.

[AppSync.6] AWS AppSync API caches should be encrypted in transit

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::AppSync::ApiCache`

AWS Config rule: appsync-cache-ct-encryption-in-transit

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS AppSync API cache is encrypted in transit. The control fails if the API cache isn't encrypted in transit.

Data in transit refers to data that moves from one location to another, such as between nodes in your cluster or between your cluster and your application. Data may move across the internet or within a private network. Encrypting data in transit reduces the risk that an unauthorized user can eavesdrop on network traffic.

Remediation

You can't change the encryption settings after enabling caching for your AWS AppSync API. Instead, you must delete the cache and and recreate it with encryption enabled. For more information, see Cache encryption in the *AWS AppSync Developer Guide*.

Security Hub controls for Amazon Athena

These AWS Security Hub controls evaluate the Amazon Athena service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Athena.1] Athena workgroups should be encrypted at rest

> **Important**
>
> Security Hub retired this control in April 2024. For more information, see Change log for Security Hub CSPM controls.

Category: Protect > Data protection > Encryption of data at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Severity: Medium

Resource type: `AWS::Athena::WorkGroup`

AWS Config rule: athena-workgroup-encrypted-at-rest

Schedule type: Change triggered

Parameters: None

This control checks if an Athena workgroup is encrypted at rest. The control fails if an Athena workgroup isn't encrypted at rest.

In Athena, you can create workgroups for running queries for teams, applications, or different workloads. Each workgroup has a setting to enable encryption on all queries. You have the option to use server-side encryption with Amazon Simple Storage Service (Amazon S3) managed keys, server-side encryption with AWS Key Management Service (AWS KMS) keys, or client-side encryption with customer managed KMS keys. Data at rest refers to any data that's stored in persistent, non-volatile storage for any duration. Encryption helps you protect the confidentiality of such data, reducing the risk that an unauthorized user can access it.

Remediation

To enable encryption at rest for Athena workgroups, see Edit a workgroup in the *Amazon Athena User Guide*. In the **Query Result Configuration** section, select **Encrypt query results**.

[Athena.2] Athena data catalogs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Athena::DataCatalog`

AWS Config rule: `tagged-athena-datacatalog` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon Athena data catalog has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the data catalog doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the data catalog isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Athena data catalog, see Tagging Athena resources in the *Amazon Athena User Guide*.

[Athena.3] Athena workgroups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Athena::WorkGroup`

AWS Config rule: `tagged-athena-workgroup` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon Athena workgroup has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the workgroup doesn't have any tag keys

or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the workgroup isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Athena workgroup, see Adding and deleting tags on an individual workgroup in the *Amazon Athena User Guide*.

[Athena.4] Athena workgroups should have logging enabled

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::Athena::WorkGroup`

AWS Config rule: athena-workgroup-logging-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Athena workgroup has logging enabled. The control fails if the workgroup doesn't have logging enabled.

Audit logs track and monitor system activities. They provide a record of events that can help you detect security breaches, investigate incidents, and comply with regulations. Audit logs also enhance the overall accountability and transparency of your organization.

Remediation

For information about enabling logging for an Athena workgroup, see Enable CloudWatch query metrics in Athena in the *Amazon Athena User Guide*.

Security Hub controls for AWS Backup

These Security Hub controls evaluate the AWS Backup service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Backup.1] AWS Backup recovery points should be encrypted at rest

Related requirements: NIST.800-53.r5 CP-9(8), NIST.800-53.r5 SI-12

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::Backup::RecoveryPoint`

AWS Config rule: `backup-recovery-point-encrypted`

Schedule type: Change triggered

Parameters: None

This control checks if an AWS Backup recovery point is encrypted at rest. The control fails if the recovery point isn't encrypted at rest.

An AWS Backup recovery point refers to a specific copy or snapshot of data that is created as part of a backup process. It represents a particular moment in time when the data was backed up and serves as a restore point in case the original data becomes lost, corrupted, or inaccessible.

Encrypting the backup recovery points adds an extra layer of protection against unauthorized access. Encryption is a best practice to protect the confidentiality, integrity, and security of backup data.

Remediation

To encrypt an AWS Backup recovery point, see Encryption for backups in AWS Backup in the *AWS Backup Developer Guide*.

[Backup.2] AWS Backup recovery points should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Backup::RecoveryPoint`

AWS Configrule: `tagged-backup-recoverypoint` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Backup recovery point has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the recovery point doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the recovery point isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS Backup recovery point

1. Open the AWS Backup console at https://console.aws.amazon.com/backup.
2. In the navigation pane, choose **Backup plans**.
3. Select a backup plan from the list.
4. In the **Backup plan tags** section, choose **Manage tags**.
5. Enter the key and value for the tag. Choose **Add new tag** for additional key-value pairs.
6. When you are finished adding tags, choose **Save**.

[Backup.3] AWS Backup vaults should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Backup::BackupVault`

AWS Configrule: `tagged-backup-backupvault` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Backup vault has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the recovery point doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the recovery point isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS Backup vault

1. Open the AWS Backup console at https://console.aws.amazon.com/backup.
2. In the navigation pane, choose **Backup vaults**.
3. Select a backup vault from the list.
4. In the **Backup vault tags** section, choose **Manage tags**.
5. Enter the key and value for the tag. Choose **Add new tag** for additional key-value pairs.
6. When you are finished adding tags, choose **Save**.

[Backup.4] AWS Backup report plans should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Backup::ReportPlan`

AWS Configrule: `tagged-backup-reportplan` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Backup report plan has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the report plan doesn't have any tag keys

or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the report plan isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS Backup report plan

1. Open the AWS Backup console at https://console.aws.amazon.com/backup.
2. In the navigation pane, choose **Backup vaults**.
3. Select a backup vault from the list.
4. In the **Backup vault tags** section, choose **Manage tags**.
5. Choose **Add new tag**. Enter the key and value for the tag. Repeat for additional key-value pairs.
6. When you are finished adding tags, choose **Save**.

[Backup.5] AWS Backup backup plans should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Backup::BackupPlan`

AWS Configrule: `tagged-backup-backupplan` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Backup backup plan has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the backup plan doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the backup plan isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS Backup backup plan

1. Open the AWS Backup console at https://console.aws.amazon.com/backup.
2. In the navigation pane, choose **Backup vaults**.
3. Select a backup vault from the list.
4. In the **Backup vault tags** section, choose **Manage tags**.
5. Choose **Add new tag**. Enter the key and value for the tag. Repeat for additional key-value pairs.
6. When you are finished adding tags, choose **Save**.

Security Hub controls for AWS Batch

These Security Hub controls evaluate the AWS Batch service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Batch.1] Batch job queues should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Batch::JobQueue`

AWS Config rule: batch-job-queue-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Batch job queue has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the job queue doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the job queue isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to a Batch job queue, see Tag your resources in the *AWS Batch User Guide*.

[Batch.2] Batch scheduling policies should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Batch::SchedulingPolicy`

AWS Config rule: batch-scheduling-policy-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Batch scheduling policy has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the scheduling policy doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the scheduling policy isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which

defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to a Batch scheduling policy, see Tag your resources in the *AWS Batch User Guide*.

[Batch.3] Batch compute environments should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Batch::ComputeEnvironment`

AWS Config rule: batch-compute-environment-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Batch compute environment has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the compute environment doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the compute environment isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to a Batch compute environment, see Tag your resources in the *AWS Batch User Guide*.

[Batch.4] Compute resources properties in managed Batch compute environments should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Batch::ComputeEnvironment`

AWS Config rule: batch-managed-compute-env-compute-resources-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether the compute resources property in a managed AWS Batch compute environment has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the compute resources property doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if a compute resources property doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix. This control doesn't evaluate unmanaged compute environments, or managed environments that use AWS Fargate resources.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to compute resources in a managed AWS Batch compute environment, see Tag your resources in the *AWS Batch User Guide*.

Security Hub controls for AWS Certificate Manager

These AWS Security Hub controls evaluate the AWS Certificate Manager (ACM) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period

Related requirements: NIST.800-53.r5 SC-28(3), NIST.800-53.r5 SC-7(16), NIST.800-171.r2 3.13.15, PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::ACM::Certificate`

AWS Config rule: acm-certificate-expiration-check

Schedule type: Change triggered and periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`daysToExpiration`	Number of days within which the ACM certificate must be renewed	Integer	14 to 365	`30`

This control checks whether an AWS Certificate Manager (ACM) certificate is renewed within the specified time period. It checks both imported certificates and certificates provided by ACM. The control fails if the certificate isn't renewed within the specified time period. Unless you provide a custom parameter value for the renewal period, Security Hub uses a default value of 30 days.

ACM can automatically renew certificates that use DNS validation. For certificates that use email validation, you must respond to a domain validation email. ACM doesn't automatically renew certificates that you import. You must renew imported certificates manually.

Remediation

ACM provides managed renewal for your SSL/TLS certificates issued by Amazon. This means that ACM either renews your certificates automatically (if you use DNS validation), or it sends you email notices when the certificate expiration approaches. These services are provided for both public and private ACM certificates.

For domains validated by email

When a certificate is 45 days from expiration, ACM sends to the domain owner an email for each domain name. To validate the domains and complete the renewal, you must respond to the email notifications.

For more information, see Renewal for domains validated by email in the *AWS Certificate Manager User Guide*.

For domains validated by DNS

ACM automatically renews certificates that use DNS validation. 60 days before the expiration, ACM verifies that the certificate can be renewed.

If it cannot validate a domain name, then ACM sends a notification that manual validation is required. It sends these notifications 45 days, 30 days, 7 days, and 1 day before the expiration.

For more information, see Renewal for domains validated by DNS in the *AWS Certificate Manager User Guide*.

[ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits

Related requirements: PCI DSS v4.0.1/4.2.1

Category: Identify > Inventory > Inventory services

Severity: High

Resource type: `AWS::ACM::Certificate`

AWS Config rule: acm-certificate-rsa-check

Schedule type: Change triggered

Parameters: None

This control checks whether RSA certificates managed by AWS Certificate Manager use a key length of at least 2,048 bits. The control fails if the key length is smaller than 2,048 bits.

The strength of encryption directly correlates with key size. We recommend key lengths of at least 2,048 bits to protect your AWS resources as computing power becomes less expensive and servers become more advanced.

Remediation

The minimum key length for RSA certificates issued by ACM is already 2,048 bits. For instructions on issuing new RSA certificates with ACM, see Issuing and managing certificates in the *AWS Certificate Manager User Guide*.

While ACM allows you to import certificates with shorter key lengths, you must use keys of at least 2,048 bits to pass this control. You can't change the key length after importing a certificate. Instead, you must delete certificates with a key length smaller than 2,048 bits. For more information about importing certificates into ACM, see Prerequisites for importing certificates in the *AWS Certificate Manager User Guide*.

[ACM.3] ACM certificates should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::ACM::Certificate`

AWS Config rule: `tagged-acm-certificate` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Certificate Manager (ACM) certificate has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the certificate doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the certificate isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an ACM certificate, see Tagging AWS Certificate Manager certificates in the *AWS Certificate Manager User Guide*.

Security Hub controls for AWS CloudFormation

These Security Hub controls evaluate the AWS CloudFormation service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)

> **Important**
>
> Security Hub retired this control in April 2024. For more information, see Change log for Security Hub CSPM controls.

Related requirements: NIST.800-53.r5 SI-4(12), NIST.800-53.r5 SI-4(5)

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: `AWS::CloudFormation::Stack`

AWS Config rule: `cloudformation-stack-notification-check`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Simple Notification Service notification is integrated with an AWS CloudFormation stack. The control fails for a CloudFormation stack if no SNS notification is associated with it.

Configuring an SNS notification with your CloudFormation stack helps immediately notify stakeholders of any events or changes occurring with the stack.

Remediation

To integrate a CloudFormation stack and an SNS topic, see Updating stacks directly in the *AWS CloudFormation User Guide*.

[CloudFormation.2] CloudFormation stacks should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::CloudFormation::Stack`

AWS Config rule: `tagged-cloudformation-stack` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS CloudFormation stack has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the stack doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the stack isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other

criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a CloudFormation stack, see CreateStack in the *AWS CloudFormation API Reference*.

Security Hub controls for Amazon CloudFront

These AWS Security Hub controls evaluate the Amazon CloudFront service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[CloudFront.1] CloudFront distributions should have a default root object configured

Related requirements: NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), PCI DSS v4.0.1/2.2.6

Category: Protect > Secure access management > Resources not publicly accessible

Severity: High

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-default-root-object-configured

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon CloudFront distribution is configured to return a specific object that is the default root object. The control fails if the CloudFront distribution does not have a default root object configured.

A user might sometimes request the distribution's root URL instead of an object in the distribution. When this happens, specifying a default root object can help you to avoid exposing the contents of your web distribution.

Remediation

To configure a default root object for a CloudFront distribution, see How to specify a default root object in the *Amazon CloudFront Developer Guide*.

[CloudFront.3] CloudFront distributions should require encryption in transit

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-viewer-policy-https

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon CloudFront distribution requires viewers to use HTTPS directly or whether it uses redirection. The control fails if `ViewerProtocolPolicy` is set to `allow-all` for `defaultCacheBehavior` or for `cacheBehaviors`.

HTTPS (TLS) can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. Only encrypted connections over HTTPS (TLS) should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the impact of TLS.

Remediation

To encrypt a CloudFront distribution in transit, see Requiring HTTPS for communication between viewers and CloudFront in the *Amazon CloudFront Developer Guide*.

[CloudFront.4] CloudFront distributions should have origin failover configured

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Low

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-origin-failover-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon CloudFront distribution is configured with an origin group that has two or more origins.

CloudFront origin failover can increase availability. Origin failover automatically redirects traffic to a secondary origin if the primary origin is unavailable or if it returns specific HTTP response status codes.

Remediation

To configure origin failover for a CloudFront distribution, see Creating an origin group in the *Amazon CloudFront Developer Guide*.

[CloudFront.5] CloudFront distributions should have logging enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-accesslogs-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether server access logging is enabled on CloudFront distributions. The control fails if access logging is not enabled for a distribution. This control only evaluates whether standard logging (legacy) is enabled for a distribution.

CloudFront access logs provide detailed information about every user request that CloudFront receives. Each log contains information such as the date and time the request was received, the IP address of the viewer that made the request, the source of the request, and the port number of the request from the viewer. These logs are useful for applications such as security and access audits and forensics investigation. For more information about analyzing access logs, see Query Amazon CloudFront logs in the *Amazon Athena User Guide*.

Remediation

To configure standard logging (legacy) for a CloudFront distribution, see Configure standard logging (legacy) in the *Amazon CloudFront Developer Guide*.

[CloudFront.6] CloudFront distributions should have WAF enabled

Related requirements: NIST.800-53.r5 AC-4(21), PCI DSS v4.0.1/6.4.2

Category: Protect > Protective services

Severity: Medium

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-associated-with-waf

Schedule type: Change triggered

Parameters: None

This control checks whether CloudFront distributions are associated with either AWS WAF Classic or AWS WAF web ACLs. The control fails if the distribution is not associated with a web ACL.

AWS WAF is a web application firewall that helps protect web applications and APIs from attacks. It allows you to configure a set of rules, called a web access control list (web ACL), that allow, block, or count web requests based on customizable web security rules and conditions that you define. Ensure your CloudFront distribution is associated with an AWS WAF web ACL to help protect it from malicious attacks.

Remediation

To associate an AWS WAF web ACL with a CloudFront distribution, see Using AWS WAF to control access to your content in the *Amazon CloudFront Developer Guide*.

[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), NIST.800-171.r2 3.13.15

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-custom-ssl-certificate

Schedule type: Change triggered

Parameters: None

This control checks whether CloudFront distributions are using the default SSL/TLS certificate CloudFront provides. This control passes if the CloudFront distribution uses a custom SSL/TLS certificate. This control fails if the CloudFront distribution uses the default SSL/TLS certificate.

Custom SSL/TLS allow your users to access content by using alternate domain names. You can store custom certificates in AWS Certificate Manager (recommended), or in IAM.

Remediation

To add an alternate domain name for a CloudFront distribution using a custom SSL/TLS certificate, see Adding an alternate domain name in the *Amazon CloudFront Developer Guide*.

[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Protect > Secure network configuration

Severity: Low

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-sni-enabled

Schedule type: Change triggered

Parameters: None

This control checks if Amazon CloudFront distributions are using a custom SSL/TLS certificate and are configured to use SNI to serve HTTPS requests. This control fails if a custom SSL/TLS certificate is associated but the SSL/TLS support method is a dedicated IP address.

Server Name Indication (SNI) is an extension to the TLS protocol that is supported by browsers and clients released after 2010. If you configure CloudFront to serve HTTPS requests using SNI, CloudFront associates your alternate domain name with an IP address for each edge location. When a viewer submits an HTTPS request for your content, DNS routes the request to the IP address for the correct edge location. The IP address to your domain name is determined during the SSL/TLS handshake negotiation; the IP address isn't dedicated to your distribution.

Remediation

To configure a CloudFront distribution to use SNI to serve HTTPS requests, see Using SNI to Serve HTTPS Requests (works for Most Clients) in the CloudFront Developer Guide. For information about custom SSL certificates, see Requirements for using SSL/TLS certificates with CloudFront.

[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3),

NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-traffic-to-origin-encrypted

Schedule type: Change triggered

Parameters: None

This control checks if Amazon CloudFront distributions are encrypting traffic to custom origins. This control fails for a CloudFront distribution whose origin protocol policy allows 'http-only'. This control also fails if the distribution's origin protocol policy is 'match-viewer' while the viewer protocol policy is 'allow-all'.

HTTPS (TLS) can be used to help prevent eavesdropping or manipulation of network traffic. Only encrypted connections over HTTPS (TLS) should be allowed.

Remediation

To update the Origin Protocol Policy to require encryption for a CloudFront connection, see Requiring HTTPS for communication between CloudFront and your custom origin in the *Amazon CloudFront Developer Guide*.

[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), NIST.800-171.r2 3.13.15, PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-no-deprecated-ssl-protocols

Schedule type: Change triggered

Parameters: None

This control checks if Amazon CloudFront distributions are using deprecated SSL protocols for HTTPS communication between CloudFront edge locations and your custom origins. This control fails if a CloudFront distribution has a `CustomOriginConfig` where `OriginSslProtocols` includes `SSLv3`.

In 2015, the Internet Engineering Task Force (IETF) officially announced that SSL 3.0 should be deprecated due to the protocol being insufficiently secure. It is recommended that you use TLSv1.2 or later for HTTPS communication to your custom origins.

Remediation

To update the Origin SSL Protocols for a CloudFront distribution, see Requiring HTTPS for communication between CloudFront and your custom origin in the *Amazon CloudFront Developer Guide*.

[CloudFront.12] CloudFront distributions should not point to non-existent S3 origins

Related requirements: NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), PCI DSS v4.0.1/2.2.6

Category: Identify > Resource configuration

Severity: High

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-s3-origin-non-existent-bucket

Schedule type: Periodic

Parameters: None

This control checks whether Amazon CloudFront distributions are pointing to non-existent Amazon S3 origins. The control fails for a CloudFront distribution if the origin is configured to point to

a non-existent bucket. This control only applies to CloudFront distributions where an S3 bucket without static website hosting is the S3 origin.

When a CloudFront distribution in your account is configured to point to a non-existent bucket, a malicious third party can create the referenced bucket and serve their own content through your distribution. We recommend checking all origins regardless of routing behavior to ensure that your distributions are pointing to appropriate origins.

Remediation

To modify a CloudFront distribution to point to a new origin, see Updating a distribution in the *Amazon CloudFront Developer Guide*.

[CloudFront.13] CloudFront distributions should use origin access control

Category: Protect > Secure access management > Resource not publicly accessible

Severity: Medium

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-s3-origin-access-control-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon CloudFront distribution with an Amazon S3 origin has origin access control (OAC) configured. The control fails if OAC isn't configured for the CloudFront distribution.

When using an S3 bucket as an origin for your CloudFront distribution, you can enable OAC. This permits access to the content in the bucket only through the specified CloudFront distribution, and prohibits access directly from the bucket or another distribution. Although CloudFront supports Origin Access Identity (OAI), OAC offers additional functionality, and distributions using OAI can migrate to OAC. While OAI provides a secure way to access S3 origins, it has limitations, such as lack of support for granular policy configurations and for HTTP/HTTPS requests that use the POST method in AWS Regions that require AWS Signature Version 4 (SigV4). OAI also doesn't support encryption with AWS Key Management Service. OAC is based on an AWS best practice of using IAM service principals to authenticate with S3 origins.

Remediation

To configure OAC for a CloudFront distribution with S3 origins, see Restricting access to an Amazon S3 origin in the *Amazon CloudFront Developer Guide*.

[CloudFront.14] CloudFront distributions should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule:`tagged-cloudfront-distribution` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon CloudFront distribution has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the distribution doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the distribution isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps

you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a CloudFront distribution, see Tagging Amazon CloudFront distributions in the *Amazon CloudFront Developer Guide*.

[CloudFront.15] CloudFront distributions should use the recommended TLS security policy

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-ssl-policy-check

Schedule type: Change triggered

Parameters: `securityPolicies`: `TLSv1.2_2021` (not customizable)

This control checks whether an Amazon CloudFront distribution is configured to use the recommended TLS security policy. The control fails if the CloudFront distribution is not configured to use the recommended TLS security policy.

If you configure an Amazon CloudFront distribution to require viewers to use HTTPS to access content, you have to choose a security policy and specify the minimum SSL/TLS protocol version to use. This determines which protocol version CloudFront uses to communicate with viewers, and the ciphers that CloudFront uses to encrypt the communications. We recommend using the latest

security policy that CloudFront provides. This ensures that CloudFront uses the latest cipher suites to encrypt data in transit between a viewer and a CloudFront distribution.

> **Note**
>
> This control generates findings only for CloudFront distributions that are configured to use custom SSL certificates and are not configured to support legacy clients.

Remediation

For information about configuring the security policy for a CloudFront distribution, see Update a distribution in the *Amazon CloudFront Developer Guide*. When you configure the security policy for a distribution, choose the latest security policy.

[CloudFront.16] CloudFront distributions should use origin access control for Lambda function URL origins

Category: Protect > Secure access management > Access control

Severity: Medium

Resource type: `AWS::CloudFront::Distribution`

AWS Config rule: cloudfront-origin-lambda-url-oac-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon CloudFront distribution with an AWS Lambda function URL as an origin has origin access control (OAC) enabled. The control fails if the CloudFront distribution has a Lambda function URL as an origin and OAC isn't enabled.

An AWS Lambda function URL is a dedicated HTTPS endpoint for a Lambda function. If a Lambda function URL is the origin for a CloudFront distribution, the function URL must be publicly accessible. Therefore, as a security best practice, you should create an OAC and add it to the Lambda function URL in a distribution. OAC uses IAM service principals to authenticate requests between CloudFront and the function URL. It also supports the use of resource-based policies to allow invocation of a function only if a request is on behalf of a CloudFront distribution specified in the policy.

Remediation

For information about configuring OAC for an Amazon CloudFront distribution that uses a Lambda function URL as an origin, see Restrict access to an AWS Lambda function URL origin in the *Amazon CloudFront Developer Guide*.

Security Hub controls for AWS CloudTrail

These AWS Security Hub controls evaluate the AWS CloudTrail service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events

Related requirements: CIS AWS Foundations Benchmark v1.2.0/2.1, CIS AWS Foundations Benchmark v1.4.0/3.1, CIS AWS Foundations Benchmark v3.0.0/3.1, NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-14(1), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), NIST.800-53.r5 SA-8(22)

Category: Identify > Logging

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: multi-region-cloudtrail-enabled

Schedule type: Periodic

Parameters:

- `readWriteType`: `ALL` (not customizable)

  `includeManagementEvents`: `true` (not customizable)

This control checks whether there is at least one multi-Region AWS CloudTrail trail that captures read and write management events. The control fails if CloudTrail is disabled or if there isn't at least one CloudTrail trail that captures read and write management events.

AWS CloudTrail records AWS API calls for your account and delivers log files to you. The recorded information includes the following information:

- Identity of the API caller
- Time of the API call
- Source IP address of the API caller
- Request parameters
- Response elements returned by the AWS service

CloudTrail provides a history of AWS API calls for an account, including API calls made from the AWS Management Console, AWS SDKs, command line tools. The history also includes API calls from higher-level AWS services such as AWS CloudFormation.

The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing. Multi-Region trails also provide the following benefits.

- A multi-Region trail helps to detect unexpected activity occurring in otherwise unused Regions.
- A multi-Region trail ensures that global service event logging is enabled for a trail by default. Global service event logging records events generated by AWS global services.
- For a multi-Region trail, management events for all read and write operations ensure that CloudTrail records management operations on all resources in an AWS account.

By default, CloudTrail trails that are created using the AWS Management Console are multi-Region trails.

Remediation

To create a new multi-Region trail in CloudTrail, see Creating a trail in the *AWS CloudTrail User Guide*. Use the following values:

Field	Value
Additional settings, Log file validation	Enabled
Choose log events, Management events, API activity	**Read** and **Write**. Clear check boxes for exclusions.

To update an existing trail, see Updating a trail in the *AWS CloudTrail User Guide*. In **Management events**, for **API activity**, choose **Read** and **Write**.

[CloudTrail.2] CloudTrail should have encryption at-rest enabled

Related requirements: CIS AWS Foundations Benchmark v1.2.0/2.7, CIS AWS Foundations Benchmark v1.4.0/3.7, CIS AWS Foundations Benchmark v3.0.0/3.5, NIST.800-53.r5 AU-9, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6), NIST.800-171.r2 3.3.8, PCI DSS v3.2.1/3.4, PCI DSS v4.0.1/10.3.2

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::CloudTrail::Trail`

AWS Config rule: cloud-trail-encryption-enabled

Schedule type: Periodic

Parameters: None

This control checks whether CloudTrail is configured to use the server-side encryption (SSE) AWS KMS key encryption. The control fails if the `KmsKeyId` isn't defined.

For an added layer of security for your sensitive CloudTrail log files, you should use server-side encryption with AWS KMS keys (SSE-KMS) for your CloudTrail log files for encryption at rest. Note that by default, the log files delivered by CloudTrail to your buckets are encrypted by Amazon server-side encryption with Amazon S3-managed encryption keys (SSE-S3).

Remediation

To enable SSE-KMS encryption for CloudTrail log files, see Update a trail to use a KMS key in the *AWS CloudTrail User Guide*.

[CloudTrail.3] At least one CloudTrail trail should be enabled

Related requirements: NIST.800-171.r2 3.3.1, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7, PCI DSS v3.2.1/10.1, PCI DSS v3.2.1/10.2.1, PCI DSS v3.2.1/10.2.2, PCI DSS v3.2.1/10.2.3, PCI DSS v3.2.1/10.2.4, PCI DSS v3.2.1/10.2.5, PCI DSS v3.2.1/10.2.6, PCI DSS v3.2.1/10.2.7, PCI

DSS v3.2.1/10.3.1, PCI DSS v3.2.1/10.3.2, PCI DSS v3.2.1/10.3.3, PCI DSS v3.2.1/10.3.4, PCI DSS v3.2.1/10.3.5, PCI DSS v3.2.1/10.3.6, PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: cloudtrail-enabled

Schedule type: Periodic

Parameters: None

This control checks whether an AWS CloudTrail trail is enabled in your AWS account. The control fails if your account doesn't have at least one CloudTrail trail enabled.

However, some AWS services do not enable logging of all APIs and events. You should implement any additional audit trails other than CloudTrail and review the documentation for each service in CloudTrail Supported Services and Integrations.

Remediation

To get started with CloudTrail and create a trail, see the Getting started with AWS CloudTrail tutorial in the *AWS CloudTrail User Guide*.

[CloudTrail.4] CloudTrail log file validation should be enabled

Related requirements: CIS AWS Foundations Benchmark v1.2.0/2.2, CIS AWS Foundations Benchmark v1.4.0/3.2, CIS AWS Foundations Benchmark v3.0.0/3.2, NIST.800-53.r5 AU-9, NIST.800-53.r5 SI-4, NIST.800-53.r5 SI-7(1), NIST.800-53.r5 SI-7(3), NIST.800-53.r5 SI-7(7), NIST.800-171.r2 3.3.8, PCI DSS v3.2.1/10.5.2, PCI DSS v3.2.1/10.5.5, PCI DSS v4.0.1/10.3.2

Category: Data protection > Data integrity

Severity: Low

Resource type: `AWS::CloudTrail::Trail`

AWS Config rule: cloud-trail-log-file-validation-enabled

Schedule type: Periodic

Parameters: None

This control checks whether log file integrity validation is enabled on a CloudTrail trail.

CloudTrail log file validation creates a digitally signed digest file that contains a hash of each log that CloudTrail writes to Amazon S3. You can use these digest files to determine whether a log file was changed, deleted, or unchanged after CloudTrail delivered the log.

Security Hub recommends that you enable file validation on all trails. Log file validation provides additional integrity checks of CloudTrail logs.

Remediation

To enable CloudTrail log file validation, see Enabling log file integrity validation for CloudTrail in the *AWS CloudTrail User Guide*.

[CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs

Related requirements: PCI DSS v3.2.1/10.5.3, CIS AWS Foundations Benchmark v1.2.0/2.4, CIS AWS Foundations Benchmark v1.4.0/3.4, NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(1), NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-6(5), NIST.800-53.r5 AU-7(1), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-20, NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-4(5), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: Low

Resource type: `AWS::CloudTrail::Trail`

AWS Config rule: cloud-trail-cloud-watch-logs-enabled

Schedule type: Periodic

Parameters: None

This control checks whether CloudTrail trails are configured to send logs to CloudWatch Logs. The control fails if the `CloudWatchLogsLogGroupArn` property of the trail is empty.

CloudTrail records AWS API calls that are made in a given account. The recorded information includes the following:

- The identity of the API caller
- The time of the API call
- The source IP address of the API caller
- The request parameters
- The response elements returned by the AWS service

CloudTrail uses Amazon S3 for log file storage and delivery. You can capture CloudTrail logs in a specified S3 bucket for long-term analysis. To perform real-time analysis, you can configure CloudTrail to send logs to CloudWatch Logs.

For a trail that is enabled in all Regions in an account, CloudTrail sends log files from all of those Regions to a CloudWatch Logs log group.

Security Hub recommends that you send CloudTrail logs to CloudWatch Logs. Note that this recommendation is intended to ensure that account activity is captured, monitored, and appropriately alarmed on. You can use CloudWatch Logs to set this up with your AWS services. This recommendation does not preclude the use of a different solution.

Sending CloudTrail logs to CloudWatch Logs facilitates real-time and historic activity logging based on user, API, resource, and IP address. You can use this approach to establish alarms and notifications for anomalous or sensitivity account activity.

Remediation

To integrate CloudTrail with CloudWatch Logs, see Sending events to CloudWatch Logs in the *AWS CloudTrail User Guide*.

[CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible

Related requirements: CIS AWS Foundations Benchmark v1.2.0/2.3, CIS AWS Foundations Benchmark v1.4.0/3.3, PCI DSS v4.0.1/1.4.4

Category: Identify > Logging

Severity: Critical

Resource type: `AWS::S3::Bucket`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic and change triggered

Parameters: None

CloudTrail logs a record of every API call made in your account. These log files are stored in an S3 bucket. CIS recommends that the S3 bucket policy, or access control list (ACL), applied to the S3 bucket that CloudTrail logs to prevents public access to the CloudTrail logs. Allowing public access to CloudTrail log content might aid an adversary in identifying weaknesses in the affected account's use or configuration.

To run this check, Security Hub first uses custom logic to look for the S3 bucket where your CloudTrail logs are stored. It then uses the AWS Config managed rules to check that bucket is publicly accessible.

If you aggregate your logs into a single centralized S3 bucket, then Security Hub only runs the check against the account and Region where the centralized S3 bucket is located. For other accounts and Regions, the control status is **No data**.

If the bucket is publicly accessible, the check generates a failed finding.

Remediation

To block public access to your CloudTrail S3 bucket, see Configuring block public access settings for your S3 buckets in the *Amazon Simple Storage Service User Guide*. Select all four Amazon S3 Block Public Access Settings.

[CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket

Related requirements: CIS AWS Foundations Benchmark v1.2.0/2.6, CIS AWS Foundations Benchmark v1.4.0/3.6, CIS AWS Foundations Benchmark v3.0.0/3.4, PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Low

Resource type: `AWS::S3::Bucket`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

S3 bucket access logging generates a log that contains access records for each request made to your S3 bucket. An access log record contains details about the request, such as the request type, the resources specified in the request worked, and the time and date the request was processed.

CIS recommends that you enable bucket access logging on the CloudTrail S3 bucket.

By enabling S3 bucket logging on target S3 buckets, you can capture all events that might affect objects in a target bucket. Configuring logs to be placed in a separate bucket enables access to log information, which can be useful in security and incident response workflows.

To run this check, Security Hub first uses custom logic to look for the bucket where your CloudTrail logs are stored and then uses the AWS Config managed rule to check if logging is enabled.

If CloudTrail delivers log files from multiple AWS accounts into a single destination Amazon S3 bucket, Security Hub evaluates this control only against the destination bucket in the Region where it's located. This streamlines your findings. However, you should turn on CloudTrail in all accounts that deliver logs to the destination bucket. For all accounts except the one that holds the destination bucket, the control status is **No data**.

Remediation

To enable server access logging for your CloudTrail S3 bucket, see Enabling Amazon S3 server access logging in the *Amazon Simple Storage Service User Guide*.

[CloudTrail.9] CloudTrail trails should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::CloudTrail::Trail`

AWS Config rule: `tagged-cloudtrail-trail` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS CloudTrail trail has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the trail doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the trail isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a CloudTrail trail, see AddTags in the *AWS CloudTrail API Reference*.

[CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys

Related requirements: NIST.800-53.r5 AU-9, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SC-12(2), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::CloudTrail::EventDataStore`

AWS Config rule: event-data-store-cmk-encryption-enabled

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`kmsKeyArns`	A list of Amazon Resource Names (ARNs) of AWS KMS keys to include in the evaluation. The control generates a `FAILED` finding if an event data store isn't encrypted with a KMS key in the list.	StringList (maximum of 3 items)	1–3 ARNs of existing KMS keys. For example: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab` .	No default value

This control checks whether an AWS CloudTrail Lake event data store is encrypted at rest with a customer managed AWS KMS key. The control fails if the event data store isn't encrypted with a

customer managed KMS key. You can optionally specify a list of KMS keys for the control to include in the evaluation.

By default, AWS CloudTrail Lake encrypts event data stores with Amazon S3 managed keys (SSE-S3), using an AES-256 algorithm. For additional control, you can configure CloudTrail Lake to encrypt an event data store with a customer managed AWS KMS key (SSE-KMS) instead. A customer managed KMS key is an AWS KMS key that you create, own, and manage in your AWS account. You have full control over this type of KMS key. This includes defining and maintaining the key policy, managing grants, rotating cryptographic material, assigning tags, creating aliases, and enabling and disabling the key. You can use a customer managed KMS key in cryptographic operations for your CloudTrail data and audit usage with CloudTrail logs.

Remediation

For information about encrypting an AWS CloudTrail Lake event data store with an AWS KMS key that you specify, see Update an event data store in the *AWS CloudTrail User Guide*. After you associate an event data store with a KMS key, the KMS key can't be removed or changed.

Security Hub controls for Amazon CloudWatch

These AWS Security Hub controls evaluate the Amazon CloudWatch service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user

Related requirements: CIS AWS Foundations Benchmark v1.2.0/1.1,CIS AWS Foundations Benchmark v1.2.0/3.3, CIS AWS Foundations Benchmark v1.4.0/1.7,CIS AWS Foundations Benchmark v1.4.0/4.3, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7, PCI DSS v3.2.1/7.2.1

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

The root user has unrestricted access to all services and resources in an AWS account. We highly recommend that you avoid using the root user for daily tasks. Minimizing the use of the root user and adopting the principle of least privilege for access management reduces the risk of accidental changes and unintended disclosure of highly privileged credentials.

As a best practice, use your root user credentials only when required to perform account and service management tasks. Apply AWS Identity and Access Management (IAM) policies directly to groups and roles but not users. For a tutorial on how to set up an administrator for daily use, see Creating your first IAM admin user and group in the *IAM User Guide*

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 1.7 in the CIS AWS Foundations Benchmark v1.4.0. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only

generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{$.userIdentity.type="Root" && $.userIdentity.invokedBy NOT EXISTS && $.eventType !="AwsServiceEvent"}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *your-metric-name* is...	Greater/Equal
than...	**1**

[CloudWatch.2] Ensure a log metric filter and alarm exist for unauthorized API calls

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.1, NIST.800-171.r2 3.13.1, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for unauthorized API calls. Monitoring unauthorized API calls helps reveal application errors and might reduce time to detect malicious activity.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.1 in the CIS AWS Foundations Benchmark v1.2. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.errorCode="*UnauthorizedOperation") \|\| ($.errorCode="AccessDenied*")}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal
than...	**`1`**

[CloudWatch.3] Ensure a log metric filter and alarm exist for Management Console sign-in without MFA

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.2

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm console logins that aren't protected by MFA. Monitoring for single-factor console logins increases visibility into accounts that aren't protected by MFA.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.2 in the CIS AWS Foundations Benchmark v1.2. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{ ($.eventName = "ConsoleLogin") && ($.additionalEventData.MFAUsed != "Yes") && ($.userIdentity.type`

Field	Value
	`= "IAMUser") && ($.respon seElements.ConsoleLogin = "Success") }`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal
than...	**`1`**

[CloudWatch.4] Ensure a log metric filter and alarm exist for IAM policy changes

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.4, CIS AWS Foundations Benchmark v1.4.0/4.4, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

This control checks whether you monitor API calls in real time by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for changes made to IAM policies. Monitoring these changes helps ensure that authentication and authorization controls remain intact.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

> **Note**
>
> Our recommended filter pattern in these remediation steps differs from the filter pattern in the CIS guidance. Our recommended filters target only events coming from IAM API calls.

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventSource=iam.amazonaws.com) && (($.eventName=DeleteGroupPolicy) || ($.eventName=DeleteRolePolicy) \|\| ($.eventName=DeleteUserPolicy) || ($.eventName=PutGroupPolicy) \|\| ($.eventName=PutRolePolicy) || ($.eventName=PutUserPolicy) \|\| ($.eventName=CreatePolicy) || ($.eventName=DeletePolicy) \|\| ($.eventName=CreatePolicyVersion) || ($.eventName=DeletePolicyVersion) ||`

Field	Value
	`($.eventName=AttachRolePolicy) || ($.eventName=DetachRolePolicy) || ($.eventName=AttachUserPolicy) || ($.eventName=DetachUserPolicy) || ($.eventName=AttachGroupPolicy) || ($.eventName=DetachGroupPolicy))}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal
than...	**`1`**

[CloudWatch.5] Ensure a log metric filter and alarm exist for CloudTrail configuration changes

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.5, CIS AWS Foundations Benchmark v1.4.0/4.5, NIST.800-171.r2 3.3.8, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for changes to CloudTrail configuration settings. Monitoring these changes helps ensure sustained visibility to activities in the account.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 4.5 in the CIS AWS Foundations Benchmark v1.4.0. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls

evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventName=CreateTrail) \|\| ($.eventName=UpdateTrail) || ($.eventName=DeleteTrail) \|\| ($.eventName=StartLogging) || ($.eventName=StopLogging)}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**

Field	Value
Default value	**0**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *your-metric-name* is...	Greater/Equal
than...	**1**

[CloudWatch.6] Ensure a log metric filter and alarm exist for AWS Management Console authentication failures

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.6, CIS AWS Foundations Benchmark v1.4.0/4.6, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for failed console authentication attempts. Monitoring failed console logins might decrease lead time to detect an attempt to brute-force a credential, which might provide an indicator, such as source IP, that you can use in other event correlations.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 4.6 in the CIS AWS Foundations Benchmark v1.4.0. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventName=ConsoleLogin) && ($.errorMessage="Failed authentication")}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal
than...	**`1`**

[CloudWatch.7] Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.7, CIS AWS Foundations Benchmark v1.4.0/4.7, NIST.800-171.r2 3.13.10, NIST.800-171.r2 3.13.16, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for customer managed keys that have changed state to disabled or scheduled deletion. Data encrypted with disabled or deleted keys is no longer accessible.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 4.7 in the CIS AWS Foundations Benchmark v1.4.0. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters. The control also fails if `ExcludeManagementEventSources` contains `kms.amazonaws.com`.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.

- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.

3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventSource=kms.amazonaws.com) && (($.eventName=DisableKey) || ($.eventName=ScheduleKeyDeletion))}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal
than...	**`1`**

[CloudWatch.8] Ensure a log metric filter and alarm exist for S3 bucket policy changes

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.8, CIS AWS Foundations Benchmark v1.4.0/4.8, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for changes to S3 bucket policies. Monitoring these changes might reduce time to detect and correct permissive policies on sensitive S3 buckets.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 4.8 in the CIS AWS Foundations Benchmark v1.4.0. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only

generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventSource=s3.amazonaws.com) && (($.eventName=PutBucketAcl) || ($.eventName=PutBucketPolicy) \|\| ($.eventName=PutBucketCors) || ($.eventName=PutBucketLifecycle) \|\| ($.eventName=PutBucketReplication) || ($.eventName=DeleteBucketPolicy) \|\| ($.eventName=DeleteBucketCors) ||`

Field	Value
	`($.eventName=DeleteBucketLifecycle) \|\| ($.eventName=DeleteBucketReplication))}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal
than...	**`1`**

[CloudWatch.9] Ensure a log metric filter and alarm exist for AWS Config configuration changes

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.9, CIS AWS Foundations Benchmark v1.4.0/4.9, NIST.800-171.r2 3.3.8, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for changes to AWS Config configuration settings. Monitoring these changes helps ensure sustained visibility of configuration items in the account.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 4.9 in the CIS AWS Foundations Benchmark v1.4.0. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventSource=config.amazonaws.com) && (($.eventName=StopConfigurationRecorder) || ($.eventName=DeleteDeliveryChannel) \|\| ($.eventName=PutDeliveryChannel) || ($.eventName=PutConfigurationRecorder))}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**

Field	Value
Default value	**0**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *your-metric-name* is...	Greater/Equal
than...	**1**

[CloudWatch.10] Ensure a log metric filter and alarm exist for security group changes

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.10, CIS AWS Foundations Benchmark v1.4.0/4.10, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Security groups are a stateful packet filter that controls ingress and egress traffic in a VPC.

CIS recommends that you create a metric filter and alarm for changes to security groups. Monitoring these changes helps ensure that resources and services aren't unintentionally exposed.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 4.10 in the CIS AWS Foundations Benchmark v1.4.0. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventName=AuthorizeSecurityGroupIngress) \|\| ($.eventName=AuthorizeSecurityGroupEgress) || ($.eventName=RevokeSecurityGroupIngress) \|\| ($.eventName=RevokeSecurityGroupEgress) || ($.eventName=CreateSecurityGroup) \|\| ($.eventName=DeleteSecurityGroup)}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *your-metric-name* is...	Greater/Equal
than...	**1**

[CloudWatch.11] Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.11, CIS AWS Foundations Benchmark v1.4.0/4.11, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. NACLs are used as a stateless packet filter to control ingress and egress traffic for subnets in a VPC.

CIS recommends that you create a metric filter and alarm for changes to NACLs. Monitoring these changes helps ensure that AWS resources and services aren't unintentionally exposed.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 4.11 in the CIS AWS Foundations Benchmark v1.4.0. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventName=CreateNetworkAcl) \|\| ($.eventName=CreateNetworkAclEntry) || ($.eventName=DeleteNetworkAcl) \|\| ($.eventName=DeleteNetworkAclEntry) || ($.eventName=ReplaceNetworkAclEntry) \|\| ($.eventName=ReplaceNetworkAclAssociation)}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal

Field	Value
than...	**1**

[CloudWatch.12] Ensure a log metric filter and alarm exist for changes to network gateways

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.12, CIS AWS Foundations Benchmark v1.4.0/4.12, NIST.800-171.r2 3.3.1, NIST.800-171.r2 3.13.1

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Network gateways are required to send and receive traffic to a destination outside a VPC.

CIS recommends that you create a metric filter and alarm for changes to network gateways. Monitoring these changes helps ensure that all ingress and egress traffic traverses the VPC border via a controlled path.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 4.12 in the CIS AWS Foundations Benchmark v1.2. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

> **Note**
>
> When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.

3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventName=CreateCustomerGateway) \|\| ($.eventName=DeleteCustomerGateway) || ($.eventName=AttachInternetGateway) \|\| ($.eventName=CreateInternetGateway) || ($.eventName=DeleteInternetGateway) \|\| ($.eventName=DetachInternetGateway)}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal
than...	**`1`**

[CloudWatch.13] Ensure a log metric filter and alarm exist for route table changes

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.13, CIS AWS Foundations Benchmark v1.4.0/4.13, NIST.800-171.r2 3.3.1, NIST.800-171.r2 3.13.1, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

This control checks whether you monitor API calls in real time by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Routing tables route network traffic between subnets and to network gateways.

CIS recommends that you create a metric filter and alarm for changes to route tables. Monitoring these changes helps ensure that all VPC traffic flows through an expected path.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.

- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

Note

Our recommended filter pattern in these remediation steps differs from the filter pattern in the CIS guidance. Our recommended filters target only events coming from Amazon Elastic Compute Cloud (EC2) API calls.

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.

3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventSource=ec2.amazonaws.com) && (($.eventName=CreateRoute) || ($.eventName=CreateRouteTable) \|\| ($.eventName=ReplaceRoute) || ($.eventName=ReplaceRouteTableAssociation) \|\| ($.eventName=DeleteRouteTable) || ($.eventName=DeleteRoute) \|\| ($.eventName=DisassociateRouteTable))}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal
than...	**`1`**

[CloudWatch.14] Ensure a log metric filter and alarm exist for VPC changes

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.14, CIS AWS Foundations Benchmark v1.4.0/4.14, NIST.800-171.r2 3.3.1, NIST.800-171.r2 3.13.1, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::Logs::MetricFilter`, `AWS::CloudWatch::Alarm`, `AWS::CloudTrail::Trail`, `AWS::SNS::Topic`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. You can have more than one VPC in an account, and you can create a peer connection between two VPCs, enabling network traffic to route between VPCs.

CIS recommends that you create a metric filter and alarm for changes to VPCs. Monitoring these changes helps ensure that authentication and authorization controls remain intact.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 4.14 in the CIS AWS Foundations Benchmark v1.4.0. This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.
The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.

- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

  We recommend organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of `NO_DATA` for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

To pass this control, follow these steps to create an Amazon SNS topic, an AWS CloudTrail trail, a metric filter, and an alarm for the metric filter.

1. Create an Amazon SNS topic. For instructions, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*. Create a topic that receives all CIS alarms, and create at least one subscription to the topic.
2. Create a CloudTrail trail that applies to all AWS Regions. For instructions, see Creating a trail in the *AWS CloudTrail User Guide*.

   Make note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group in the next step.
3. Create a metric filter. For instructions, see Create a metric filter for a log group in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Define pattern, Filter pattern	`{($.eventName=CreateVpc) \|\| ($.eventName=DeleteVpc) ||`

Field	Value
	`($.eventName=ModifyVpcAttribute) \|\| ($.eventName=AcceptVpcPeeringConnection) || ($.eventName=CreateVpcPeeringConnection) \|\| ($.eventName=DeleteVpcPeeringConnection) || ($.eventName=RejectVpcPeeringConnection) \|\| ($.eventName=AttachClassicLinkVpc) || ($.eventName=DetachClassicLinkVpc) \|\| ($.eventName=DisableVpcClassicLink) || ($.eventName=EnableVpcClassicLink)}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Create a CloudWatch alarm based on a log group-metric filter in the *Amazon CloudWatch User Guide*. Use the following values:

Field	Value
Conditions, Threshold type	Static
Whenever *`your-metric-name`* is...	Greater/Equal
than...	**`1`**

[CloudWatch.15] CloudWatch alarms should have specified actions configured

Related requirements: NIST.800-53.r5 AU-6(1), NIST.800-53.r5 AU-6(5), NIST.800-53.r5 CA-7, NIST.800-53.r5 IR-4(1), NIST.800-53.r5 IR-4(5), NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-20, NIST.800-53.r5 SI-4(12), NIST.800-53.r5 SI-4(5), NIST.800-171.r2 3.3.4, NIST.800-171.r2 3.14.6

Category: Detect > Detection services

Severity: High

Resource type: `AWS::CloudWatch::Alarm`

AWS Config rule: `cloudwatch-alarm-action-check`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`alarmActionRequired`	The control produces a `PASSED` finding if the parameter is set to `true` and the alarm has an action when the alarm state changes to `ALARM`.	Boolean	Not customizable	`true`
`insufficientDataActionRequired`	The control produces a `PASSED` finding if the parameter is set to `true` and the alarm has an action when the alarm state changes to `INSUFFICIENT_DATA` .	Boolean	`true` or `false`	`false`
`okActionRequired`	The control produces a `PASSED` finding if the parameter is set to `true` and the alarm has an action when	Boolean	`true` or `false`	`false`

Parameter	Description	Type	Allowed custom values	Security Hub default value
	the alarm state changes to OK.			

This control checks whether an Amazon CloudWatch alarm has at least one action configured for the `ALARM` state. The control fails if the alarm doesn't have an action configured for the `ALARM` state. Optionally, you can include custom parameter values to also require alarm actions for the `INSUFFICIENT_DATA` or `OK` states.

> **Note**
>
> Security Hub evaluates this control based on CloudWatch metric alarms. Metric alarms may be part of composite alarms that have the specified actions configured. The control generates `FAILED` findings in the following cases:
>
> - The specified actions aren't configured for a metric alarm.
> - The metric alarm is part of a composite alarm that has the specified actions configured.

This control focuses on whether a CloudWatch alarm has an alarm action configured, whereas CloudWatch.17 focuses on the activation status of a CloudWatch alarm action.

We recommend CloudWatch alarm actions to automatically alert you when a monitored metric is outside the defined threshold. Monitoring alarms help you identify unusual activities and quickly respond to security and operational issues when an alarm goes into a specific state. The most common type of alarm action is to notify one or more users by sending a message to an Amazon Simple Notification Service (Amazon SNS) topic.

Remediation

For information about actions supported by CloudWatch alarms, see Alarm actions in the *Amazon CloudWatch User Guide*.

[CloudWatch.16] CloudWatch log groups should be retained for a specified time period

Category: Identify > Logging

Related requirements: NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-11, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-12

Severity: Medium

Resource type: `AWS::Logs::LogGroup`

AWS Config rule: `cw-loggroup-retention-period-check`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value	
`minRetentionTime`	Minimum retention period in days for CloudWatch log groups	Enum	`365, 400, 545, 731, 1827, 3653`	`365`	

This control checks whether an Amazon CloudWatch log group has a retention period of at least the specified number of days. The control fails if the retention period is less than the specified number. Unless you provide a custom parameter value for the retention period, Security Hub uses a default value of 365 days.

CloudWatch Logs centralize logs from all of your systems, applications, and AWS services in a single, highly scalable service. You can use CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud (EC2) instances, AWS CloudTrail, Amazon Route 53, and other sources. Retaining your logs for at least 1 year can help you comply with log retention standards.

Remediation

To configure log retention settings, see Change log data retention in CloudWatch Logs in the *Amazon CloudWatch User Guide*.

[CloudWatch.17] CloudWatch alarm actions should be activated

Category: Detect > Detection services

Related requirements: NIST.800-53.r5 AU-6(1), NIST.800-53.r5 AU-6(5), NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-4(12)

Severity: High

Resource type: `AWS::CloudWatch::Alarm`

AWS Config rule: `cloudwatch-alarm-action-enabled-check`

Schedule type: Change triggered

Parameters: None

This control checks whether CloudWatch alarm actions are activated (`ActionEnabled` should be set to true). The control fails if the alarm action for a CloudWatch alarm is deactivated.

> **Note**
>
> Security Hub evaluates this control based on CloudWatch metric alarms. Metric alarms may be part of composite alarms that have the alarm actions activated. The control generates `FAILED` findings in the following cases:
>
> - The specified actions aren't configured for a metric alarm.
> - The metric alarm is part of a composite alarm that has alarm actions activated.

This control focuses on the activation status of a CloudWatch alarm action, whereas CloudWatch.15 focuses on whether any `ALARM` action is configured in a CloudWatch alarm.

Alarm actions automatically alert you when a monitored metric is outside the defined threshold. If the alarm action is deactivated, no actions are run when the alarm changes state, and you won't be alerted to changes in monitored metrics. We recommend activating CloudWatch alarm actions to help you quickly respond to security and operational issues.

Remediation

To activate a CloudWatch alarm action (console)

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, under **Alarms**, choose **All alarms**.
3. Select the alarm that you want to activate actions for.
4. For **Actions**, choose **Alarm actions–new**, and then choose **Enable**.

For more information about activating CloudWatch alarm actions, see Alarm actions in the *Amazon CloudWatch User Guide*.

Security Hub controls for CodeArtifact

These Security Hub controls evaluate the AWS CodeArtifact service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[CodeArtifact.1]CodeArtifact repositories should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::CodeArtifact::Repository`

AWS Config rule: `tagged-codeartifact-repository` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource	StringList (maximum of 6 items)	1–6 tag keys that meet AWS	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
	must contain. Tag keys are case sensitive.		requirements.	

This control checks whether an AWS CodeArtifact repository has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the repository doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the repository isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a CodeArtifact repository, see Tag a repository in CodeArtifact in the *AWS CodeArtifact User Guide*.

Security Hub controls for CodeBuild

These Security Hub controls evaluate the AWS CodeBuild service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials

Related requirements: NIST.800-53.r5 SA-3, PCI DSS v3.2.1/8.2.1, PCI DSS v4.0.1/8.3.2

Category: Protect > Secure development

Severity: Critical

Resource type: `AWS::CodeBuild::Project`

AWS Config rule: `codebuild-project-source-repo-url-check`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS CodeBuild project Bitbucket source repository URL contains personal access tokens or a user name and password. The control fails if the Bitbucket source repository URL contains personal access tokens or a user name and password.

> **Note**
>
> This control evaluates both the primary source and secondary sources of a CodeBuild build project. For more information about project sources, see Multiple input sources and output artifacts sample in the *AWS CodeBuild User Guide*.

Sign-in credentials shouldn't be stored or transmitted in clear text or appear in the source repository URL. Instead of personal access tokens or sign-in credentials, you should access your source provider in CodeBuild, and change your source repository URL to contain only the path to the Bitbucket repository location. Using personal access tokens or sign-in credentials could result in unintended data exposure or unauthorized access.

Remediation

You can update your CodeBuild project to use OAuth.

To remove basic authentication / (GitHub) Personal Access Token from CodeBuild project source

1. Open the CodeBuild console at https://console.aws.amazon.com/codebuild/.
2. Choose the build project that contains personal access tokens or a user name and password.
3. From **Edit**, choose **Source**.
4. Choose **Disconnect from GitHub / Bitbucket**.
5. Choose **Connect using OAuth**, then choose **Connect to GitHub / Bitbucket**.
6. When prompted, choose **authorize as appropriate**.
7. Reconfigure your repository URL and additional configuration settings, as needed.
8. Choose **Update source**.

For more information, refer to CodeBuild use case-based samples in the *AWS CodeBuild User Guide*.

[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials

Related requirements: NIST.800-53.r5 IA-5(7), NIST.800-53.r5 SA-3, PCI DSS v3.2.1/8.2.1, PCI DSS v4.0.1/8.3.2

Category: Protect > Secure development

Severity: Critical

Resource type: `AWS::CodeBuild::Project`

AWS Config rule: `codebuild-project-envvar-awscred-check`

Schedule type: Change triggered

Parameters: None

This control checks whether the project contains the environment variables `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY`.

Authentication credentials `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY` should never be stored in clear text, as this could lead to unintended data exposure and unauthorized access.

Remediation

To remove environment variables from a CodeBuild project, see Change a build project's settings in AWS CodeBuild in the *AWS CodeBuild User Guide*. Ensure nothing is selected for **Environment variables**.

You can store environment variables with sensitive values in the AWS Systems Manager Parameter Store or AWS Secrets Manager and then retrieve them from your build spec. For instructions, see the box labeled **Important** in the Environment section in the *AWS CodeBuild User Guide*.

[CodeBuild.3] CodeBuild S3 logs should be encrypted

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SI-7(6), PCI DSS v4.0.1/10.3.2

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Low

Resource type: `AWS::CodeBuild::Project`

AWS Config rule: `codebuild-project-s3-logs-encrypted`

Schedule type: Change triggered

Parameters: None

This control checks if Amazon S3 logs for an AWS CodeBuild project are encrypted. The control fails if encryption is deactivated for S3 logs for a CodeBuild project.

Encryption of data at rest is a recommended best practice to add a layer of access management around your data. Encrypting the logs at rest reduces the risk that a user not authenticated by AWS will access the data stored on disk. It adds another set of access controls to limit the ability of unauthorized users to access the data.

Remediation

To change the encryption settings for CodeBuild project S3 logs, see Change a build project's settings in AWS CodeBuild in the *AWS CodeBuild User Guide*.

[CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration

Related requirements: NIST.800-53.r5 AC-2(12), NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-9(7), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4, NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::CodeBuild::Project`

AWS Config rule: `codebuild-project-logging-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether a CodeBuild project environment has at least one log option, either to S3 or CloudWatch logs enabled. This control fails if a CodeBuild project environment does not have at least one log option enabled.

From a security perspective, logging is an important feature to enable for future forensics efforts in the case of any security incidents. Correlating anomalies in CodeBuild projects with threat detections can increase confidence in the accuracy of those threat detections.

Remediation

For more information on how to configure CodeBuild project log settings, see Create a build project (console) in the CodeBuild User Guide.

[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled

> **Important**
>
> Security Hub retired this control in April 2024. For more information, see Change log for Security Hub CSPM controls.

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-6(10), NIST.800-53.r5 AC-6(2)

Category: Protect > Secure Access Management

Severity: High

Resource type: `AWS::CodeBuild::Project`

AWS Config rule: `codebuild-project-environment-privileged-check`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS CodeBuild project environment has privileged mode enabled or disabled. The control fails if an CodeBuild project environment has privileged mode enabled.

By default, Docker containers do not allow access to any devices. Privileged mode grants a build project's Docker container access to all devices. Setting `privilegedMode` with value `true` permits the Docker daemon to run inside a Docker container. The Docker daemon listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes. This parameter should only be set to true if the build project is used to build Docker images. Otherwise, this setting should be disabled to prevent unintended access to Docker APIs as well as the container's underlying hardware. Setting `privilegedMode` to `false` helps protect critical resources from tampering and deletion.

Remediation

To configure CodeBuild project environment settings, see Create a build project (console) in the *CodeBuild User Guide*. In the **Environment** section, don't select the **Privileged** setting.

[CodeBuild.7] CodeBuild report group exports should be encrypted at rest

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::CodeBuild::ReportGroup`

AWS Config rule: `codebuild-report-group-encrypted-at-rest`

Schedule type: Change triggered

Parameters: None

This control checks whether the test results of an AWS CodeBuild report group that are exported to an Amazon Simple Storage Service (Amazon S3) bucket are encrypted at rest. The control fails if the report group export isn't encrypted at rest.

Data at rest refers to data that's stored in persistent, non-volatile storage for any duration. Encrypting data at rest helps you protect its confidentiality, which reduces the risk that an unauthorized user can access it.

Remediation

To encrypt the report group export to S3, see Update a report group in the *AWS CodeBuild User Guide*.

Security Hub controls for Amazon CodeGuru Profiler

These Security Hub controls evaluate the Amazon CodeGuru Profiler service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::CodeGuruProfiler::ProfilingGroup`

AWS Config rule: `codeguruprofiler-profiling-group-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource	StringList (maximum of 6 items)	1–6 tag keys that meet AWS	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
	must contain. Tag keys are case sensitive.		requirements.	

This control checks whether an Amazon CodeGuru Profiler profiling group has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the profiling group doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the profiling group isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to a CodeGuru Profiler profiling group, see Tagging profiling groups in the *Amazon CodeGuru Profiler User Guide*.

Security Hub controls for Amazon CodeGuru Reviewer

These Security Hub controls evaluate the Amazon CodeGuru Reviewer service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::CodeGuruReviewer::RepositoryAssociation`

AWS Config rule: `codegurureviewer-repository-association-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon CodeGuru Reviewer repository association has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the repository association doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the repository association isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other

criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to a CodeGuru Reviewer repository association, see Tagging a repository association in the *Amazon CodeGuru Reviewer User Guide*.

Security Hub controls for Amazon Cognito

These AWS Security Hub controls evaluate the Amazon Cognito service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::Cognito::UserPool`

AWS Config rule: cognito-user-pool-advanced-security-enabled

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`SecurityMode`	The threat protection enforcement mode that the control checks for.	String	`AUDIT`, `ENFORCED`	`ENFORCED`

This control checks whether an Amazon Cognito user pool has threat protection activated with the enforcement mode set to full function for standard authentication. The control fails if the user pool has threat protection deactivated or if the enforcement mode isn't set to full function for standard authentication. Unless you provide custom parameter values, Security Hub uses the default value of `ENFORCED` for enforcement mode set to full function for standard authentication.

After you create an Amazon Cognito user pool, you can activate threat protection and customize the actions that are taken in response to different risks. Or, you can use audit mode to gather metrics on detected risks without applying any security mitigations. In audit mode, threat protection publishes metrics to Amazon CloudWatch. You can see metrics after Amazon Cognito generates its first event.

Remediation

For information about activating threat protection for an Amazon Cognito user pool, see Advanced security with threat protection in the *Amazon Cognito Developer Guide*.

[Cognito.2] Cognito identity pools should not allow unauthenticated identities

Category: Protect > Secure access management > Passwordless authentication

Severity: Medium

Resource type: `AWS::Cognito::IdentityPool`

AWS Config rule: cognito-identity-pool-unauth-access-check

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Cognito identity pool is configured to allow unauthenticated identities. The control fails if guest access is activated (the `AllowUnauthenticatedIdentities` parameter is set to `true`) for the identity pool.

If an Amazon Cognito identity pool allows unauthenticated identities, the identity pool provides temporary AWS credentials to users who haven't authenticated through an identity provider (guests). This creates security risks because it allows anonymous access to AWS resources. If you deactivate guest access, you can help ensure that only properly authenticated users can access your AWS resources, which reduces the risk of unauthorized access and potential security breaches. As a best practice, an identity pool should require authentication through supported identity providers. If unauthenticated access is necessary, it's important to carefully restrict permissions for unauthenticated identities, and regularly review and monitor their usage.

Remediation

For information about deactivating guest access for an Amazon Cognito identity pool, see Activate or deactivate guest access in the *Amazon Cognito Developer Guide*.

Security Hub controls for AWS Config

These Security Hub controls evaluate the AWS Config service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Config.1] AWS Config should be enabled and use the service-linked role for resource recording

Related requirements: CIS AWS Foundations Benchmark v1.2.0/2.5, CIS AWS Foundations Benchmark v1.4.0/3.5, CIS AWS Foundations Benchmark v3.0.0/3.3, NIST.800-53.r5 CM-3, NIST.800-53.r5 CM-6(1), NIST.800-53.r5 CM-8, NIST.800-53.r5 CM-8(2), PCI DSS v3.2.1/10.5.2, PCI DSS v3.2.1/11.5

Category: Identify > Inventory

Severity: Critical

Resource type: `AWS::::Account`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`includeConfigServiceLinkedRoleCheck`	The control doesn't evaluate whether AWS Config uses the service-linked role if the parameter is set to `false`.	Boolean	`true` or `false`	`true`

This control checks whether AWS Config is enabled in your account in the current AWS Region, records all resources that correspond to controls that are enabled in the current Region, and uses the service-linked AWS Config role. The name of the service-linked role is **AWSServiceRoleForConfig**. If you don't use the service-linked role and don't set the `includeConfigServiceLinkedRoleCheck` parameter to `false`, the control fails because other roles might not have the necessary permissions for AWS Config to accurately record your resources.

The AWS Config service performs configuration management of supported AWS resources in your account and delivers log files to you. The recorded information includes the configuration item (AWS resource), relationships between configuration items, and any configuration changes within resources. Global resources are resources that are available in any Region.

The control is evaluated as follows:

- If the current Region is set as your aggregation Region, the control produces PASSED findings only if AWS Identity and Access Management (IAM) global resources are recorded (if you have enabled controls that require them).
- If the current Region is set as a linked Region, the control doesn't evaluate whether IAM global resources are recorded.
- If the current Region isn't in your aggregator, or if cross-Region aggregation isn't set up in your account, the control produces PASSED findings only if IAM global resources are recorded (if you have enabled controls that require them).

Control results aren't impacted by whether you choose daily or continuous recording of changes in resource state in AWS Config. However, the results of this control can change when new controls are released if you have configured automatic enablement of new controls or have a central configuration policy that automatically enables new controls. In these cases, if you don't record

all resources, you must configure recording for resources that are associated with new controls in order to receive a PASSED finding.

Security Hub security checks work as intended only if you enable AWS Config in all Regions and configure resource recording for controls that require it.

> **Note**
>
> Config.1 requires that AWS Config is enabled in all Regions in which you use Security Hub. Since Security Hub is a Regional service, the check performed for this control evaluates only the current Region for the account.
> To allow security checks against IAM global resources in a Region, you must record IAM global resources in that Region. Regions that don't have IAM global resources recorded will receive a default PASSED finding for controls that check IAM global resources. Since IAM global resources are identical across AWS Regions, we recommend that you record IAM global resources in only the home Region (if cross-Region aggregation is enabled in your account). IAM resources will be recorded only in the Region in which global resource recording is turned on.
> The IAM globally recorded resource types that AWS Config supports are IAM users, groups, roles, and customer managed policies. You can consider disabling Security Hub controls that check these resource types in Regions where global resource recording is turned off. For more information, see Suggested controls to disable in Security Hub CSPM.

Remediation

In the home Region and Regions that aren't part of an aggregator, record all resources that are required for controls that are enabled in the current Region, including IAM global resources if you have enabled controls that require IAM global resources.

In linked Regions, you can use any AWS Config recording mode, as long as you are recording all resources that correspond to controls that are enabled in the current Region. In linked Regions, if you have enabled controls that require recording of IAM global resources, you won't receive a `FAILED` finding (your recording of other resources is sufficient).

The `StatusReasons` field in the `Compliance` object of your finding can help you determine why you have a failed finding for this control. For more information, see Compliance details for control findings.

For a list of which resources must be recorded for each control, see Required AWS Config resources for control findings. For general information about enabling AWS Config and configuring resource recording, see Enabling and configuring AWS Config for Security Hub CSPM.

Security Hub controls for Amazon Connect

These Security Hub controls evaluate the Amazon Connect service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Connect.1] Amazon Connect Customer Profiles object types should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::CustomerProfiles::ObjectType`

AWS Config rule: `customerprofiles-object-type-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Connect Customer Profiles object type has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the object type doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks

for the existence of a tag key and fails if the object type isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to a Customer Profiles object type, see Add tags to resources in Amazon Connect in the *Amazon Connect Administrator Guide*.

[Connect.2] Amazon Connect instances should have CloudWatch logging enabled

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::Connect::Instance`

AWS Config rule: connect-instance-logging-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Connect instance is configured to generate and store flow logs in an Amazon CloudWatch log group. The control fails if the Amazon Connect instance isn't configured to generate and store flow logs in a CloudWatch log group.

Amazon Connect flow logs provide real-time details about events in Amazon Connect flows. A *flow* defines the customer experience with an Amazon Connect contact center from start to finish. By default, when you create a new Amazon Connect instance, an Amazon CloudWatch log group is created automatically to store flow logs for the instance. Flow logs can help you analyze flows, find errors, and monitor operational metrics. You can also set up alerts for specific events that can occur in a flow.

Remediation

For information about enabling flow logs for an Amazon Connect instance, see Enable Amazon Connect flow logs in an Amazon CloudWatch log group in the *Amazon Connect Administrator Guide*.

Security Hub controls for Amazon Data Firehose

These Security Hub controls evaluate the Amazon Data Firehose service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[DataFirehose.1] Firehose delivery streams should be encrypted at rest

Related requirements: NIST.800-53.r5 AC-3, NIST.800-53.r5 AU-3, NIST.800-53.r5 SC-12, NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::KinesisFirehose::DeliveryStream`

AWS Config rule: `kinesis-firehose-delivery-stream-encrypted`

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon Data Firehose delivery stream is encrypted at rest with server-side encryption. This control fails if a Firehose delivery stream isn't encrypted at rest with server-side encryption.

Server-side encryption is a feature in Amazon Data Firehose delivery streams that automatically encrypts data before it's at rest by using a key created in AWS Key Management Service (AWS KMS). Data is encrypted before it's written to the Data Firehose stream storage layer, and decrypted after it’s retrieved from storage. This allows you to comply with regulatory requirements and enhance the security of your data.

Remediation

To enable server-side encryption on Firehose delivery streams,, see Data Protection in Amazon Data Firehose in the *Amazon Data Firehose Developer Guide*.

Security Hub controls for AWS DataSync

These Security Hub controls evaluate the AWS DataSync service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[DataSync.1] DataSync tasks should have logging enabled

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::DataSync::Task`

AWS Config rule: datasync-task-logging-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS DataSync task has logging enabled. The control fails if the task doesn't have logging enabled.

Audit logs track and monitor system activities. They provide a record of events that can help you detect security breaches, investigate incidents, and comply with regulations. Audit logs also enhance the overall accountability and transparency of your organization.

Remediation

For information about configuring logging for AWS DataSync tasks, see Monitoring data transfers with Amazon CloudWatch Logs in the *AWS DataSync User Guide*.

[DataSync.2] DataSync tasks should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::DataSync::Task`

AWS Config rule: datasync-task-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS DataSync task has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the task doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the task doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner,

environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an AWS DataSync task, see Tagging your AWS DataSync tasks in the *AWS DataSync User Guide*.

Security Hub controls for Amazon Detective

This AWS Security Hub control evaluates the Amazon Detective service and resources. The control might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Detective.1] Detective behavior graphs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Detective::Graph`

AWS Config rule: `tagged-detective-graph` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon Detective behavior graph has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the behavior graph doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the behavior graph isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Detective behavior graph, see Adding tags to a behavior graph in the *Amazon Detective Administration Guide*.

Security Hub controls for AWS DMS

These Security Hub controls evaluate the AWS Database Migration Service (AWS DMS) service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[DMS.1] Database Migration Service replication instances should not be public

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v3.2.1/1.2.1,PCI DSS v3.2.1/1.3.1,PCI DSS v3.2.1/1.3.4,PCI DSS v3.2.1/1.3.2,PCI DSS v3.2.1/1.3.6, PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration

Severity: Critical

Resource type: `AWS::DMS::ReplicationInstance`

AWS Config rule: `dms-replication-not-public`

Schedule type: Periodic

Parameters: None

This control checks whether AWS DMS replication instances are public. To do this, it examines the value of the `PubliclyAccessible` field.

A private replication instance has a private IP address that you cannot access outside of the replication network. A replication instance should have a private IP address when the source and target databases are in the same network. The network must also be connected to the replication instance's VPC using a VPN, AWS Direct Connect, or VPC peering. To learn more about public and

private replication instances, see Public and private replication instances in the *AWS Database Migration Service User Guide*.

You should also ensure that access to your AWS DMS instance configuration is limited to only authorized users. To do this, restrict users' IAM permissions to modify AWS DMS settings and resources.

Remediation

You can't change the public access setting for a DMS replication instance after creating it. To change the public access setting, delete your current instance, and then recreate it. Don't select the **Publicly accessible** option.

[DMS.2] DMS certificates should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::DMS::Certificate`

AWS Config rule: `tagged-dms-certificate` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS DMS certificate has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the certificate doesn't have any tag keys

or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the certificate isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a DMS certificate, see Tagging resources in AWS Database Migration Service in the *AWS Database Migration Service User Guide*.

[DMS.3] DMS event subscriptions should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::DMS::EventSubscription`

AWS Config rule: `tagged-dms-eventsubscription` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS DMS event subscription has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the event subscription doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the event subscription isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a DMS event subscription, see Tagging resources in AWS Database Migration Service in the *AWS Database Migration Service User Guide*.

[DMS.4] DMS replication instances should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::DMS::ReplicationInstance`

AWS Config rule: `tagged-dms-replicationinstance` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS DMS replication instance has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the replication instance doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the replication instance isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a DMS replication instance, see Tagging resources in AWS Database Migration Service in the *AWS Database Migration Service User Guide*.

[DMS.5] DMS replication subnet groups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::DMS::ReplicationSubnetGroup`

AWS Config rule: `tagged-dms-replicationsubnetgroup` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS DMS replication subnet group has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the replication subnet group doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the replication subnet group isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a DMS replication subnet group, see Tagging resources in AWS Database Migration Service in the *AWS Database Migration Service User Guide*.

[DMS.6] DMS replication instances should have automatic minor version upgrade enabled

Related requirements: NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5), PCI DSS v4.0.1/6.3.3

Category: Identify > Vulnerability, patch, and version management

Severity: Medium

Resource type: `AWS::DMS::ReplicationInstance`

AWS Config rule: dms-auto-minor-version-upgrade-check

Schedule type: Change triggered

Parameters: None

This control checks if automatic minor version upgrade is enabled for an AWS DMS replication instance. The control fails if automatic minor version upgrade isn't enabled for a DMS replication instance.

DMS provides automatic minor version upgrade to each supported replication engine so that you can keep your replication instance up-to-date. Minor versions can introduce new software features, bug fixes, security patches, and performance improvements. By enabling automatic minor version upgrade on DMS replication instances, minor upgrades are applied automatically during the maintenance window or immediately if the **Apply changes immediately option is chosen**.

Remediation

To enable automatic minor version upgrade on DMS replication instances, see Modifying a replication instance in the *AWS Database Migration Service User Guide*.

[DMS.7] DMS replication tasks for the target database should have logging enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::DMS::ReplicationTask`

AWS Config rule: dms-replication-task-targetdb-logging

Schedule type: Change triggered

Parameters: None

This control checks whether logging is enabled with the minimum severity level of `LOGGER_SEVERITY_DEFAULT` for DMS replication tasks `TARGET_APPLY` and `TARGET_LOAD`. The

control fails if logging isn't enabled for these tasks or if the minimum severity level is less than `LOGGER_SEVERITY_DEFAULT`.

DMS uses Amazon CloudWatch to log information during the migration process. Using logging task settings, you can specify which component activities are logged and how much information is logged. You should specify logging for the following tasks:

- `TARGET_APPLY` – Data and data definition language (DDL) statements are applied to the target database.
- `TARGET_LOAD` – Data is loaded into the target database.

Logging plays a critical role in DMS replication tasks by enabling monitoring, troubleshooting, auditing, performance analysis, error detection, and recovery, as well as historical analysis and reporting. It helps ensure the successful replication of data between databases while maintaining data integrity and compliance with regulatory requirements. Logging levels other than `DEFAULT` are rarely needed for these components during troubleshooting. We recommend keeping the logging level as `DEFAULT` for these components unless specifically requested to change it by Support. A minimal logging level of `DEFAULT` ensures that informational messages, warnings, and error messages are written to the logs. This control checks if the logging level is at least one of the following for the preceding replication tasks: `LOGGER_SEVERITY_DEFAULT`, `LOGGER_SEVERITY_DEBUG`, or `LOGGER_SEVERITY_DETAILED_DEBUG`.

Remediation

To enable logging for target database DMS replication tasks, see Viewing and managing AWS DMS task logs in the *AWS Database Migration Service User Guide*.

[DMS.8] DMS replication tasks for the source database should have logging enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::DMS::ReplicationTask`

AWS Config rule: dms-replication-task-sourcedb-logging

Schedule type: Change triggered

Parameters: None

This control checks whether logging is enabled with the minimum severity level of LOGGER_SEVERITY_DEFAULT for DMS replication tasks SOURCE_CAPTURE and SOURCE_UNLOAD. The control fails if logging isn't enabled for these tasks or if the minimum severity level is less than LOGGER_SEVERITY_DEFAULT.

DMS uses Amazon CloudWatch to log information during the migration process. Using logging task settings, you can specify which component activities are logged and how much information is logged. You should specify logging for the following tasks:

- SOURCE_CAPTURE – Ongoing replication or change data capture (CDC) data is captured from the source database or service, and passed to the SORTER service component.
- SOURCE_UNLOAD – Data is unloaded from the source database or service during full load.

Logging plays a critical role in DMS replication tasks by enabling monitoring, troubleshooting, auditing, performance analysis, error detection, and recovery, as well as historical analysis and reporting. It helps ensure the successful replication of data between databases while maintaining data integrity and compliance with regulatory requirements. Logging levels other than DEFAULT are rarely needed for these components during troubleshooting. We recommend keeping the logging level as DEFAULT for these components unless specifically requested to change it by Support. A minimal logging level of DEFAULT ensures that informational messages, warnings, and error messages are written to the logs. This control checks if the logging level is at least one of the following for the preceding replication tasks: LOGGER_SEVERITY_DEFAULT, LOGGER_SEVERITY_DEBUG, or LOGGER_SEVERITY_DETAILED_DEBUG.

Remediation

To enable logging for source database DMS replication tasks, see Viewing and managing AWS DMS task logs in the *AWS Database Migration Service User Guide*.

[DMS.9] DMS endpoints should use SSL

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::DMS::Endpoint`

AWS Config rule: `dms-endpoint-ssl-configured`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS DMS endpoint uses an SSL connection. The control fails if the endpoint doesn't use SSL.

SSL/TLS connections provide a layer of security by encrypting connections between DMS replication instances and your database. Using certificates provides an extra layer of security by validating that the connection is being made to the expected database. It does so by checking the server certificate that is automatically installed on all database instances that you provision. By enabling SSL connection on your DMS endpoints, you protect the confidentiality of the data during the migration.

Remediation

To add an SSL connection to a new or existing DMS endpoint, see Using SSL with AWS Database Migration Service in the *AWS Database Migration Service User Guide*.

[DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled

Related requirements: NIST.800-53.r5 AC-2, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-17, NIST.800-53.r5 IA-2, NIST.800-53.r5 IA-5, PCI DSS v4.0.1/7.3.1

Category: Protect > Secure access management > Passwordless authentication

Severity: Medium

Resource type: `AWS::DMS::Endpoint`

AWS Config rule: `dms-neptune-iam-authorization-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS DMS endpoint for an Amazon Neptune database is configured with IAM authorization. The control fails if the DMS endpoint doesn't have IAM authorization enabled.

AWS Identity and Access Management (IAM) provides fine-grained access control across AWS. With IAM, you can specify who can access which services and resources, and under which conditions. With IAM policies, you manage permissions to your workforce and systems to ensure least-privilege permissions. By enabling IAM authorization on AWS DMS endpoints for Neptune databases, you can grant authorization privileges to IAM users by using a service role specified by the `ServiceAccessRoleARN` parameter.

Remediation

To enable IAM authorization on DMS endpoints for Neptune databases, see Using Amazon Neptune as a target for AWS Database Migration Service in the *AWS Database Migration Service User Guide*.

[DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled

Related requirements: NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-6, NIST.800-53.r5 IA-2, NIST.800-53.r5 IA-5, PCI DSS v4.0.1/7.3.1

Category: Protect > Secure access management > Passwordless authentication

Severity: Medium

Resource type: `AWS::DMS::Endpoint`

AWS Config rule: `dms-mongo-db-authentication-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS DMS endpoint for MongoDB is configured with an authentication mechanism. The control fails if an authentication type isn't set for the endpoint.

AWS Database Migration Service supports two authentications methods for MongoDB —**MONGODB-CR** for MongoDB version 2.x, and **SCRAM-SHA-1** for MongoDB version 3.x or later. These authentication methods are used to authenticate and encrypt MongoDB passwords if users want to use the passwords to access the databases. Authentication on AWS DMS endpoints ensures that only authorized users can access and modify the data being migrated between databases.

Without proper authentication, unauthorized users may be able to gain access to sensitive data during the migration process. This can result in data breaches, data loss, or other security incidents.

Remediation

To enable an authentication mechanism on DMS endpoints for MongoDB, see Using MongoDB as a source for AWS DMS in the *AWS Database Migration Service User Guide*.

[DMS.12] DMS endpoints for Redis OSS should have TLS enabled

Related requirements: NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-13, PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::DMS::Endpoint`

AWS Config rule: `dms-redis-tls-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS DMS endpoint for Redis OSS is configured with a TLS connection. The control fails if the endpoint doesn't have TLS enabled.

TLS provides end-to-end security when data is sent between applications or databases over the internet. When you configure SSL encryption for your DMS endpoint, it enables encrypted communication between the source and target databases during the migration process. This helps prevent eavesdropping and interception of sensitive data by malicious actors. Without SSL encryption, sensitive data may be accessed, resulting in data breaches, data loss, or other security incidents.

Remediation

To enable a TLS connection on DMS endpoints for Redis, see Using Redis as a target for AWS Database Migration Service in the *AWS Database Migration Service User Guide*.

Security Hub controls for Amazon DocumentDB

These AWS Security Hub controls evaluate the Amazon DocumentDB (with MongoDB compatibility) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: docdb-cluster-encrypted

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon DocumentDB cluster is encrypted at rest. The control fails if an Amazon DocumentDB cluster isn't encrypted at rest.

Data at rest refers to any data that's stored in persistent, non-volatile storage for any duration. Encryption helps you protect the confidentiality of such data, reducing the risk that an unauthorized user gets access to it. Data in Amazon DocumentDB clusters should be encrypted at rest for an added layer of security. Amazon DocumentDB uses the 256-bit Advanced Encryption Standard (AES-256) to encrypt your data using encryption keys stored in AWS Key Management Service (AWS KMS).

Remediation

You can enable encryption at rest when you create an Amazon DocumentDB cluster. You can't change encryption settings after creating a cluster. For more information, see Enabling encryption at rest for an Amazon DocumentDB cluster in the *Amazon DocumentDB Developer Guide*.

[DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period

Related requirements: NIST.800-53.r5 SI-12, PCI DSS v4.0.1/3.2.1

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: docdb-cluster-backup-retention-check

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value	
`minimumBackupRetentionPeriod`	Minimum backup retention period in days	Integer	7 to 35	7	

This control checks whether an Amazon DocumentDB cluster has a backup retention period greater than or equal to the specified time frame. The control fails if the backup retention period is less than the specified time frame. Unless you provide a custom parameter value for the backup retention period, Security Hub uses a default value of 7 days.

Backups help you recover more quickly from a security incident and strengthen the resilience of your systems. By automating backups for your Amazon DocumentDB clusters, you'll be able to restore your systems to a point in time and minimize downtime and data loss. In Amazon DocumentDB, clusters have a default backup retention period of 1 day. This must be increased to a value between 7 and 35 days to pass this control.

Remediation

To change the backup retention period for your Amazon DocumentDB clusters, see Modifying an Amazon DocumentDB cluster in the *Amazon DocumentDB Developer Guide*. For **Backup**, choose the backup retention period.

[DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7,

NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration

Severity: Critical

Resource type: `AWS::RDS::DBClusterSnapshot`

AWS Config rule: docdb-cluster-snapshot-public-prohibited

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon DocumentDB manual cluster snapshot is public. The control fails if the manual cluster snapshot is public.

An Amazon DocumentDB manual cluster snapshot should not be public unless intended. If you share an unencrypted manual snapshot as public, the snapshot is available to all AWS accounts. Public snapshots may result in unintended data exposure.

> **Note**
>
> This control evaluates manual cluster snapshots. You can't share an Amazon DocumentDB automated cluster snapshot. However, you can create a manual snapshot by copying the automated snapshot, and then share the copy.

Remediation

To remove public access for Amazon DocumentDB manual cluster snapshots, see Sharing a snapshot in the *Amazon DocumentDB Developer Guide*. Programmatically, you can use the Amazon DocumentDB operation `modify-db-snapshot-attribute`. Set `attribute-name` as `restore` and `values-to-remove` as `all`.

[DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3,

NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.3.3

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: docdb-cluster-audit-logging-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon DocumentDB cluster publishes audit logs to Amazon CloudWatch Logs. The control fails if the cluster doesn't publish audit logs to CloudWatch Logs.

Amazon DocumentDB (with MongoDB compatibility) allows you to audit events that were performed in your cluster. Examples of logged events include successful and failed authentication attempts, dropping a collection in a database, or creating an index. By default, auditing is disabled in Amazon DocumentDB and requires that you take action to enable it.

Remediation

To publish Amazon DocumentDB audit logs to CloudWatch Logs, see Enabling auditing in the *Amazon DocumentDB Developer Guide*.

[DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-3, NIST.800-53.r5 SC-5(2)

Category: Protect > Data protection > Data deletion protection

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: docdb-cluster-deletion-protection-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon DocumentDB cluster has deletion protection enabled. The control fails if the cluster doesn't have deletion protection enabled.

Enabling cluster deletion protection offers an additional layer of protection against accidental database deletion or deletion by an unauthorized user. An Amazon DocumentDB cluster can't be deleted while deletion protection is enabled. You must first disable deletion protection before a delete request can succeed. Deletion protection is enabled by default when you create a cluster in the Amazon DocumentDB console.

Remediation

To enable deletion protection for an existing Amazon DocumentDB cluster, see Modifying an Amazon DocumentDB cluster in the *Amazon DocumentDB Developer Guide*. In the **Modify Cluster** section, choose **Enable** for **Deletion protection**.

[DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: docdb-cluster-encrypted-in-transit

Schedule type: Periodic

Parameters: `excludeTlsParameters`: `disabled`, `enabled` (not customizable)

This controls checks whether an Amazon DocumentDB cluster requires TLS for connections to the cluster. The control fails if the cluster parameter group associated with the cluster is not in sync, or the TLS cluster parameter is set to `disabled` or `enabled`.

You can use TLS to encrypt the connection between an application and an Amazon DocumentDB cluster. Use of TLS can help protect data from being intercepted while the data is in transit between an application and an Amazon DocumentDB cluster. Encryption in transit for an Amazon DocumentDB cluster is managed using the TLS parameter in the cluster parameter group that's associated with the cluster. When encryption in transit is enabled, secure connections using TLS are required to connect to the cluster. We recommend using the following TLS parameters: `tls1.2+`, `tls1.3+`, and `fips-140-3`.

Remediation

For information about changing the TLS settings for an Amazon DocumentDB cluster, see Encrypting data in transit in the *Amazon DocumentDB Developer Guide*.

Security Hub controls for DynamoDB

These AWS Security Hub controls evaluate the Amazon DynamoDB service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[DynamoDB.1] DynamoDB tables should automatically scale capacity with demand

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-2(2), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::DynamoDB::Table`

AWS Config rule: `dynamodb-autoscaling-enabled`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Valid custom values	Security Hub default value
`minProvisionedReadCapacity`	Minimum number of provisioned read capacity units for DynamoDB auto scaling	Integer	`1` to `40000`	No default value
`targetReadUtilization`	Target utilization percentage for read capacity	Integer	`20` to `90`	No default value

Parameter	Description	Type	Valid custom values	Security Hub default value
`minProvisionedWriteCapacity`	Minimum number of provisioned write capacity units for DynamoDB auto scaling	Integer	`1` to `40000`	No default value
`targetWriteUtilization`	Target utilization percentage for write capacity	Integer	`20` to `90`	No default value

This control checks whether an Amazon DynamoDB table can scale its read and write capacity as needed. The control fails if the table doesn't use on-demand capacity mode or provisioned mode with auto scaling configured. By default, this control only requires that one of these modes be configured, without regard to specific levels of read or write capacity. Optionally, you can provide custom parameter values to require specific levels of read and write capacity or target utilization.

Scaling capacity with demand avoids throttling exceptions, which helps to maintain availability of your applications. DynamoDB tables that use on-demand capacity mode are limited only by the DynamoDB throughput default table quotas. To raise these quotas, you can file a support ticket with Support. DynamoDB tables that use provisioned mode with auto scaling adjust the provisioned throughput capacity dynamically in response to traffic patterns. For more information about DynamoDB request throttling, see Request throttling and burst capacity in the *Amazon DynamoDB Developer Guide*.

Remediation

To enable DynamoDB automatic scaling on existing tables in capacity mode, see Enabling DynamoDB auto scaling on existing tables in the *Amazon DynamoDB Developer Guide*.

[DynamoDB.2] DynamoDB tables should have point-in-time recovery enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-12, NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: `AWS::DynamoDB::Table`

AWS Config rule: `dynamodb-pitr-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether point-in-time recovery (PITR) is enabled for an Amazon DynamoDB table.

Backups help you to recover more quickly from a security incident. They also strengthen the resilience of your systems. DynamoDB point-in-time recovery automates backups for DynamoDB tables. It reduces the time to recover from accidental delete or write operations. DynamoDB tables that have PITR enabled can be restored to any point in time in the last 35 days.

Remediation

To restore a DynamoDB table to a point in time, see Restoring a DynamoDB table to a point in time in the *Amazon DynamoDB Developer Guide*.

[DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::DAX::Cluster`

AWS Config rule: `dax-encryption-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon DynamoDB Accelerator (DAX) cluster is encrypted at rest. The control fails if the DAX cluster isn't encrypted at rest.

Encrypting data at rest reduces the risk of data stored on disk being accessed by a user not authenticated to AWS. The encryption adds another set of access controls to limit the ability of unauthorized users to access to the data. For example, API permissions are required to decrypt the data before it can be read.

Remediation

You cannot enable or disable encryption at rest after a cluster is created. You must recreate the cluster in order to enable encryption at rest. For detailed instructions on how to create a DAX cluster with encryption at rest enabled, see Enabling encryption at rest using the AWS Management Console in the *Amazon DynamoDB Developer Guide*.

[DynamoDB.4] DynamoDB tables should be present in a backup plan

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-12, NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: `AWS::DynamoDB::Table`

AWS Config rule: `dynamodb-resources-protected-by-backup-plan`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`backupVaultLockCheck`	The control produces a PASSED finding if the parameter is set to `true` and the resource uses AWS Backup Vault Lock.	Boolean	`true` or `false`	No default value

This control evaluates whether an Amazon DynamoDB table in `ACTIVE` state is covered by a backup plan. The control fails if the DynamoDB table isn't covered by a backup plan. If you set the `backupVaultLockCheck` parameter equal to `true`, the control passes only if the DynamoDB table is backed up in an AWS Backup locked vault.

AWS Backup is a fully managed backup service that helps you centralize and automate the backing up of data across AWS services. With AWS Backup, you can create backup plans that define your backup requirements, such as how frequently to back up your data and how long to retain those backups. Including DynamoDB tables in your backup plans helps you protect your data from unintended loss or deletion.

Remediation

To add a DynamoDB table to an AWS Backup backup plan, see Assigning resources to a backup plan in the *AWS Backup Developer Guide*.

[DynamoDB.5] DynamoDB tables should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::DynamoDB::Table`

AWS Config rule: `tagged-dynamodb-table` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon DynamoDB table has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the table doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the table isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a DynamoDB table, see Tagging resources in DynamoDB in the *Amazon DynamoDB Developer Guide*.

[DynamoDB.6] DynamoDB tables should have deletion protection enabled

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-3, NIST.800-53.r5 SC-5(2)

Category: Protect > Data protection > Data deletion protection

Severity: Medium

Resource type: `AWS::DynamoDB::Table`

AWS Config rule: dynamodb-table-deletion-protection-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon DynamoDB table has deletion protection enabled. The control fails if a DynamoDB table doesn't have deletion protection enabled.

You can protect a DynamoDB table from accidental deletion with the deletion protection property. Enabling this property for tables helps ensure that tables don't get accidentally deleted during regular table management operations by your administrators. This helps prevent disruption to your normal business operations.

Remediation

To enable deletion protection for a DynamoDB table, see Using deletion protection in the *Amazon DynamoDB Developer Guide*.

[DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit

Related requirements: NIST.800-53.r5 AC-17, NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::DAX::Cluster`

AWS Config rule: dax-tls-endpoint-encryption

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon DynamoDB Accelerator (DAX) cluster is encrypted in transit, with the endpoint encryption type set to TLS. The control fails if the DAX cluster isn't encrypted in transit.

HTTPS (TLS) can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. You should only allow encrypted connections over TLS to access DAX clusters. However, encrypting data in transit can affect

performance. You should test your application with encryption turned on to understand the performance profile and the impact of TLS.

Remediation

You can't change the TLS encryption setting after creating a DAX cluster. To encrypt an existing DAX cluster, create a new cluster with encryption in transit enabled, shift your application's traffic to it, and then delete the old cluster. For more information, see Using deletion protection in the *Amazon DynamoDB Developer Guide*.

Security Hub controls for Amazon EC2

These AWS Security Hub controls evaluate the Amazon Elastic Compute Cloud (Amazon EC2) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[EC2.1] Amazon EBS snapshots should not be publicly restorable

Related requirements: PCI DSS v3.2.1/1.2.1,PCI DSS v3.2.1/1.3.1,PCI DSS v3.2.1/1.3.4,PCI DSS v3.2.1/7.2.1, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration

Severity: Critical

Resource type: `AWS::::Account`

AWS Config rule: `ebs-snapshot-public-restorable-check`

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Elastic Block Store snapshots are not public. The control fails if Amazon EBS snapshots are restorable by anyone.

EBS snapshots are used to back up the data on your EBS volumes to Amazon S3 at a specific point in time. You can use the snapshots to restore previous states of EBS volumes. It is rarely acceptable to share a snapshot with the public. Typically the decision to share a snapshot publicly was made

in error or without a complete understanding of the implications. This check helps ensure that all such sharing was fully planned and intentional.

Remediation

To make a public EBS snapshot private, see Share a snapshot in the *Amazon EC2 User Guide*. For **Actions, Modify permissions**, choose **Private**.

[EC2.2] VPC default security groups should not allow inbound or outbound traffic

Related requirements: PCI DSS v3.2.1/1.2.1,PCI DSS v3.2.1/1.3.4,PCI DSS v3.2.1/2.1, CIS AWS Foundations Benchmark v1.2.0/4.3, CIS AWS Foundations Benchmark v1.4.0/5.3, CIS AWS Foundations Benchmark v3.0.0/5.4, NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(5)

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::EC2::SecurityGroup`

AWS Config rule: `vpc-default-security-group-closed`

Schedule type: Change triggered

Parameters: None

This control checks whether the default security group of a VPC allows inbound or outbound traffic. The control fails if the security group allows inbound or outbound traffic.

The rules for the default security group allow all outbound and inbound traffic from network interfaces (and their associated instances) that are assigned to the same security group. We recommend that you don't use the default security group. Because the default security group cannot be deleted, you should change the default security group rules setting to restrict inbound and outbound traffic. This prevents unintended traffic if the default security group is accidentally configured for resources such as EC2 instances.

Remediation

To remediate this issue, start by creating new least-privilege security groups. For instructions, see Create a security group in the *Amazon VPC User Guide*. Then, assign the new security groups to

your EC2 instances. For instructions, see Change an instance's security group in the *Amazon EC2 User Guide*.

After you assign the new security groups to your resources, remove all inbound and outbound rules from the default security groups. For instructions, see Configure security group rules in the *Amazon VPC User Guide*.

[EC2.3] Attached Amazon EBS volumes should be encrypted at-rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::EC2::Volume`

AWS Config rule: `encrypted-volumes`

Schedule type: Change triggered

Parameters: None

This control checks whether the EBS volumes that are in an attached state are encrypted. To pass this check, EBS volumes must be in use and encrypted. If the EBS volume is not attached, then it is not subject to this check.

For an added layer of security of your sensitive data in EBS volumes, you should enable EBS encryption at rest. Amazon EBS encryption offers a straightforward encryption solution for your EBS resources that doesn't require you to build, maintain, and secure your own key management infrastructure. It uses KMS keys when creating encrypted volumes and snapshots.

To learn more about Amazon EBS encryption, see Amazon EBS encryption in the *Amazon EC2 User Guide*.

Remediation

There's no direct way to encrypt an existing unencrypted volume or snapshot. You can only encrypt a new volume or snapshot when you create it.

If you enabled encryption by default, Amazon EBS encrypts the resulting new volume or snapshot using your default key for Amazon EBS encryption. Even if you have not enabled encryption by

default, you can enable encryption when you create an individual volume or snapshot. In both cases, you can override the default key for Amazon EBS encryption and choose a symmetric customer managed key.

For more information, see Creating an Amazon EBS volume and Copying an Amazon EBS snapshot in the *Amazon EC2 User Guide*.

[EC2.4] Stopped EC2 instances should be removed after a specified time period

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Identify > Inventory

Severity: Medium

Resource type: `AWS::EC2::Instance`

AWS Config rule: `ec2-stopped-instance`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`AllowedDays`	Number of days the EC2 instance is allowed to be in a stopped state before generating a failed finding.	Integer	1 to 365	`30`

This control checks whether an Amazon EC2 instance has been stopped for longer than the allowed number of days. The control fails if an EC2 instance is stopped for longer than the maximum allowed time period. Unless you provide a custom parameter value for the maximum allowed time period, Security Hub uses a default value of 30 days.

When an EC2 instance has not run for a significant period of time, it creates a security risk because the instance is not being actively maintained (analyzed, patched, updated). If it is later launched, the lack of proper maintenance could result in unexpected issues in your AWS environment. To

safely maintain an EC2 instance over time in an inactive state, start it periodically for maintenance and then stop it after maintenance. Ideally, this should be an automated process.

Remediation

To terminate an inactive EC2 instance, see Terminate an instance in the *Amazon EC2 User Guide*.

[EC2.6] VPC flow logging should be enabled in all VPCs

Related requirements: CIS AWS Foundations Benchmark v1.2.0/2.9, CIS AWS Foundations Benchmark v1.4.0/3.9, CIS AWS Foundations Benchmark v3.0.0/3.7, NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-7(8), NIST.800-171.r2 3.1.20, NIST.800-171.r2 3.3.1, NIST.800-171.r2 3.13.1, PCI DSS v3.2.1/10.3.3, PCI DSS v3.2.1/10.3.4, PCI DSS v3.2.1/10.3.5, PCI DSS v3.2.1/10.3.6

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::EC2::VPC`

AWS Config rule: `vpc-flow-logs-enabled`

Schedule type: Periodic

Parameters:

- `trafficType`: `REJECT` (not customizable)

This control checks whether Amazon VPC Flow Logs are found and enabled for VPCs. The traffic type is set to `Reject`. The control fails if VPC Flow Logs aren't enabled for VPCs in your account.

> **Note**
>
> This control doesn't check whether Amazon VPC Flow Logs are enabled through Amazon Security Lake for the AWS account.

With the VPC Flow Logs feature, you can capture information about the IP address traffic going to and from network interfaces in your VPC. After you create a flow log, you can view and retrieve its data in CloudWatch Logs. To reduce cost, you can also send your flow logs to Amazon S3.

Security Hub recommends that you enable flow logging for packet rejects for VPCs. Flow logs provide visibility into network traffic that traverses the VPC and can detect anomalous traffic or provide insight during security workflows.

By default, the record includes values for the different components of the IP address flow, including the source, destination, and protocol. For more information and descriptions of the log fields, see VPC Flow Logs in the *Amazon VPC User Guide*.

Remediation

To create a VPC Flow Log, see Create a Flow Log in the *Amazon VPC User Guide*. After you open the Amazon VPC console, choose **Your VPCs**. For **Filter**, choose **Reject** or **All**.

[EC2.7] EBS default encryption should be enabled

Related requirements: CIS AWS Foundations Benchmark v1.4.0/2.2.1, CIS AWS Foundations Benchmark v3.0.0/2.2.1, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `ec2-ebs-encryption-by-default`

Schedule type: Periodic

Parameters: None

This control checks whether account-level encryption is enabled by default for Amazon Elastic Block Store (Amazon EBS) volumes. The control fails if the account level encryption isn't enabled for EBS volumes.

When encryption is enabled for your account, Amazon EBS volumes and snapshot copies are encrypted at rest. This adds an additional layer of protection for your data. For more information, see Encryption by default in the *Amazon EC2 User Guide*.

Remediation

To configure default encryption for Amazon EBS volumes, see Encryption by default in the *Amazon EC2 User Guide*.

[EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)

Related requirements: CIS AWS Foundations Benchmark v3.0.0/5.6, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6, PCI DSS v4.0.1/2.2.6

Category: Protect > Network Security

Severity: High

Resource type: `AWS::EC2::Instance`

AWS Config rule: `ec2-imdsv2-check`

Schedule type: Change triggered

Parameters: None

This control checks whether your EC2 instance metadata version is configured with Instance Metadata Service Version 2 (IMDSv2). The control passes if `HttpTokens` is set to required for IMDSv2. The control fails if `HttpTokens` is set to `optional`.

You use instance metadata to configure or manage the running instance. The IMDS provides access to temporary, frequently rotated credentials. These credentials remove the need to hard code or distribute sensitive credentials to instances manually or programmatically. The IMDS is attached locally to every EC2 instance. It runs on a special "link local" IP address of 169.254.169.254. This IP address is only accessible by software that runs on the instance.

Version 2 of the IMDS adds new protections for the following types of vulnerabilities. These vulnerabilities could be used to try to access the IMDS.

- Open website application firewalls
- Open reverse proxies
- Server-side request forgery (SSRF) vulnerabilities
- Open Layer 3 firewalls and network address translation (NAT)

Security Hub recommends that you configure your EC2 instances with IMDSv2.

Remediation

To configure EC2 instances with IMDSv2, see Recommended path to requiring IMDSv2 in the *Amazon EC2 User Guide*.

[EC2.9] Amazon EC2 instances should not have a public IPv4 address

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::EC2::Instance`

AWS Config rule: `ec2-instance-no-public-ip`

Schedule type: Change triggered

Parameters: None

This control checks whether EC2 instances have a public IP address. The control fails if the `publicIp` field is present in the EC2 instance configuration item. This control applies to IPv4 addresses only.

A public IPv4 address is an IP address that is reachable from the internet. If you launch your instance with a public IP address, then your EC2 instance is reachable from the internet. A private IPv4 address is an IP address that is not reachable from the internet. You can use private IPv4 addresses for communication between EC2 instances in the same VPC or in your connected private network.

IPv6 addresses are globally unique, and therefore are reachable from the internet. However, by default all subnets have the IPv6 addressing attribute set to false. For more information about IPv6, see IP addressing in your VPC in the *Amazon VPC User Guide*.

If you have a legitimate use case to maintain EC2 instances with public IP addresses, then you can suppress the findings from this control. For more information about front-end architecture options, see the AWS Architecture Blog or the This Is My Architecture series AWS video series.

Remediation

Use a non-default VPC so that your instance isn't assigned a public IP address by default.

When you launch an EC2 instance into a default VPC, it is assigned a public IP address. When you launch an EC2 instance into a non-default VPC, the subnet configuration determines whether it

receives a public IP address. The subnet has an attribute to determine if new EC2 instances in the subnet receive a public IP address from the public IPv4 address pool.

You can disassociate an automatically-assigned public IP address from your EC2 instance. For more information, see Public IPv4 addresses and external DNS hostnames in the *Amazon EC2 User Guide*.

[EC2.10] Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-171.r2 3.1.3, NIST.800-171.r2 3.13.1

Category: Protect > Secure network configuration > API private access

Severity: Medium

Resource type: `AWS::EC2::VPC`

AWS Config rule: `service-vpc-endpoint-enabled`

Schedule type: Periodic

Parameters:

- `serviceName`: `ec2` (not customizable)

This control checks whether a service endpoint for Amazon EC2 is created for each VPC. The control fails if a VPC does not have a VPC endpoint created for the Amazon EC2 service.

This control evaluates resources in single account. It cannot describe resources that are outside of the account. Because AWS Config and Security Hub do not conduct cross-account checks, you will see `FAILED` findings for VPCs that are shared across accounts. Security Hub recommends that you suppress these `FAILED` findings.

To improve the security posture of your VPC, you can configure Amazon EC2 to use an interface VPC endpoint. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to access Amazon EC2 API operations privately. It restricts all network traffic between your VPC and Amazon EC2 to the Amazon network. Because endpoints are supported within the same

Region only, you cannot create an endpoint between a VPC and a service in a different Region. This prevents unintended Amazon EC2 API calls to other Regions.

To learn more about creating VPC endpoints for Amazon EC2, see Amazon EC2 and interface VPC endpoints in the *Amazon EC2 User Guide*.

Remediation

To create an interface endpoint to Amazon EC2 from the Amazon VPC console, see Create a VPC endpoint in the *AWS PrivateLink Guide*. For **Service name**, choose **com.amazonaws.*region*.ec2**.

You can also create and attach an endpoint policy to your VPC endpoint to control access to the Amazon EC2 API. For instructions on creating a VPC endpoint policy, see Create an endpoint policy in the *Amazon EC2 User Guide*.

[EC2.12] Unused Amazon EC2 EIPs should be removed

Related requirements: PCI DSS v3.2.1/2.4, NIST.800-53.r5 CM-8(1)

Category: Protect > Secure network configuration

Severity: Low

Resource type: `AWS::EC2::EIP`

AWS Config rule: `eip-attached`

Schedule type: Change triggered

Parameters: None

This control checks whether Elastic IP (EIP) addresses that are allocated to a VPC are attached to EC2 instances or in-use elastic network interfaces (ENIs).

A failed finding indicates you may have unused EC2 EIPs.

This will help you maintain an accurate asset inventory of EIPs in your cardholder data environment (CDE).

Remediation

To release an unused EIP, see Release an Elastic IP address in the *Amazon EC2 User Guide*.

[EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22

Related requirements: CIS AWS Foundations Benchmark v1.2.0/4.1, NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 CM-7, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(5), NIST.800-171.r2 3.1.3, NIST.800-171.r2 3.13.1, PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/2.2.2, PCI DSS v4.0.1/1.3.1

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::EC2::SecurityGroup`

AWS Config rule: `restricted-ssh`

Schedule type: Change triggered and periodic

Parameters: None

This control checks whether an Amazon EC2 security group allows ingress from 0.0.0.0/0 or ::/0 to port 22. The control fails if the security group allows ingress from 0.0.0.0/0 or ::/0 to port 22.

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. We recommend that no security group allow unrestricted ingress access to port 22. Removing unfettered connectivity to remote console services, such as SSH, reduces a server's exposure to risk.

Remediation

To prohibit ingress to port 22, remove the rule that allows such access for each security group associated with a VPC. For instructions, see Update security group rules in the *Amazon EC2 User Guide*. After selecting a security group in the Amazon EC2 console, choose **Actions, Edit inbound rules**. Remove the rule that allows access to port 22.

[EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389

Related requirements: CIS AWS Foundations Benchmark v1.2.0/4.2, PCI DSS v4.0.1/1.3.1

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::EC2::SecurityGroup`

AWS Config rule: `restricted-common-ports` (created rule is `restricted-rdp`)

Schedule type: Change triggered and periodic

Parameters: None

This control checks whether an Amazon EC2 security group allows ingress from 0.0.0.0/0 or ::/0 to port 3389. The control fails if the security group allows ingress from 0.0.0.0/0 or ::/0 to port 3389.

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. We recommend that no security group allow unrestricted ingress access to port 3389. Removing unfettered connectivity to remote console services, such as RDP, reduces a server's exposure to risk.

Remediation

To prohibit ingress to port 3389, remove the rule that allows such access for each security group associated with a VPC. For instructions, see Update security group rules in the *Amazon VPC User Guide*. After selecting a security group in the Amazon VPC Console, choose **Actions, Edit inbound rules**. Remove the rule that allows access to port 3389.

[EC2.15] Amazon EC2 subnets should not automatically assign public IP addresses

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Network Security

Severity: Medium

Resource type: `AWS::EC2::Subnet`

AWS Config rule: `subnet-auto-assign-public-ip-disabled`

Schedule type: Change triggered

Parameters: None

This control checks whether the assignment of public IPs in Amazon Virtual Private Cloud (Amazon VPC) subnets have `MapPublicIpOnLaunch` set to `FALSE`. The control passes if the flag is set to `FALSE`.

All subnets have an attribute that determines whether a network interface created in the subnet automatically receives a public IPv4 address. Instances that are launched into subnets that have this attribute enabled have a public IP address assigned to their primary network interface.

Remediation

To configure a subnet to not assign public IP addresses, see Modify the IP addressing attributes of your subnet in the *Amazon VPC User Guide*.

[EC2.16] Unused Network Access Control Lists should be removed

Related requirements: NIST.800-53.r5 CM-8(1), NIST.800-171.r2 3.4.7, PCI DSS v4.0.1/1.2.7

Category: Protect > Network Security

Severity: Low

Resource type: `AWS::EC2::NetworkAcl`

AWS Config rule: `vpc-network-acl-unused-check`

Schedule type: Change triggered

Parameters: None

This control checks whether there are any unused network access control lists (network ACLs) in your virtual private cloud (VPC). The control fails if the network ACL isn't associated with a subnet. The control doesn't generate findings for an unused default network ACL.

The control checks the item configuration of the resource `AWS::EC2::NetworkAcl` and determines the relationships of the network ACL.

If the only relationship is the VPC of the network ACL, the control fails.

If other relationships are listed, then the control passes.

Remediation

For instructions on deleting an unused network ACL, see Deleting a network ACL in the *Amazon VPC User Guide*. You can't delete the default network ACL or an ACL that is associated with subnets.

[EC2.17] Amazon EC2 instances should not use multiple ENIs

Related requirements: NIST.800-53.r5 AC-4(21)

Category: Protect > Network Security

Severity: Low

Resource type: `AWS::EC2::Instance`

AWS Config rule: `ec2-instance-multiple-eni-check`

Schedule type: Change triggered

Parameters: None

This control checks whether an EC2 instance uses multiple Elastic Network Interfaces (ENIs) or Elastic Fabric Adapters (EFAs). This control passes if a single network adapter is used. The control includes an optional parameter list to identify the allowed ENIs. This control also fails if an EC2 instance that belongs to an Amazon EKS cluster uses more than one ENI. If your EC2 instances need to have multiple ENIs as part of an Amazon EKS cluster, you can suppress those control findings.

Multiple ENIs can cause dual-homed instances, meaning instances that have multiple subnets. This can add network security complexity and introduce unintended network paths and access.

Remediation

To detach a network interface from an EC2 instance, see Detach a network interface from an instance in the *Amazon EC2 User Guide*.

[EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(5), NIST.800-171.r2 3.1.3, NIST.800-171.r2 3.1.20, NIST.800-171.r2 3.13.1

Category: Protect > Secure network configuration > Security group configuration

Severity: High

Resource type: `AWS::EC2::SecurityGroup`

AWS Config rule: `vpc-sg-open-only-to-authorized-ports`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`authorizedTcpPorts`	List of authorized TCP ports	IntegerList (minimum of 1 item and maximum of 32 items)	1 to 65535	`[80,443]`
`authorizedUdpPorts`	List of authorized UDP ports	IntegerList (minimum of 1 item and maximum of 32 items)	1 to 65535	No default value

This control checks whether an Amazon EC2 security group permits unrestricted incoming traffic from unauthorized ports. The control status is determined as follows:

- If you use the default value for `authorizedTcpPorts`, the control fails if the security group permits unrestricted incoming traffic from any port other than ports 80 and 443.
- If you provide custom values for `authorizedTcpPorts` or `authorizedUdpPorts`, the control fails if the security group permits unrestricted incoming traffic from any unlisted port.

Security groups provide stateful filtering of ingress and egress network traffic to AWS. Security group rules should follow the principal of least privileged access. Unrestricted access (IP address with a /0 suffix) increases the opportunity for malicious activity such as hacking, denial-of-service attacks, and loss of data. Unless a port is specifically allowed, the port should deny unrestricted access.

Remediation

To modify a security group, see Work with security groups in the *Amazon VPC User Guide*.

[EC2.19] Security groups should not allow unrestricted access to ports with high risk

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-7, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(5), NIST.800-171.r2 3.1.3, NIST.800-171.r2 3.1.20, NIST.800-171.r2 3.13.1

Category: Protect > Restricted network access

Severity: Critical

Resource type: `AWS::EC2::SecurityGroup`

AWS Config rule: `restricted-common-ports` (created rule is `vpc-sg-restricted-common-ports`)

Schedule type: Change triggered and periodic

Parameters: `"blockedPorts": "20,21,22,23,25,110,135,143,445,1433,1434,3000,3306,3389,4333,5000,5432,5500,560` (not customizable)

This control checks whether unrestricted incoming traffic for an Amazon EC2 security group is accessible to the specified ports that are considered to be high risk. This control fails if any of the rules in a security group allow ingress traffic from '0.0.0.0/0' or '::/0' to those ports.

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. Unrestricted access (0.0.0.0/0) increases opportunities for malicious activity, such as hacking, denial-of-service attacks, and loss of data. No security group should allow unrestricted ingress access to the following ports:

- 20, 21 (FTP)
- 22 (SSH)
- 23 (Telnet)
- 25 (SMTP)
- 110 (POP3)
- 135 (RPC)
- 143 (IMAP)
- 445 (CIFS)

- 1433, 1434 (MSSQL)
- 3000 (Go, Node.js, and Ruby web development frameworks)
- 3306 (mySQL)
- 3389 (RDP)
- 4333 (ahsp)
- 5000 (Python web development frameworks)
- 5432 (postgresql)
- 5500 (fcp-addr-srvr1)
- 5601 (OpenSearch Dashboards)
- 8080 (proxy)
- 8088 (legacy HTTP port)
- 8888 (alternative HTTP port)
- 9200 or 9300 (OpenSearch)

Remediation

To delete rules from a security group, see Delete rules from a security group in the *Amazon EC2 User Guide*.

[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5), NIST.800-171.r2 3.1.13, NIST.800-171.r2 3.1.20

Category: Recover > Resilience > High availability

Severity: Medium

Resource type:`AWS::EC2::VPNConnection`

AWS Config rule: `vpc-vpn-2-tunnels-up`

Schedule type: Change triggered

Parameters: None

A VPN tunnel is an encrypted link where data can pass from the customer network to or from AWS within an AWS Site-to-Site VPN connection. Each VPN connection includes two VPN tunnels which

you can simultaneously use for high availability. Ensuring that both VPN tunnels are up for a VPN connection is important for confirming a secure and highly available connection between an AWS VPC and your remote network.

This control checks that both VPN tunnels provided by AWS Site-to-Site VPN are in UP status. The control fails if one or both tunnels are in DOWN status.

Remediation

To modify VPN tunnel options, see Modifying Site-to-Site VPN tunnel options in the AWS Site-to-Site VPN User Guide.

[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389

Related requirements: CIS AWS Foundations Benchmark v1.4.0/5.1, CIS AWS Foundations Benchmark v3.0.0/5.1, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-7, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(5), NIST.800-171.r2 3.1.3, NIST.800-171.r2 3.1.20, NIST.800-171.r2 3.13.1, PCI DSS v4.0.1/1.3.1

Category: Protect > Secure Network Configuration

Severity: Medium

Resource type:`AWS::EC2::NetworkAcl`

AWS Config rule: `nacl-no-unrestricted-ssh-rdp`

Schedule type: Change triggered

Parameters: None

This control checks whether a network access control list (network ACL) allows unrestricted access to the default TCP ports for SSH/RDP ingress traffic. The control fails if the network ACL inbound entry allows a source CIDR block of '0.0.0.0/0' or '::/0' for TCP ports 22 or 3389. The control doesn't generate findings for a default network ACL.

Access to remote server administration ports, such as port 22 (SSH) and port 3389 (RDP), should not be publicly accessible, as this may allow unintended access to resources within your VPC.

Remediation

To edit network ACL traffic rules, see Work with network ACLs in the *Amazon VPC User Guide*.

[EC2.22] Unused Amazon EC2 security groups should be removed

Category: Identify > Inventory

Severity: Medium

Resource type: `AWS::EC2::NetworkInterface`, `AWS::EC2::SecurityGroup`

AWS Config rule: `ec2-security-group-attached-to-eni-periodic`

Schedule type: Periodic

Parameters: None

This control checks whether security groups are attached to Amazon Elastic Compute Cloud (Amazon EC2) instances or to an elastic network interface. The control fails if the security group is not associated with an Amazon EC2 instance or an elastic network interface.

> **Important**
>
> On September 20, 2023, Security Hub removed this control from the AWS Foundational Security Best Practices and NIST SP 800-53 Revision 5 standards. This control continues to be part of the AWS Control Tower service-managed standard. This control produces a passed finding if security groups are attached to EC2 instances or an elastic network interface. However, for certain use cases, unattached security groups don't pose a security risk. You can use other EC2 controls—such as EC2.2, EC2.13, EC2.14, EC2.18, and EC2.19—to monitor your security groups.

Remediation

To create, assign and delete security groups, see Security groups for your EC2 instances in the *Amazon EC2 User Guide*.

[EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests

Related requirements: NIST.800-53.r5 AC-4(21), NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Protect > Secure network configuration

Severity: High

Resource type:`AWS::EC2::TransitGateway`

AWS Config rule: `ec2-transit-gateway-auto-vpc-attach-disabled`

Schedule type: Change triggered

Parameters: None

This control checks if EC2 transit gateways are automatically accepting shared VPC attachments. This control fails for a transit gateway that automatically accepts shared VPC attachment requests.

Turning on `AutoAcceptSharedAttachments` configures a transit gateway to automatically accept any cross-account VPC attachment requests without verifying the request or the account the attachment is originating from. To follow the best practices of authorization and authentication, we recommended turning off this feature to ensure that only authorized VPC attachment requests are accepted.

Remediation

To modify a transit gateway, see Modify a transit gateway in the Amazon VPC Developer Guide.

[EC2.24] Amazon EC2 paravirtual instance types should not be used

Related requirements: NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Identify > Vulnerability, patch, and version management

Severity: Medium

Resource type:`AWS::EC2::Instance`

AWS Config rule: `ec2-paravirtual-instance-check`

Schedule type: Change triggered

Parameters: None

This control checks whether the virtualization type of an EC2 instance is paravirtual. The control fails if the `virtualizationType` of the EC2 instance is set to `paravirtual`.

Linux Amazon Machine Images (AMIs) use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). The main differences between PV and HVM AMIs are the way

in which they boot and whether they can take advantage of special hardware extensions (CPU, network, and storage) for better performance.

Historically, PV guests had better performance than HVM guests in many cases, but because of enhancements in HVM virtualization and the availability of PV drivers for HVM AMIs, this is no longer true. For more information, see Linux AMI virtualization types in the Amazon EC2 User Guide.

Remediation

To update an EC2 instance to a new instance type, see Change the instance type in the *Amazon EC2 User Guide*.

[EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type:`AWS::EC2::LaunchTemplate`

AWS Config rule: `ec2-launch-template-public-ip-disabled`

Schedule type: Change triggered

Parameters: None

This control checks if Amazon EC2 launch templates are configured to assign public IP addresses to network interfaces upon launch. The control fails if an EC2 launch template is configured to assign a public IP address to network interfaces or if there is at least one network interface that has a public IP address.

A public IP address is one that is reachable from the internet. If you configure your network interfaces with a public IP address, then the resources associated with those network interfaces may be reachable from the internet. EC2 resources shouldn't be publicly accessible because this may permit unintended access to your workloads.

Remediation

To update an EC2 launch template, see Change the default network interface settings in the *Amazon EC2 Auto Scaling User Guide*.

[EC2.28] EBS volumes should be covered by a backup plan

Category: Recover > Resilience > Backups enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-12, NIST.800-53.r5 SI-13(5)

Severity: Low

Resource type: `AWS::EC2::Volume`

AWS Config rule: `ebs-resources-protected-by-backup-plan`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`backupVaultLockCheck`	The control produces a PASSED finding if the parameter is set to `true` and the resource uses AWS Backup Vault Lock.	Boolean	`true` or `false`	No default value

This control evaluates if an Amazon EBS volume in `in-use` state is covered by a backup plan. The control fails if an EBS volume isn't covered by a backup plan. If you set the `backupVaultLockCheck` parameter equal to `true`, the control passes only if the EBS volume is backed up in an AWS Backup locked vault.

Backups help you recover more quickly from a security incident. They also strengthen the resilience of your systems. Including Amazon EBS volumes in a backup plan helps you protect your data from unintended loss or deletion.

Remediation

To add an Amazon EBS volume to an AWS Backup backup plan, see Assigning resources to a backup plan in the *AWS Backup Developer Guide*.

[EC2.33] EC2 transit gateway attachments should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::TransitGatewayAttachment`

AWS Config rule: `tagged-ec2-transitgatewayattachment` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 transit gateway attachment has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the transit gateway attachment doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the transit gateway attachment isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps

you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 transit gateway attachment, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.34] EC2 transit gateway route tables should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::TransitGatewayRouteTable`

AWS Config rule: `tagged-ec2-transitgatewayroutetable` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource	StringList (maximum of 6 items)	1–6 tag keys that meet AWS	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
	must contain. Tag keys are case sensitive.		requirements.	

This control checks whether an Amazon EC2 transit gateway route table has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the transit gateway route table doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the transit gateway route table isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 transit gateway route table, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.35] EC2 network interfaces should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::NetworkInterface`

AWS Config rule: `tagged-ec2-networkinterface` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 network interface has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the network interface doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the network interface isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 network interface, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.36] EC2 customer gateways should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::CustomerGateway`

AWS Config rule: `tagged-ec2-customergateway` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 customer gateway has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the customer gateway doesn't

have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the customer gateway isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 customer gateway, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.37] EC2 Elastic IP addresses should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::EIP`

AWS Config rule: `tagged-ec2-eip` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 Elastic IP address has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the Elastic IP address doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the Elastic IP address isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 Elastic IP address, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.38] EC2 instances should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::Instance`

AWS Config rule: `tagged-ec2-instance` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 instance has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the instance doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the instance isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 instance, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.39] EC2 internet gateways should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::InternetGateway`

AWS Config rule: `tagged-ec2-internetgateway` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 internet gateway has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the internet gateway doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the internet gateway isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 internet gateway, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.40] EC2 NAT gateways should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::NatGateway`

AWS Config rule: `tagged-ec2-natgateway` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 network address translation (NAT) gateway has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the NAT gateway doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the NAT gateway isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 NAT gateway, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.41] EC2 network ACLs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::NetworkAcl`

AWS Config rule: `tagged-ec2-networkacl` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 network access control list (network ACL) has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the network ACL doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the network ACL isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 network ACL, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.42] EC2 route tables should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::RouteTable`

AWS Config rule: `tagged-ec2-routetable` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 route table has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the route table doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the route table isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 route table, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.43] EC2 security groups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::SecurityGroup`

AWS Config rule: `tagged-ec2-securitygroup` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 security group has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the security group doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the security group isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 security group, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.44] EC2 subnets should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::Subnet`

AWS Config rule: `tagged-ec2-subnet` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 subnet has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the subnet doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the subnet isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 subnet, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.45] EC2 volumes should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::Volume`

AWS Config rule: `tagged-ec2-volume` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 volume has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the volume doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the volume isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 volume, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.46] Amazon VPCs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::VPC`

AWS Config rule: `tagged-ec2-vpc` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Virtual Private Cloud (Amazon VPC) has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the Amazon VPC doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the Amazon VPC isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a VPC, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.47] Amazon VPC endpoint services should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::VPCEndpointService`

AWS Config rule: `tagged-ec2-vpcendpointservice` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon VPC endpoint service has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the endpoint service doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the endpoint service isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals.

You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Amazon VPC endpoint service, see Manage Tags in the Configure an endpoint service section of the *AWS PrivateLink Guide*.

[EC2.48] Amazon VPC flow logs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::FlowLog`

AWS Config rule: `tagged-ec2-flowlog` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon VPC flow log has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the flow log doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the flow log isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Amazon VPC flow log, see Tag a flow log in the *Amazon VPC User Guide*.

[EC2.49] Amazon VPC peering connections should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::VPCPeeringConnection`

AWS Config rule: `tagged-ec2-vpcpeeringconnection` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon VPC peering connection has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the peering connection doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the peering connection isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Amazon VPC peering connection, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.50] EC2 VPN gateways should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::VPNGateway`

AWS Config rule: `tagged-ec2-vpngateway` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 VPN gateway has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the VPN gateway doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the VPN gateway isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other

criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 VPN gateway, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.51] EC2 Client VPN endpoints should have client connection logging enabled

Related requirements: NIST.800-53.r5 AC-2(12), NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-9(7), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4, NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), NIST.800-171.r2 3.1.12, NIST.800-171.r2 3.1.20, PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Low

Resource type: `AWS::EC2::ClientVpnEndpoint`

AWS Config rule: `ec2-client-vpn-connection-log-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS Client VPN endpoint has client connection logging enabled. The control fails if the endpoint doesn't have client connection logging enabled.

Client VPN endpoints allow remote clients to securely connect to resources in a Virtual Private Cloud (VPC) in AWS. Connection logs allow you to track user activity on the VPN endpoint and provides visibility. When you enable connection logging, you can specify the name of a log stream in the log group. If you don't specify a log stream, the Client VPN service creates one for you.

Remediation

To enable connection logging, see Enable connection logging for an existing Client VPN endpoint in the *AWS Client VPN Administrator Guide*.

[EC2.52] EC2 transit gateways should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::TransitGateway`

AWS Config rule: `tagged-ec2-transitgateway` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon EC2 transit gateway has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the transit gateway doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the

parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the transit gateway isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EC2 transit gateway, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports

Related requirements: CIS AWS Foundations Benchmark v3.0.0/5.2, PCI DSS v4.0.1/1.3.1

Category: Protect > Secure network configuration > Security group configuration

Severity: High

Resource type: `AWS::EC2::SecurityGroup`

AWS Config rule: `vpc-sg-port-restriction-check`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`ipType`	The IP version	String	Not customizable	`IPv4`
`restrictPorts`	List of ports that should reject ingress traffic	IntegerList	Not customizable	`22,3389`

This control checks whether an Amazon EC2 security group allows ingress from 0.0.0.0/0 to remote server administration ports (ports 22 and 3389). The control fails if the security group allows ingress from 0.0.0.0/0 to port 22 or 3389.

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. We recommend that no security group allow unrestricted ingress access to remote server administration ports, such as SSH to port 22 and RDP to port 3389, using either the TDP (6), UDP (17), or ALL (-1) protocols. Permitting public access to these ports increases resource attack surface and the risk of resource compromise.

Remediation

To update an EC2 security group rule to prohibit ingress traffic to the specified ports, see Update security group rules in the *Amazon EC2 User Guide*. After selecting a security group in the Amazon EC2 console, choose **Actions, Edit inbound rules**. Remove the rule that allows access to port 22 or port 3389.

[EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports

Related requirements: CIS AWS Foundations Benchmark v3.0.0/5.3, PCI DSS v4.0.1/1.3.1

Category: Protect > Secure network configuration > Security group configuration

Severity: High

Resource type: `AWS::EC2::SecurityGroup`

AWS Config rule: `vpc-sg-port-restriction-check`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`ipType`	The IP version	String	Not customizable	`IPv6`
`restrictPorts`	List of ports that should reject ingress traffic	IntegerList	Not customizable	`22,3389`

This control checks whether an Amazon EC2 security group allows ingress from ::/0 to remote server administration ports (ports 22 and 3389). The control fails if the security group allows ingress from ::/0 to port 22 or 3389.

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. We recommend that no security group allow unrestricted ingress access to remote server administration ports, such as SSH to port 22 and RDP to port 3389, using either the TDP (6), UDP (17), or ALL (-1) protocols. Permitting public access to these ports increases resource attack surface and the risk of resource compromise.

Remediation

To update an EC2 security group rule to prohibit ingress traffic to the specified ports, see Update security group rules in the *Amazon EC2 User Guide*. After selecting a security group in the Amazon EC2 console, choose **Actions, Edit inbound rules**. Remove the rule that allows access to port 22 or port 3389.

[EC2.55] VPCs should be configured with an interface endpoint for ECR API

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4)

Category: Protect > Secure access management > Access control

Severity: Medium

Resource type: `AWS::EC2::VPC`, `AWS::EC2::VPCEndpoint`

AWS Config rule: vpc-endpoint-enabled

Schedule type: Periodic

Parameters:

Parameter	Required	Description	Type	Allowed custom values	Security Hub default value
`serviceNames`	Required	The name of the service that the control evaluates	String	Not customizable	`ecr.api`
`vpcIds`	Optional	Comma-separated list of Amazon VPC IDs for VPC endpoints. If provided, the control fails if the services specified in the `serviceNa`	StringList	Customize with one or more VPC IDs	No default value

Parameter	Required	Description	Type	Allowed custom values	Security Hub default value
		me parameter don't have one of these VPC endpoints.			

This control checks whether a virtual private cloud (VPC) that you manage has an interface VPC endpoint for Amazon ECR API. The control fails if the VPC doesn't have an interface VPC endpoint for ECR API. This control evaluates resources in a single account.

AWS PrivateLink enables customers to access services hosted on AWS in a highly available and scalable manner, while keeping all the network traffic within the AWS network. Service users can privately access services powered by PrivateLink from their VPC or their on-premises, without using public IPs, and without requiring traffic to traverse across the internet.

Remediation

To configure a VPC endpoint, see Access an AWS service using an interface VPC endpoint in the *AWS PrivateLink Guide*.

[EC2.56] VPCs should be configured with an interface endpoint for Docker Registry

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4)

Category: Protect > Secure access management > Access control

Severity: Medium

Resource type: `AWS::EC2::VPC`, `AWS::EC2::VPCEndpoint`

AWS Config rule: vpc-endpoint-enabled

Schedule type: Periodic

Parameters:

Parameter	Required	Description	Type	Allowed custom values	Security Hub default value
`serviceNames`	Required	The name of the service that the control evaluates	String	Not customizable	`ecr.dkr`
`vpcIds`	Optional	Comma-separated list of Amazon VPC IDs for VPC endpoints. If provided, the control fails if the services specified in the `serviceName` parameter don't have one of these VPC endpoints.	StringList	Customize with one or more VPC IDs	No default value

This control checks whether a virtual private cloud (VPC) that you manage has an interface VPC endpoint for Docker Registry. The control fails if the VPC doesn't have an interface VPC endpoint for Docker Registry. This control evaluates resources in a single account.

AWS PrivateLink enables customers to access services hosted on AWS in a highly available and scalable manner, while keeping all the network traffic within the AWS network. Service users can privately access services powered by PrivateLink from their VPC or their on-premises, without using public IPs, and without requiring traffic to traverse across the internet.

Remediation

To configure a VPC endpoint, see Access an AWS service using an interface VPC endpoint in the *AWS PrivateLink Guide*.

[EC2.57] VPCs should be configured with an interface endpoint for Systems Manager

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4)

Category: Protect > Secure access management > Access control

Severity: Medium

Resource type: `AWS::EC2::VPC`, `AWS::EC2::VPCEndpoint`

AWS Config rule: vpc-endpoint-enabled

Schedule type: Periodic

Parameters:

Parameter	Required	Description	Type	Allowed custom values	Security Hub default value
`serviceNames`	Required	The name of the service	String	Not customizable	`ssm`

Parameter	Required	Description	Type	Allowed custom values	Security Hub default value
		that the control evaluates			
`vpcIds`	Optional	Comma-separated list of Amazon VPC IDs for VPC endpoints. If provided, the control fails if the services specified in the `serviceName` parameter don't have one of these VPC endpoints.	StringList	Customize with one or more VPC IDs	No default value

This control checks whether a virtual private cloud (VPC) that you manage has an interface VPC endpoint for AWS Systems Manager. The control fails if the VPC doesn't have an interface VPC endpoint for Systems Manager. This control evaluates resources in a single account.

AWS PrivateLink enables customers to access services hosted on AWS in a highly available and scalable manner, while keeping all the network traffic within the AWS network. Service users can

privately access services powered by PrivateLink from their VPC or their on-premises, without using public IPs, and without requiring traffic to traverse across the internet.

Remediation

To configure a VPC endpoint, see Access an AWS service using an interface VPC endpoint in the *AWS PrivateLink Guide*.

[EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4)

Category: Protect > Secure access management > Access control

Severity: Medium

Resource type: `AWS::EC2::VPC`, `AWS::EC2::VPCEndpoint`

AWS Config rule: vpc-endpoint-enabled

Schedule type: Periodic

Parameters:

Parameter	Required	Description	Type	Allowed custom values	Security Hub default value
`serviceNames`	Required	The name of the service that the control evaluates	String	Not customizable	`ssm-contacts`

Parameter	Required	Description	Type	Allowed custom values	Security Hub default value
`vpcIds`	Optional	Comma-separated list of Amazon VPC IDs for VPC endpoints. If provided, the control fails if the services specified in the `serviceName` parameter don't have one of these VPC endpoints.	StringList	Customize with one or more VPC IDs	No default value

This control checks whether a virtual private cloud (VPC) that you manage has an interface VPC endpoint for AWS Systems Manager Incident Manager Contacts. The control fails if the VPC doesn't have an interface VPC endpoint for Systems Manager Incident Manager Contacts. This control evaluates resources in a single account.

AWS PrivateLink enables customers to access services hosted on AWS in a highly available and scalable manner, while keeping all the network traffic within the AWS network. Service users can privately access services powered by PrivateLink from their VPC or their on-premises, without using public IPs, and without requiring traffic to traverse across the internet.

Remediation

To configure a VPC endpoint, see Access an AWS service using an interface VPC endpoint in the *AWS PrivateLink Guide*.

[EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4)

Category: Protect > Secure access management > Access control

Severity: Medium

Resource type: `AWS::EC2::VPC`, `AWS::EC2::VPCEndpoint`

AWS Config rule: vpc-endpoint-enabled

Schedule type: Periodic

Parameters:

Parameter	Required	Description	Type	Allowed custom values	Security Hub default value
`serviceNames`	Required	The name of the service that the control evaluates	String	Not customizable	`ssm-incidents`
`vpcIds`	Optional	Comma-separated list of Amazon	StringList	Customize with one or more VPC IDs	No default value

Parameter	Required	Description	Type	Allowed custom values	Security Hub default value
		VPC IDs for VPC endpoints. If provided, the control fails if the services specified in the `serviceName` parameter don't have one of these VPC endpoints.			

This control checks whether a virtual private cloud (VPC) that you manage has an interface VPC endpoint for AWS Systems Manager Incident Manager. The control fails if the VPC doesn't have an interface VPC endpoint for Systems Manager Incident Manager. This control evaluates resources in a single account.

AWS PrivateLink enables customers to access services hosted on AWS in a highly available and scalable manner, while keeping all the network traffic within the AWS network. Service users can privately access services powered by PrivateLink from their VPC or their on-premises, without using public IPs, and without requiring traffic to traverse across the internet.

Remediation

To configure a VPC endpoint, see Access an AWS service using an interface VPC endpoint in the *AWS PrivateLink Guide*.

[EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)

Related requirements: PCI DSS v4.0.1/2.2.6

Category: Protect > Network Security

Severity: Low

Resource type: `AWS::EC2::LaunchTemplate`

AWS Config rule: ec2-launch-template-imdsv2-check

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EC2 launch template is configured with Instance Metadata Service Version 2 (IMDSv2). The control fails if `HttpTokens` is set to `optional`.

Running resources on supported software versions ensures optimal performance, security, and access to the latest features. Regular updates safeguard against vulnerabilities, which help ensure a stable and efficient user experience.

Remediation

To require IMDSv2 on an EC2 launch template, see Configure the Instance Metadata Service options in the *Amazon EC2 User Guide*.

[EC2.171] EC2 VPN connections should have logging enabled

Related requirements: CIS AWS Foundations Benchmark v3.0.0/5.3, PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::EC2::VPNConnection`

AWS Config rule: ec2-vpn-connection-logging-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS Site-to-Site VPN connection has Amazon CloudWatch Logs enabled for both tunnels. The control fails if a Site-to-Site VPN connection doesn't have CloudWatch Logs enabled for both tunnels.

AWS Site-to-Site VPN logs provide you with deeper visibility into your Site-to-Site VPN deployments. With this feature, you have access to Site-to-Site VPN connection logs that provide details on IP Security (IPsec) tunnel establishment, Internet Key Exchange (IKE) negotiations, and dead peer detection (DPD) protocol messages. Site-to-Site VPN logs can be published to CloudWatch Logs. This feature provides customers with a single consistent way to access and analyze detailed logs for all of their Site-to-Site VPN connections.

Remediation

To enable tunnel logging on an EC2 VPN connection, see AWS Site-to-Site VPN logs in the *AWS Site-to-Site VPN User Guide*.

[EC2.172] EC2 VPC Block Public Access settings should block internet gateway traffic

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: Medium

Resource type: `AWS::EC2::VPCBlockPublicAccessOptions`

AWS Config rule: `ec2-vpc-bpa-internet-gateway-blocked` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`vpcBpaInternetGatewayBlockMode`	String value of the VPC BPA options mode.	Enum	`block-bidirectional` , `block-ingress`	No default value

This control checks whether Amazon EC2 VPC Block Public Access (BPA) settings are configured to block internet gateway traffic for all Amazon VPCs in the AWS account. The control fails if VPC BPA settings aren't configured to block internet gateway traffic. For the control to pass, the VPC BPA `InternetGatewayBlockMode` must be set to `block-bidirectional` or `block-ingress`. If the parameter `vpcBpaInternetGatewayBlockMode` is provided, the control passes only if the VPC BPA value for `InternetGatewayBlockMode` matches the parameter.

Configuring the VPC BPA settings for your account in an AWS Region lets you block resources in VPCs and subnets that you own in that Region from reaching or being reached from the internet through internet gateways and egress-only internet gateways. If you need specific VPCs and subnets to be able to reach or be reachable from the internet, you can exclude them by configuring VPC BPA exclusions. For instructions on creating and deleting exclusions, see Create and delete exclusions in the *Amazon VPC User Guide*.

Remediation

To enable bi-directional BPA at the account level, see Enable BPA bidirectional mode for your account in the *Amazon VPC User Guide*. To enable ingress-only BPA, see Change VPC BPA mode to ingress-only. To enable VPC BPA at the Organization level, see Enable VPC BPA at the Organization level.

[EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::EC2::SpotFleet`

AWS Config rule: ec2-spot-fleet-request-ct-encryption-at-rest

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EC2 Spot Fleet request that specifies launch parameters is configured to enable encryption for all Amazon Elastic Block Store (Amazon EBS) volumes attached to EC2 instances. The control fails if the Spot Fleet request specifies launch parameters and doesn't enable encryption for one or more EBS volumes specified in the request.

For an additional layer of security, you should enable encryption for Amazon EBS volumes. Encryption operations then occur on the servers that host Amazon EC2 instances, which helps ensure the security of both data at rest and data in transit between an instance and its attached EBS storage. Amazon EBS encryption is a straightforward encryption solution for EBS resources associated with your EC2 instances. With EBS encryption, you aren't required to build, maintain, and secure your own key management infrastructure. EBS encryption uses AWS KMS keys when creating encrypted volumes.

Notes

This control doesn't generate findings for Amazon EC2 Spot Fleet requests that use launch templates. It also doesn't generate findings for Spot Fleet requests that don't explicitly specify a value for the `encrypted` parameter.
On July 23, 2025, Security Hub changed the title of this control. Previously, the title of this control was: *EC2 Spot Fleet requests should enable encryption for attached EBS volumes*.
The new title more accurately reflects that the control only checks Spot Fleet requests that specify launch parameters.

Remediation

There's no direct way to encrypt an existing, unencrypted Amazon EBS volume. You can encrypt a new volume only when you create it.

However, if you enable encryption by default, Amazon EBS encrypts new volumes by using your default key for EBS encryption. If you don't enable encryption by default, you can enable encryption when you create an individual volume. In both cases, you can override the default key for EBS encryption and choose a customer managed AWS KMS key. For more information about EBS encryption, see Amazon EBS encryption in the *Amazon EBS User Guide*.

For information about creating an Amazon EC2 Spot Fleet request, see Create a Spot Fleet in the *Amazon Elastic Compute Cloud User Guide*.

[EC2.174] EC2 DHCP option sets should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::DHCPOptions`

AWS Config rule: ec2-dhcp-options-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 DHCP option set has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the option set doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the option set doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an Amazon EC2 DHCP option set, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.175] EC2 launch templates should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::LaunchTemplate`

AWS Config rule: ec2-launch-template-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 launch template has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the launch template doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the launch template doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use

tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an Amazon EC2 launch template, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.176] EC2 prefix lists should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::PrefixList`

AWS Config rule: ec2-prefix-list-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 prefix list has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the prefix list doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the prefix list doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an Amazon EC2 prefix list, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.177] EC2 traffic mirror sessions should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::TrafficMirrorSession`

AWS Config rule: ec2-traffic-mirror-session-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 traffic mirror session has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the session doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the session doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

Note

Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an Amazon EC2 traffic mirror session, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.178] EC2 traffic mirror filters should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::TrafficMirrorFilter`

AWS Config rule: ec2-traffic-mirror-filter-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 traffic mirror filter has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the filter doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the filter doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an Amazon EC2 traffic mirror filter, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.179] EC2 traffic mirror targets should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EC2::TrafficMirrorTarget`

AWS Config rule: ec2-traffic-mirror-target-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 traffic mirror target has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the target doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any

values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the target doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an Amazon EC2 traffic mirror target, see Tag your Amazon EC2 resources in the *Amazon EC2 User Guide*.

[EC2.180] EC2 network interfaces should have source/destination checking enabled

Category: Protect > Network Security

Severity: Medium

Resource type: `AWS::EC2::NetworkInterface`

AWS Config rule: ec2-enis-source-destination-check-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether source/destination checking is enabled for an Amazon EC2 elastic network interface (ENI) that's managed by users. The control fails if source/destination checking

is disabled for the user-managed ENI. This control checks only the following types of ENIs: `aws_codestar_connections_managed`, `branch`, `efa`, `interface`, `lambda`, and `quicksight`.

Source/destination checking for Amazon EC2 instances and attached ENIs should be enabled and configured consistently across your EC2 instances. Each ENI has its own setting for source/destination checks. If source/destination checking is enabled, Amazon EC2 enforces source/destination address validation, which ensures that an instance is either the source or the destination of any traffic that it receives. This provides an additional layer of network security by preventing resources from handling unintended traffic and preventing IP address spoofing.

> **Note**
>
> If you're using an EC2 instance as a NAT instance and you disabled source/destination checking for its ENI, you can use a NAT gateway instead.

Remediation

For information about enabling source/destination checks for an Amazon EC2 ENI, see Modify network interface attributes in the *Amazon EC2 User Guide*.

Security Hub controls for Auto Scaling

These Security Hub controls evaluate the Amazon EC2 Auto Scaling service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[AutoScaling.1] Auto Scaling groups associated with a load balancer should use ELB health checks

Related requirements: PCI DSS v3.2.1/2.2, NIST.800-53.r5 CA-7, NIST.800-53.r5 CP-2(2), NIST.800-53.r5 SI-2

Category: Identify > Inventory

Severity: Low

Resource type: `AWS::AutoScaling::AutoScalingGroup`

AWS Config rule: `autoscaling-group-elb-healthcheck-required`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EC2 Auto Scaling group that is associated with a load balancer uses Elastic Load Balancing (ELB) health checks. The control fails if the Auto Scaling group doesn't use ELB health checks.

ELB health checks help ensure that an Auto Scaling group can determine an instance's health based on additional tests provided by the load balancer. Using Elastic Load Balancing health checks also helps support the availability of applications that use EC2 Auto Scaling groups.

Remediation

To add Elastic Load Balancing health checks, see Add Elastic Load Balancing health checks in the *Amazon EC2 Auto Scaling User Guide*.

[AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-2(2), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::AutoScaling::AutoScalingGroup`

AWS Config rule: `autoscaling-multiple-az`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value	
`minAvailabilityZones`	Minimum number of Availability Zones	Enum	`2, 3, 4, 5, 6`	`2`	

This control checks whether an Amazon EC2 Auto Scaling group spans at least the specified number of Availability Zones (AZs). The control fails if an Auto Scaling group doesn't span at least the specified number of AZs. Unless you provide a custom parameter value for the minimum number of AZs, Security Hub uses a default value of two AZs.

An Auto Scaling group that doesn't span multiple AZs can't launch instances in another AZ to compensate if the configured single AZ becomes unavailable. However, an Auto Scaling group with a single Availability Zone may be preferred in some use cases, such as batch jobs or when inter-AZ transfer costs need to be kept to a minimum. In such cases, you can disable this control or suppress its findings.

Remediation

To add AZs to an existing Auto Scaling group, see Add and remove Availability Zones in the *Amazon EC2 Auto Scaling User Guide*.

[AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)

Related requirements: NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, PCI DSS v4.0.1/2.2.6

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::AutoScaling::LaunchConfiguration`

AWS Config rule: `autoscaling-launchconfig-requires-imdsv2`

Schedule type: Change triggered

Parameters: None

This control checks whether IMDSv2 is enabled on all instances launched by Amazon EC2 Auto Scaling groups. The control fails if the Instance Metadata Service (IMDS) version isn't included in the launch configuration or is configured as `token optional`, which is a setting that allows either IMDSv1 or IMDSv2.

IMDS provides data about your instance that you can use to configure or manage the running instance.

Version 2 of the IMDS adds new protections that weren't available in IMDSv1 to further safeguard your EC2 instances.

Remediation

An Auto Scaling group is associated with one launch configuration at a time. You cannot modify a launch configuration after you create it. To change the launch configuration for an Auto Scaling group, use an existing launch configuration as the basis for a new launch configuration with IMDSv2 enabled. For more information, see Configure instance metadata options for new instances in the *Amazon EC2 User Guide*.

[AutoScaling.4] Auto Scaling group launch configuration should not have a metadata response hop limit greater than 1

> **Important**
>
> Security Hub retired this control in April 2024. For more information, see Change log for Security Hub CSPM controls.

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::AutoScaling::LaunchConfiguration`

AWS Config rule: `autoscaling-launch-config-hop-limit`

Schedule type: Change triggered

Parameters: None

This control checks the number of network hops that a metadata token can travel. The control fails if the metadata response hop limit is greater than 1.

The Instance Metadata Service (IMDS) provides metadata information about an Amazon EC2 instance and is useful for application configuration. Restricting the HTTP PUT response for the metadata service to only the EC2 instance protects the IMDS from unauthorized use.

The Time To Live (TTL) field in the IP packet is reduced by one on every hop. This reduction can be used to ensure that the packet does not travel outside EC2. IMDSv2 protects EC2 instances that may have been misconfigured as open routers, layer 3 firewalls, VPNs, tunnels, or NAT devices, which prevents unauthorized users from retrieving metadata. With IMDSv2, the PUT response that contains the secret token cannot travel outside the instance because the default metadata response hop limit is set to 1. However, if this value is greater than 1, the token can leave the EC2 instance.

Remediation

To modify the metadata response hop limit for an existing launch configuration, see Modify instance metadata options for existing instances in the *Amazon EC2 User Guide*.

[Autoscaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::AutoScaling::LaunchConfiguration`

AWS Config rule: `autoscaling-launch-config-public-ip-disabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an Auto Scaling group's associated launch configuration assigns a public IP address to the group's instances. The control fails if the associated launch configuration assigns a public IP address.

Amazon EC2 instances in an Auto Scaling group launch configuration should not have an associated public IP address, except for in limited edge cases. Amazon EC2 instances should only be accessible from behind a load balancer instead of being directly exposed to the internet.

Remediation

An Auto Scaling group is associated with one launch configuration at a time. You cannot modify a launch configuration after you create it. To change the launch configuration for an Auto Scaling group, use an existing launch configuration as the basis for a new launch configuration. Then, update the Auto Scaling group to use the new launch configuration. For step-by-step instructions, see Change the launch configuration for an Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*. When creating the new launch configuration, under **Additional configuration**, for **Advanced details, IP address type**, choose **Do not assign a public IP address to any instances**.

After you change the launch configuration, Auto Scaling launches new instances with the new configuration options. Existing instances aren't affected. To update an existing instance, we recommend that you refresh your instance, or allow automatic scaling to gradually replace older instances with newer instances based on your termination policies. For more information about updating Auto Scaling instances, see Update Auto Scaling instances in the *Amazon EC2 Auto Scaling User Guide*.

[AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-2(2), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::AutoScaling::AutoScalingGroup`

AWS Config rule: `autoscaling-multiple-instance-types`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EC2 Auto Scaling group uses multiple instance types. The control fails if the Auto Scaling group has only one instance type defined.

You can enhance availability by deploying your application across multiple instance types running in multiple Availability Zones. Security Hub recommends using multiple instance types so that the Auto Scaling group can launch another instance type if there is insufficient instance capacity in your chosen Availability Zones.

Remediation

To create an Auto Scaling group with multiple instance types, see Auto Scaling groups with multiple instance types and purchase options in the *Amazon EC2 Auto Scaling User Guide*.

[AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Identify > Resource Configuration

Severity: Medium

Resource type: `AWS::AutoScaling::AutoScalingGroup`

AWS Config rule: `autoscaling-launch-template`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EC2 Auto Scaling group is created from an EC2 launch template. This control fails if an Amazon EC2 Auto Scaling group is not created with a launch template or if a launch template is not specified in a mixed instances policy.

An EC2 Auto Scaling group can be created from either an EC2 launch template or a launch configuration. However, using a launch template to create an Auto Scaling group ensures that you have access to the latest features and improvements.

Remediation

To create an Auto Scaling group with an EC2 launch template, see Create an Auto Scaling group using a launch template in the *Amazon EC2 Auto Scaling User Guide*. For information about how to replace a launch configuration with a launch template, see Replace a launch configuration with a launch template in the *Amazon EC2 User Guide*.

[AutoScaling.10] EC2 Auto Scaling groups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AutoScaling::AutoScalingGroup`

AWS Config rule: `tagged-autoscaling-autoscalinggroup` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EC2 Auto Scaling group has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the Auto Scaling group doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the Auto Scaling group isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Auto Scaling group, see Tag Auto Scaling groups and instances in the *Amazon EC2 Auto Scaling User Guide*.

Security Hub controls for Amazon ECR

These Security Hub controls evaluate the Amazon Elastic Container Registry (Amazon ECR) service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[ECR.1] ECR private repositories should have image scanning configured

Related requirements: NIST.800-53.r5 RA-5, PCI DSS v4.0.1/6.2.3, PCI DSS v4.0.1/6.2.4

Category: Identify > Vulnerability, patch, and version management

Severity: High

Resource type: `AWS::ECR::Repository`

AWS Config rule: ecr-private-image-scanning-enabled

Schedule type: Periodic

Parameters: None

This control checks whether a private Amazon ECR repository has image scanning configured. The control fails if the private ECR repository isn't configured for scan on push or continuous scanning.

ECR image scanning helps in identifying software vulnerabilities in your container images. Configuring image scanning on ECR repositories adds a layer of verification for the integrity and safety of the images being stored.

Remediation

To configure image scanning for an ECR repository, see Image scanning in the *Amazon Elastic Container Registry User Guide*.

[ECR.2] ECR private repositories should have tag immutability configured

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-8(1)

Category: Identify > Inventory > Tagging

Severity: Medium

Resource type: `AWS::ECR::Repository`

AWS Config rule: ecr-private-tag-immutability-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether a private ECR repository has tag immutability enabled. This control fails if a private ECR repository has tag immutability disabled. This rule passes if tag immutability is enabled and has the value `IMMUTABLE`.

Amazon ECR Tag Immutability enables customers to rely on the descriptive tags of an image as a reliable mechanism to track and uniquely identify images. An immutable tag is static, which means each tag refers to a unique image. This improves reliability and scalability as the use of a static tag will always result in the same image being deployed. When configured, tag immutability prevents the tags from being overridden, which reduces the attack surface.

Remediation

To create a repository with immutable tags configured or to update the image tag mutability settings for an existing repository, see Image tag mutability in the *Amazon Elastic Container Registry User Guide*.

[ECR.3] ECR repositories should have at least one lifecycle policy configured

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Identify > Resource configuration

Severity: Medium

Resource type: `AWS::ECR::Repository`

AWS Config rule: ecr-private-lifecycle-policy-configured

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon ECR repository has at least one lifecycle policy configured. This control fails if an ECR repository does not have any lifecycle policies configured.

Amazon ECR lifecycle policies enable you to specify the lifecycle management of images in a repository. By configuring lifecycle policies, you can automate the cleanup of unused images and the expiration of images based on age or count. Automating these tasks can help you avoid unintentionally using outdated images in your repository.

Remediation

To configure a lifecycle policy, see Creating a lifecycle policy preview in the *Amazon Elastic Container Registry User Guide*.

[ECR.4] ECR public repositories should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::ECR::PublicRepository`

AWS Config rule: `tagged-ecr-publicrepository` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon ECR public repository has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the public repository doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the public repository isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an ECR public repository, see Tagging an Amazon ECR public repository in the *Amazon Elastic Container Registry User Guide*.

[ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys

Related requirements: NIST.800-53.r5 SC-12(2), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 CA-9(1), NIST.800-53.r5 SI-7(6), NIST.800-53.r5 AU-9

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::ECR::Repository`

AWS Config rule: ecr-repository-cmk-encryption-enabled

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`kmsKeyArns`	A list of Amazon Resource Names (ARNs) of AWS KMS keys to include in the evaluation. The control generates a `FAILED` finding if an ECR repository isn't encrypted with a KMS key in the list.	StringList (maximum of 10 items)	1–10 ARNs of existing KMS keys. For example: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`	No default value

This control checks whether an Amazon ECR repository is encrypted at rest with a customer managed AWS KMS key. The control fails if the ECR repository isn't encrypted with a customer managed KMS key. You can optionally specify a list of KMS keys for the control to include in the evaluation.

By default, Amazon ECR encrypts repository data with Amazon S3 managed keys (SSE-S3), using an AES-256 algorithm. For additional control, you can configure Amazon ECR to encrypt the data with an AWS KMS key (SSE-KMS or DSSE-KMS) instead. The KMS key can be: an AWS managed key that Amazon ECR creates and manages for you and has the alias `aws/ecr`, or a customer managed key that you create and manage in your AWS account. With a customer managed KMS key, you have full control of the key. This includes defining and maintaining the key policy, managing grants, rotating cryptographic material, assigning tags, creating aliases, and enabling and disabling the key.

> **Note**
>
> AWS KMS supports cross-account access to KMS keys. If an ECR repository is encrypted with a KMS key that's owned by another account, this control doesn't perform cross-account checks when it evaluates the repository. The control doesn't assess whether Amazon ECR can access and use the key when performing cryptographic operations for the repository.

Remediation

You can't change the encryption settings for an existing ECR repository. However, you can specify different encryption settings for ECR repositories that you subsequently create. Amazon ECR supports the use of different encryption settings for individual repositories.

For more information about encryption options for ECR repositories, see Encryption at rest in the *Amazon ECR User Guide*. For more information about customer managed AWS KMS keys, see AWS KMS keys in the *AWS Key Management Service Developer Guide*.

Security Hub controls for Amazon ECS

These Security Hub controls evaluate the Amazon Elastic Container Service (Amazon ECS) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6

Category: Protect > Secure access management

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Config rule: ecs-task-definition-user-for-host-mode-check

Schedule type: Change triggered

Parameters:

- `SkipInactiveTaskDefinitions`: `true` (not customizable)

This control checks whether an active Amazon ECS task definition with host networking mode has `privileged` or `user` container definitions. The control fails for task definitions that have host network mode and container definitions of `privileged=false`, empty and `user=root`, or empty.

This control only evaluates the latest active revision of an Amazon ECS task definition.

The purpose of this control is to ensure that access is defined intentionally when you run tasks that use the host network mode. If a task definition has elevated privileges, it is because you have chosen that configuration. This control checks for unexpected privilege escalation when a task definition has host networking enabled, and you don't choose elevated privileges.

Remediation

For information about how to update a task definition, see Updating a task definition in the *Amazon Elastic Container Service Developer Guide*.

When you update a task definition, it doesn't update running tasks that were launched from the previous task definition. To update a running task, you must redeploy the task with the new task definition.

[ECS.2] ECS services should not have public IP addresses assigned to them automatically

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::ECS::Service`

AWS Config rule: `ecs-service-assign-public-ip-disabled` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon ECS services are configured to automatically assign public IP addresses. This control fails if `AssignPublicIP` is `ENABLED`. This control passes if `AssignPublicIP` is `DISABLED`.

A public IP address is an IP address that is reachable from the internet. If you launch your Amazon ECS instances with a public IP address, then your Amazon ECS instances are reachable from the internet. Amazon ECS services should not be publicly accessible, as this may allow unintended access to your container application servers.

Remediation

First, you must create a task definition for your cluster that uses the `awsvpc` network mode and specifies **FARGATE** for `requiresCompatibilities`. Then, for **Compute configuration**, choose **Launch type** and **FARGATE**. Finally, for the **Networking** field, turn off **Public IP** to disable automatic public IP assignment for your service.

[ECS.3] ECS task definitions should not share the host's process namespace

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Identify > Resource configuration

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Config rule: ecs-task-definition-pid-mode-check

Schedule type: Change triggered

Parameters: None

This control checks if Amazon ECS task definitions are configured to share a host's process namespace with its containers. The control fails if the task definition shares the host's process namespace with the containers running on it. This control only evaluates the latest active revision of an Amazon ECS task definition.

A process ID (PID) namespace provides separation between processes. It prevents system processes from being visible, and allows PIDs to be reused, including PID 1. If the host's PID namespace is

shared with containers, it would allow containers to see all of the processes on the host system. This reduces the benefit of process level isolation between the host and the containers. These circumstances could lead to unauthorized access to processes on the host itself, including the ability to manipulate and terminate them. Customers shouldn't share the host's process namespace with containers running on it.

Remediation

To configure the `pidMode` on a task definition, see Task definition parameters in the Amazon Elastic Container Service Developer Guide.

[ECS.4] ECS containers should run as non-privileged

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6

Category: Protect > Secure access management > Root user access restrictions

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Config rule: ecs-containers-nonprivileged

Schedule type: Change triggered

Parameters: None

This control checks if the `privileged` parameter in the container definition of Amazon ECS Task Definitions is set to `true`. The control fails if this parameter is equal to `true`. This control only evaluates the latest active revision of an Amazon ECS task definition.

We recommend that you remove elevated privileges from your ECS task definitions. When the privilege parameter is `true`, the container is given elevated privileges on the host container instance (similar to the root user).

Remediation

To configure the `privileged` parameter on a task definition, see Advanced container definition parameters in the Amazon Elastic Container Service Developer Guide.

[ECS.5] ECS containers should be limited to read-only access to root filesystems

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6

Category: Protect > Secure access management

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Config rule: ecs-containers-readonly-access

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon ECS container has read-only access to its root file system. The control fails if the `readonlyRootFilesystem` parameter is set to `false`, or the parameter doesn't exist in the container definition within the task definition. This control evaluates only the latest active revision of an Amazon ECS task definition.

If the `readonlyRootFilesystem` parameter is set to `true` in an Amazon ECS task definition, the ECS container is given read-only access to its root file system. This reduces security attack vectors because the container instance's root file system can't be tampered with or written to without explicit volume mounts that have read-write permissions for file system folders and directories. Enabling this option also adheres to the principle of least privilege.

Remediation

To give an Amazon ECS container read-only access to its root file system, add the `readonlyRootFilesystem` parameter to the task definition for the container, and set the value for the parameter to `true`. For information about task definition parameters and how to add them to a task definition, see Amazon ECS task definitions and Updating a task definition in the *Amazon Elastic Container Service Developer Guide*.

[ECS.8] Secrets should not be passed as container environment variables

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, PCI DSS v4.0.1/8.6.2

Category: Protect > Secure development > Credentials not hard-coded

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Config rule: ecs-no-environment-secrets

Schedule type: Change triggered

Parameters: `secretKeys`: `AWS_ACCESS_KEY_ID,AWS_SECRET_ACCESS_KEY,ECS_ENGINE_AUTH_DATA` (not customizable)

This control checks if the key value of any variables in the `environment` parameter of container definitions includes `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, or `ECS_ENGINE_AUTH_DATA`. This control fails if a single environment variable in any container definition equals `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, or `ECS_ENGINE_AUTH_DATA`. This control does not cover environmental variables passed in from other locations such as Amazon S3. This control only evaluates the latest active revision of an Amazon ECS task definition.

AWS Systems Manager Parameter Store can help you improve the security posture of your organization. We recommend using the Parameter Store to store secrets and credentials instead of directly passing them into your container instances or hard coding them into your code.

Remediation

To create parameters using SSM, see Creating Systems Manager parameters in the *AWS Systems Manager User Guide*. For more information about creating a task definition that specifies a secret, see Specifying sensitive data using Secrets Manager in the *Amazon Elastic Container Service Developer Guide*.

[ECS.9] ECS task definitions should have a logging configuration

Related requirements: NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Config rule: ecs-task-definition-log-configuration

Schedule type: Change triggered

Parameters: None

This control checks if the latest active Amazon ECS task definition has a logging configuration specified. The control fails if the task definition doesn't have the `logConfiguration` property defined or if the value for `logDriver` is null in at least one container definition.

Logging helps you maintain the reliability, availability, and performance of Amazon ECS. Collecting data from task definitions provides visibility, which can help you debug processes and find the root cause of errors. If you are using a logging solution that does not have to be defined in the ECS task definition (such as a third party logging solution), you can disable this control after ensuring that your logs are properly captured and delivered.

Remediation

To define a log configuration for your Amazon ECS task definitions, see Specifying a log configuration in your task definition in the *Amazon Elastic Container Service Developer Guide*.

[ECS.10] ECS Fargate services should run on the latest Fargate platform version

Related requirements: NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5), PCI DSS v4.0.1/6.3.3

Category: Identify > Vulnerability, patch, and version management

Severity: Medium

Resource type: `AWS::ECS::Service`

AWS Config rule: ecs-fargate-latest-platform-version

Schedule type: Change triggered

Parameters:

- `latestLinuxVersion: 1.4.0` (not customizable)
- `latestWindowsVersion: 1.0.0` (not customizable)

This control checks if Amazon ECS Fargate services are running the latest Fargate platform version. This control fails if the platform version is not the latest.

AWS Fargate platform versions refer to a specific runtime environment for Fargate task infrastructure, which is a combination of kernel and container runtime versions. New platform versions are released as the runtime environment evolves. For example, a new version may be released for kernel or operating system updates, new features, bug fixes, or security updates. Security updates and patches are deployed automatically for your Fargate tasks. If a security issue is found that affects a platform version, AWS patches the platform version.

Remediation

To update an existing service, including its platform version, see Updating a service in the *Amazon Elastic Container Service Developer Guide*.

[ECS.12] ECS clusters should use Container Insights

Related requirements: NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::ECS::Cluster`

AWS Config rule: ecs-container-insights-enabled

Schedule type: Change triggered

Parameters: None

This control checks if ECS clusters use Container Insights. This control fails if Container Insights are not set up for a cluster.

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon ECS clusters. Use CloudWatch Container Insights to collect, aggregate, and summarize metrics and logs from your containerized applications and microservices. CloudWatch automatically collects metrics for many resources, such as CPU, memory, disk, and network. Container Insights also provides diagnostic information, such as container restart failures, to help you isolate issues and resolve them quickly. You can also set CloudWatch alarms on metrics that Container Insights collects.

Remediation

To use Container Insights, see Updating a service in the *Amazon CloudWatch User Guide*.

[ECS.13] ECS services should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::ECS::Service`

AWS Config rule: `tagged-ecs-service` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon ECS service has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the service doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the service isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which

defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an ECS service, see Tagging your Amazon ECS resources in the *Amazon Elastic Container Service Developer Guide*.

[ECS.14] ECS clusters should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::ECS::Cluster`

AWS Config rule: `tagged-ecs-cluster` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon ECS cluster has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the cluster doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the cluster isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an ECS cluster, see Tagging your Amazon ECS resources in the *Amazon Elastic Container Service Developer Guide*.

[ECS.15] ECS task definitions should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::ECS::TaskDefinition`

AWS Config rule: `tagged-ecs-taskdefinition` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon ECS task definition has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the task definition doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the task definition isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an ECS task definition, see Tagging your Amazon ECS resources in the *Amazon Elastic Container Service Developer Guide*.

[ECS.16] ECS task sets should not automatically assign public IP addresses

Related requirements: PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::ECS::TaskSet`

AWS Config rule: `ecs-taskset-assign-public-ip-disabled` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon ECS task set is configured to automatically assign public IP addresses. The control fails if `AssignPublicIP` is set to `ENABLED`.

A public IP address is reachable from the internet. If you configure your task set with a public IP address, the resources associated with the task set can be reached from the internet. ECS task sets shouldn't be publicly accessible, as this may allow unintended access to your container application servers.

Remediation

To update an ECS task set so that it doesn't use a public IP address, see Updating an Amazon ECS task definition using the console in the *Amazon Elastic Container Service Developer Guide*.

[ECS.17] ECS task definitions should not use host network mode

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::ECS::TaskDefinition`

AWS Config rule: ecs-task-definition-network-mode-not-host

Schedule type: Change triggered

Parameters: None

This control checks whether the latest active revision of an Amazon ECS task definition uses `host` network mode. The control fails if the latest active revision of the ECS task definition uses `host` network mode.

When using `host` network mode, the networking of an Amazon ECS container is tied directly to the underlying host that's running the container. Consequently, this mode allows containers to connect to private loopback network services on the host and to impersonate the host. Other significant drawbacks are that there's no way to remap a container port when using `host` network mode, and you can't run more than a single instantiation of a task on each host.

Remediation

For information about networking modes and options for Amazon ECS tasks that are hosted on Amazon EC2 instances, see Amazon ECS task networking options for the EC2 launch type in the *Amazon Elastic Container Service Developer Guide*. For information about creating a new revision of a task definition and specifying a different network mode, see Updating an Amazon ECS task definition in that guide.

If the Amazon ECS task definition was created by AWS Batch, see Networking modes for AWS Batch jobs to learn about networking modes and typical usage for AWS Batch job types and to choose a secure option.

Security Hub controls for Amazon EFS

These Security Hub controls evaluate the Amazon Elastic File System (Amazon EFS) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS

Related requirements: CIS AWS Foundations Benchmark v3.0.0/2.4.1, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::EFS::FileSystem`

AWS Config rule: efs-encrypted-check

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Elastic File System is configured to encrypt the file data using AWS KMS. The check fails in the following cases.

- `Encrypted` is set to `false` in the `DescribeFileSystems` response.
- The `KmsKeyId` key in the `DescribeFileSystems` response does not match the `KmsKeyId` parameter for `efs-encrypted-check`.

Note that this control does not use the `KmsKeyId` parameter for `efs-encrypted-check`. It only checks the value of `Encrypted`.

For an added layer of security for your sensitive data in Amazon EFS, you should create encrypted file systems. Amazon EFS supports encryption for file systems at-rest. You can enable encryption of data at rest when you create an Amazon EFS file system. To learn more about Amazon EFS encryption, see Data encryption in Amazon EFS in the *Amazon Elastic File System User Guide*.

Remediation

For details on how to encrypt a new Amazon EFS file system, see Encrypting data at rest in the *Amazon Elastic File System User Guide*.

[EFS.2] Amazon EFS volumes should be in backup plans

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-12, NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > Backup

Severity: Medium

Resource type: `AWS::EFS::FileSystem`

AWS Config rule: efs-in-backup-plan

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Elastic File System (Amazon EFS) file systems are added to the backup plans in AWS Backup. The control fails if Amazon EFS file systems are not included in the backup plans.

Including EFS file systems in the backup plans helps you to protect your data from deletion and data loss.

Remediation

To enable automatic backups for an existing Amazon EFS file system, see Getting started 4: Create Amazon EFS automatic backups in the *AWS Backup Developer Guide*.

[EFS.3] EFS access points should enforce a root directory

Related requirements: NIST.800-53.r5 AC-6(10)

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::EFS::AccessPoint`

AWS Config rule: efs-access-point-enforce-root-directory

Schedule type: Change triggered

Parameters: None

This control checks if Amazon EFS access points are configured to enforce a root directory. The control fails if the value of `Path` is set to `/` (the default root directory of the file system).

When you enforce a root directory, the NFS client using the access point uses the root directory configured on the access point instead of the file system's root directory. Enforcing a root directory for an access point helps restrict data access by ensuring that users of the access point can only reach files of the specified subdirectory.

Remediation

For instructions on how to enforce a root directory for an Amazon EFS access point, see Enforcing a root directory with an access point in the *Amazon Elastic File System User Guide*.

[EFS.4] EFS access points should enforce a user identity

Related requirements: NIST.800-53.r5 AC-6(2), PCI DSS v4.0.1/7.3.1

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::EFS::AccessPoint`

AWS Config rule: efs-access-point-enforce-user-identity

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon EFS access points are configured to enforce a user identity. This control fails if a POSIX user identity is not defined while creating the EFS access point.

Amazon EFS access points are application-specific entry points into an EFS file system that make it easier to manage application access to shared datasets. Access points can enforce a user identity, including the user's POSIX groups, for all file system requests that are made through the access point. Access points can also enforce a different root directory for the file system so that clients can only access data in the specified directory or its subdirectories.

Remediation

To enforce a user identity for an Amazon EFS access point, see Enforcing a user identity using an access point in the *Amazon Elastic File System User Guide*.

[EFS.5] EFS access points should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EFS::AccessPoint`

AWS Configrule: `tagged-efs-accesspoint` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EFS access point has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the access point doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the access point isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EFS access point, see Tagging Amazon EFS resources in the *Amazon Elastic File System User Guide*.

[EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch

Category: Protect > Network security > Resources not publicly accessible

Severity: Medium

Resource type: `AWS::EFS::FileSystem`

AWS Config rule: efs-mount-target-public-accessible

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon EFS mount target is associated with subnets that assign public IP addresses on launch. The control fails if the mount target is associated with subnets that assign public IP addresses on launch.

All subnets have an attribute that determines whether a network interface created in the subnet automatically receives a public IPv4 address. Amazon EFS mount targets that are launched into subnets that have this attribute enabled have a public IP address assigned to their primary network interface.

> **Note**
>
> On August 13, 2025, Security Hub changed the title and description of this control. The new title and description more precisely reflect the scope and nature of the check that the control performs. Previously, the title of this control was: *EFS mount targets should not be associated with a public subnet*.

Remediation

To associate an existing mount target with a different subnet, you must create a new mount target in a subnet that does not assign public IP addresses on launch and then remove the old mount

target. For information about managing mount targets, see Creating and managing mount targets and security groups in the *Amazon Elastic File System User Guide*.

[EFS.7] EFS file systems should have automatic backups enabled

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: `AWS::EFS::FileSystem`

AWS Config rule: efs-automatic-backups-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EFS file system has automatic backups enabled. This control fails if the EFS file system doesn't have automatic backups enabled.

A data backup is a copy of your system, configuration, or application data that's stored separately from the original. Enabling regular backups helps you safeguard valuable data against unforeseen events like system failures, cyberattacks, or accidental deletions. Having a robust backup strategy also facilitates quicker recovery, business continuity, and peace of mind in the face of potential data loss.

Remediation

For information about using AWS Backup for EFS file systems, see Backing up EFS file systems in the *Amazon Elastic File System User Guide*.

[EFS.8] EFS file systems should be encrypted at rest

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::EFS::FileSystem`

AWS Config rule: efs-filesystem-ct-encrypted

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EFS file system encrypts data with AWS Key Management Service (AWS KMS). The control fails if a file system isn't encrypted.

Data at rest refers to data that's stored in persistent, non-volatile storage for any duration. Encrypting data at rest helps you protect its confidentiality, which reduces the risk that an unauthorized user can access it.

Remediation

To enable encryption at rest for a new EFS file system, see Encrypting data at rest in the *Amazon Elastic File System User Guide*.

Security Hub controls for Amazon EKS

These Security Hub controls evaluate the Amazon Elastic Kubernetes Service (Amazon EKS) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[EKS.1] EKS cluster endpoints should not be publicly accessible

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::EKS::Cluster`

AWS Config rule: eks-endpoint-no-public-access

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon EKS cluster endpoint is publicly accessible. The control fails if an EKS cluster has an endpoint that is publicly accessible.

When you create a new cluster, Amazon EKS creates an endpoint for the managed Kubernetes API server that you use to communicate with your cluster. By default, this API server endpoint is

publicly available to the internet. Access to the API server is secured using a combination of AWS Identity and Access Management (IAM) and native Kubernetes Role Based Access Control (RBAC). By removing public access to the endpoint, you can avoid unintentional exposure and access to your cluster.

Remediation

To modify endpoint access for an existing EKS cluster, see Modifying cluster endpoint access in the Amazon EKS User Guide. You can set up endpoint access for a new EKS cluster when creating it. For instructions on creating a new Amazon EKS cluster, see Creating an Amazon EKS cluster in the Amazon EKS User Guide.

[EKS.2] EKS clusters should run on a supported Kubernetes version

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5), PCI DSS v4.0.1/12.3.4

Category: Identify > Vulnerability, patch, and version management

Severity: High

Resource type: `AWS::EKS::Cluster`

AWS Config rule: eks-cluster-supported-version

Schedule type: Change triggered

Parameters:

- `oldestVersionSupported`: `1.31` (not customizable)

This control checks whether an Amazon Elastic Kubernetes Service (Amazon EKS) cluster runs on a supported Kubernetes version. The control fails if the EKS cluster runs on an unsupported version.

If your application doesn't require a specific version of Kubernetes, we recommend that you use the latest available Kubernetes version that's supported by EKS for your clusters. For more information, see Amazon EKS Kubernetes release calendar and Understand the Kubernetes version lifecycle on Amazon EKS in the Amazon EKS User Guide.

Remediation

To update an EKS cluster, see Update an existing cluster to a new Kubernetes version in the Amazon EKS User Guide.

[EKS.3] EKS clusters should use encrypted Kubernetes secrets

Related requirements: NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-12, NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, PCI DSS v4.0.1/8.3.2

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::EKS::Cluster`

AWS Config rule: eks-cluster-secrets-encrypted

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon EKS cluster uses encrypted Kubernetes secrets. The control fails if the cluster's Kubernetes secrets aren't encrypted.

When you encrypt secrets, you can use AWS Key Management Service (AWS KMS) keys to provide envelope encryption of Kubernetes secrets stored in etcd for your cluster. This encryption is in addition to the EBS volume encryption that is enabled by default for all data (including secrets) that is stored in etcd as part of an EKS cluster. Using secrets encryption for your EKS cluster allows you to deploy a defense in depth strategy for Kubernetes applications by encrypting Kubernetes secrets with a KMS key that you define and manage.

Remediation

To enable secrets encryption on an EKS cluster, see Enabling secret encryption on an existing cluster in the Amazon EKS User Guide.

[EKS.6] EKS clusters should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EKS::Cluster`

AWS Config rule: `tagged-eks-cluster` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EKS cluster has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the cluster doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the cluster isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EKS cluster, see Tagging your Amazon EKS resources in the Amazon EKS User Guide.

[EKS.7] EKS identity provider configurations should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::EKS::IdentityProviderConfig`

AWS Config rule: `tagged-eks-identityproviderconfig` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EKS identity provider configuration has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the configuration doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the configuration isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps

you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EKS identity provider configurations, see Tagging your Amazon EKS resources in the Amazon EKS User Guide.

[EKS.8] EKS clusters should have audit logging enabled

Related requirements: NIST.800-53.r5 AC-2(12), NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-9(7), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4, NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::EKS::Cluster`

AWS Config rule: eks-cluster-log-enabled

Schedule type: Change triggered

Parameters:

- `logTypes: audit` (not customizable)

This control checks whether an Amazon EKS cluster has audit logging enabled. The control fails if audit logging isn't enabled for the cluster.

> **Note**
>
> This control doesn't check whether Amazon EKS audit logging is enabled through Amazon Security Lake for the AWS account.

EKS control plane logging provides audit and diagnostic logs directly from the EKS control plane to Amazon CloudWatch Logs in your account. You can select the log types you need, and logs are sent as log streams to a group for each EKS cluster in CloudWatch. Logging provides visibility into the access and performance of EKS clusters. By sending EKS control plane logs for your EKS clusters to CloudWatch Logs, you can record operations for audit and diagnostic purposes in a central location.

Remediation

To enable audit logs for your EKS cluster, see Enabling and disabling control plane logs in the Amazon EKS User Guide.

Security Hub controls for ElastiCache

These AWS Security Hub controls evaluate the Amazon ElastiCache service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-12, NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > Backups enabled

Severity: High

Resource type: `AWS::ElastiCache::CacheCluster`, `AWS:ElastiCache:ReplicationGroup`

AWS Config rule: elasticache-redis-cluster-automatic-backup-check

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`snapshotRetentionPeriod`	Minimum snapshot retention period in days	Integer	1 to 35	1

This control evaluates whether an Amazon ElastiCache (Redis OSS) cluster has automatic backups scheduled. The control fails if the `SnapshotRetentionLimit` for the Redis cluster is less than the specified time period. Unless you provide a custom parameter value for the snapshot retention period, Security Hub uses a default value of 1 day.

Amazon ElastiCache (Redis OSS) clusters can back up their data. You can use the backup to restore a cluster or seed a new cluster. The backup consists of the cluster's metadata, along with all of the data in the cluster. All backups are written to Amazon Simple Storage Service (Amazon S3), which provides durable storage. You can restore your data by creating a new Redis cluster and populating it with data from a backup. You can manage backups using the AWS Management Console, the AWS Command Line Interface (AWS CLI), and the ElastiCache API.

Remediation

To schedule automatic backups on an ElastiCache (Redis OSS) cluster, see Scheduling automatic backups in the *Amazon ElastiCache User Guide*.

[ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled

Related requirements: NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5) PCI DSS v4.0.1/6.3.3

Category: Identify > Vulnerability, patch, and version management

Severity: High

Resource type: `AWS::ElastiCache::CacheCluster`

AWS Config rule: `elasticache-auto-minor-version-upgrade-check`

Schedule type: Periodic

Parameters: None

This control evaluates whether Amazon ElastiCache automatically applies minor version upgrades to a cache cluster. The control fails if the cache cluster doesn't have minor version upgrades automatically applied.

> **Note**
>
> This control doesn't apply to ElastiCache Memcached clusters.

Automatic minor version upgrade is a feature that you can enable in Amazon ElastiCache to automatically upgrade your cache clusters when a new minor cache engine version is available. These upgrades might include security patches and bug fixes. Staying up-to-date with patch installation is an important step in securing systems.

Remediation

To automatically apply minor version upgrades to an existing ElastiCache cache cluster, see Version management for ElastiCache in the *Amazon ElastiCache User Guide*.

[ElastiCache.3] ElastiCache replication groups should have automatic failover enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::ElastiCache::ReplicationGroup`

AWS Config rule: `elasticache-repl-grp-auto-failover-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether an ElastiCache replication groups has automatic failover enabled. The control fails if automatic failover isn't enabled for a replication group.

When automatic failover is enabled for a replication group, the role of primary node will automatically fail over to one of the read replicas. This failover and replica promotion ensure that you can resume writing to the new primary after promotion is complete, which reduces overall downtime in case of failure.

Remediation

To enable automatic failover for an existing ElastiCache replication group,, see Modifying an ElastiCache cluster in the *Amazon ElastiCache User Guide*. If you use the ElastiCache console, set **Auto failover** to enabled.

[ElastiCache.4] ElastiCache replication groups should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::ElastiCache::ReplicationGroup`

AWS Config rule: `elasticache-repl-grp-encrypted-at-rest`

Schedule type: Periodic

Parameters: None

This control checks whether an ElastiCache replication group is encrypted at rest. The control fails if the replication group isn't encrypted at rest.

Encrypting data at rest reduces the risk that an unauthenticated user gets access to data that is stored on disk. ElastiCache (Redis OSS) replication groups should be encrypted at rest for an added layer of security.

Remediation

To configure at-rest encryption on an ElastiCache replication group, see Enabling at-rest encryption in the *Amazon ElastiCache User Guide*.

[ElastiCache.5] ElastiCache replication groups should be encrypted in transit

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::ElastiCache::ReplicationGroup`

AWS Config rule: `elasticache-repl-grp-encrypted-in-transit`

Schedule type: Periodic

Parameters: None

This control checks whether an ElastiCache replication group is encrypted in transit. The control fails if the replication group isn't encrypted in transit.

Encrypting data in transit reduces the risk that an unauthorized user can eavesdrop on network traffic. Enabling encryption in transit on an ElastiCache replication group encrypts your data whenever it's moving from one place to another, such as between nodes in your cluster or between your cluster and your application.

Remediation

To configure in-transit encryption on an ElastiCache replication group, see Enabling in-transit encryption in the *Amazon ElastiCache User Guide*.

[ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6, PCI DSS v4.0.1/8.3.1

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::ElastiCache::ReplicationGroup`

AWS Config rule: `elasticache-repl-grp-redis-auth-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether an ElastiCache (Redis OSS) replication group has Redis OSS AUTH enabled. The control fails if the Redis OSS version of the replication group nodes is below 6.0 and `AuthToken` isn't in use.

When you use Redis authentication tokens, or passwords, Redis requires a password before allowing clients to run commands, which improves data security. For Redis 6.0 and later versions, we recommend using Role-Based Access Control (RBAC). Since RBAC is not supported for Redis versions earlier than 6.0, this control only evaluates versions which can't use the RBAC feature.

Remediation

To use Redis AUTH on an ElastiCache (Redis OSS) replication group, see Modifying the AUTH token on an existing ElastiCache (Redis OSS) cluster in the *Amazon ElastiCache User Guide*.

[ElastiCache.7] ElastiCache clusters should not use the default subnet group

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(5)

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::ElastiCache::CacheCluster`

AWS Config rule: `elasticache-subnet-group-check`

Schedule type: Periodic

Parameters: None

This control checks whether an ElastiCache cluster is configured with a custom subnet group. The control fails if `CacheSubnetGroupName` for an ElastiCache cluster has the value `default`.

When launching an ElastiCache cluster, a default subnet group is created if one doesn't exist already. The default group uses subnets from the default Virtual Private Cloud (VPC). We recommend using custom subnet groups that are more restrictive of the subnets that the cluster resides in, and the networking that the cluster inherits from the subnets.

Remediation

To create a new subnet group for an ElastiCache cluster, see Creating a subnet group in the *Amazon ElastiCache User Guide*.

Security Hub controls for Elastic Beanstalk

These AWS Security Hub controls evaluate the AWS Elastic Beanstalk service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled

Related requirements: NIST.800-53.r5 CA-7,NIST.800-53.r5 SI-2

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: `AWS::ElasticBeanstalk::Environment`

AWS Config rule: `beanstalk-enhanced-health-reporting-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether enhanced health reporting is enabled for your AWS Elastic Beanstalk environments.

Elastic Beanstalk enhanced health reporting enables a more rapid response to changes in the health of the underlying infrastructure. These changes could result in a lack of availability of the application.

Elastic Beanstalk enhanced health reporting provides a status descriptor to gauge the severity of the identified issues and identify possible causes to investigate. The Elastic Beanstalk health agent,

included in supported Amazon Machine Images (AMIs), evaluates logs and metrics of environment EC2 instances.

For additional information, see Enhanced health reporting and monitoring in the *AWS Elastic Beanstalk Developer Guide*.

Remediation

For instructions on how to enable enhanced health reporting, see Enabling enhanced health reporting using the Elastic Beanstalk console in the *AWS Elastic Beanstalk Developer Guide*.

[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled

Related requirements: NIST.800-53.r5 SI-2,NIST.800-53.r5 SI-2(2),NIST.800-53.r5 SI-2(4),NIST.800-53.r5 SI-2(5), PCI DSS v4.0.1/6.3.3

Category: Identify > Vulnerability, patch, and version management

Severity: High

Resource type: `AWS::ElasticBeanstalk::Environment`

AWS Config rule: `elastic-beanstalk-managed-updates-enabled`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`UpdateLevel`	Version update level	Enum	`minor`, `patch`	No default value

This control checks whether managed platform updates are enabled for an Elastic Beanstalk environment. The control fails if no managed platform updates are enabled. By default, the control passes if any type of platform update is enabled. Optionally, you can provide a custom parameter value to require a specific update level.

Enabling managed platform updates ensures that the latest available platform fixes, updates, and features for the environment are installed. Keeping up to date with patch installation is an important step in securing systems.

Remediation

To enable managed platform updates, see To configure managed platform updates under Managed platform updates in the *AWS Elastic Beanstalk Developer Guide*.

[ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch

Related requirements: PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: High

Resource type: `AWS::ElasticBeanstalk::Environment`

AWS Config rule: `elastic-beanstalk-logs-to-cloudwatch`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`RetentionInDays`	Number of days to keep log events before expiration	Enum	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365 , 400, 545, 731, 1827, 3653	No default value

This control checks whether an Elastic Beanstalk environment is configured to send logs to CloudWatch Logs. The control fails if an Elastic Beanstalk environment isn't configured to send logs to CloudWatch Logs. Optionally, you can provide a custom value for the `RetentionInDays`

parameter if you want the control to pass only if logs are retained for the specified number of days before expiration.

CloudWatch helps you collect and monitor various metrics for your applications and infrastructure resources. You can also use CloudWatch to configure alarm actions based on specific metrics. We recommend integrating Elastic Beanstalk with CloudWatch to get increased visibility into your Elastic Beanstalk environment. Elastic Beanstalk logs include the eb-activity.log, access logs from the environment nginx or Apache proxy server, and logs that are specific to an environment.

Remediation

To integrate Elastic Beanstalk with CloudWatch Logs, see Streaming instance logs to CloudWatch Logs in the *AWS Elastic Beanstalk Developer Guide*.

Security Hub controls for Elastic Load Balancing

These AWS Security Hub controls evaluate the Elastic Load Balancing service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS

Related requirements: PCI DSS v3.2.1/2.3,PCI DSS v3.2.1/4.1, NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6)

Category: Detect > Detection services

Severity: Medium

Resource type: `AWS::ElasticLoadBalancingV2::LoadBalancer`

AWS Config rule: alb-http-to-https-redirection-check

Schedule type: Periodic

Parameters: None

This control checks whether HTTP to HTTPS redirection is configured on all HTTP listeners of Application Load Balancers. The control fails if any of the HTTP listeners of Application Load Balancers do not have HTTP to HTTPS redirection configured.

Before you start to use your Application Load Balancer, you must add one or more listeners. A listener is a process that uses the configured protocol and port to check for connection requests. Listeners support both the HTTP and HTTPS protocols. You can use an HTTPS listener to offload the work of encryption and decryption to your load balancer. To enforce encryption in transit, you should use redirect actions with Application Load Balancers to redirect client HTTP requests to an HTTPS request on port 443.

To learn more, see Listeners for your Application Load Balancers in *User Guide for Application Load Balancers*.

Remediation

To redirect HTTP requests to HTTPS, you must add an Application Load Balancer listener rule or edit an existing rule.

For instructions on adding a new rule, see Add a rule in the *User Guide for Application Load Balancers*. For **Protocol : Port**, choose **HTTP**, and then enter **80**. For **Add action, Redirect to**, choose **HTTPS**, and then enter **443**.

For instructions on editing an existing rule, see Edit a rule in the *User Guide for Application Load Balancers*. For **Protocol : Port**, choose **HTTP**, and then enter **80**. For **Add action, Redirect to**, choose **HTTPS**, and then enter **443**.

[ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(5), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), NIST.800-171.r2 3.13.8

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::ElasticLoadBalancing::LoadBalancer`

AWS Config rule: elb-acm-certificate-required

Schedule type: Change triggered

Parameters: None

This control checks whether the Classic Load Balancer uses HTTPS/SSL certificates provided by AWS Certificate Manager (ACM). The control fails if the Classic Load Balancer configured with HTTPS/SSL listener does not use a certificate provided by ACM.

To create a certificate, you can use either ACM or a tool that supports the SSL and TLS protocols, such as OpenSSL. Security Hub recommends that you use ACM to create or import certificates for your load balancer.

ACM integrates with Classic Load Balancers so that you can deploy the certificate on your load balancer. You also should automatically renew these certificates.

Remediation

For information about how to associate an ACM SSL/TLS certificate with a Classic Load Balancer, see the AWS Knowledge Center article How can I associate an ACM SSL/TLS certificate with a Classic, Application, or Network Load Balancer?

[ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), NIST.800-171.r2 3.13.8, NIST.800-171.r2 3.13.15, PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::ElasticLoadBalancing::LoadBalancer`

AWS Config rule: elb-tls-https-listeners-only

Schedule type: Change triggered

Parameters: None

This control checks whether your Classic Load Balancer listeners are configured with HTTPS or TLS protocol for front-end (client to load balancer) connections. The control is applicable if a Classic

Load Balancer has listeners. If your Classic Load Balancer does not have a listener configured, then the control does not report any findings.

The control passes if the Classic Load Balancer listeners are configured with TLS or HTTPS for front-end connections.

The control fails if the listener is not configured with TLS or HTTPS for front-end connections.

Before you start to use a load balancer, you must add one or more listeners. A listener is a process that uses the configured protocol and port to check for connection requests. Listeners can support both HTTP and HTTPS/TLS protocols. You should always use an HTTPS or TLS listener, so that the load balancer does the work of encryption and decryption in transit.

Remediation

To remediate this issue, update your listeners to use the TLS or HTTPS protocol.

To change all noncompliant listeners to TLS/HTTPS listeners

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select your Classic Load Balancer.
4. On the **Listeners** tab, choose **Edit**.
5. For all listeners where **Load Balancer Protocol** is not set to HTTPS or SSL, change the setting to HTTPS or SSL.
6. For all modified listeners, on the **Certificates** tab, choose **Change default**.
7. For **ACM and IAM certificates**, select a certificate.
8. Choose **Save as default**.
9. After you update all of the listeners, choose **Save**.

[ELB.4] Application Load Balancer should be configured to drop invalid http headers

Related requirements: NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8(2), PCI DSS v4.0.1/6.2.4

Category: Protect > Network Security

Severity: Medium

Resource type: `AWS::ElasticLoadBalancingV2::LoadBalancer`

AWS Config rule: alb-http-drop-invalid-header-enabled

Schedule type: Change triggered

Parameters: None

This control evaluates whether an Application Load Balancer is configured to drop invalid HTTP headers. The control fails if the value of `routing.http.drop_invalid_header_fields.enabled` is set to `false`.

By default, Application Load Balancers are not configured to drop invalid HTTP header values. Removing these header values prevents HTTP desync attacks.

> **Note**
>
> We recommend disabling this control if ELB.12 is enabled in your account. For more information, see [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode.

Remediation

To remediate this issue, configure your load balancer to drop invalid header fields.

To configure the load balancer to drop invalid header fields

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Load balancers**.
3. Choose an Application Load Balancer.
4. From **Actions**, choose **Edit attributes**.
5. Under **Drop Invalid Header Fields**, choose **Enable**.
6. Choose **Save**.

[ELB.5] Application and Classic Load Balancers logging should be enabled

Related requirements: NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::ElasticLoadBalancing::LoadBalancer`, `AWS::ElasticLoadBalancingV2::LoadBalancer`

AWS Config rule: elb-logging-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether the Application Load Balancer and the Classic Load Balancerhave logging enabled. The control fails if `access_logs.s3.enabled` is `false`.

Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and to troubleshoot issues.

To learn more, see Access logs for your Classic Load Balancer in *User Guide for Classic Load Balancers*.

Remediation

To enable access logs, see Step 3: Configure access logs in the *User Guide for Application Load Balancers*.

[ELB.6] Application, Gateway, and Network Load Balancers should have deletion protection enabled

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-3, NIST.800-53.r5 SC-5(2)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::ElasticLoadBalancingV2::LoadBalancer`

AWS Config rule: elb-deletion-protection-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Application, Gateway, or Network Load Balancer has deletion protection enabled. The control fails if deletion protection is disabled.

Enable deletion protection to protect your Application, Gateway, or Network Load Balancer from deletion.

Remediation

To prevent your load balancer from being deleted accidentally, you can enable deletion protection. By default, deletion protection is disabled for your load balancer.

If you enable deletion protection for your load balancer, you must disable delete protection before you can delete the load balancer.

To enable deletion protection for an Application Load Balancer, see Deletion protection in the *User Guide for Application Load Balancers*. To enable deletion protection for a Gateway Load Balancer, see Deletion protection in the *User Guide for Gateway Load Balancers*. To enable deletion protection for a Network Load Balancer, see Deletion protection in the *User Guide for Network Load Balancers*.

[ELB.7] Classic Load Balancers should have connection draining enabled

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Recover > Resilience

Severity: Medium

Resource type: `AWS::ElasticLoadBalancing::LoadBalancer`

AWS Config rule: `elb-connection-draining-enabled` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether Classic Load Balancers have connection draining enabled.

Enabling connection draining on Classic Load Balancers ensures that the load balancer stops sending requests to instances that are de-registering or unhealthy. It keeps the existing connections open. This is particularly useful for instances in Auto Scaling groups, to ensure that connections aren't severed abruptly.

Remediation

To enable connection draining on Classic Load Balancers, see Configure connection draining for your Classic Load Balancer in *User Guide for Classic Load Balancers*.

[ELB.8] Classic Load Balancers with SSL listeners should use a predefined security policy that has strong AWS Configuration

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), NIST.800-171.r2 3.13.8, NIST.800-171.r2 3.13.15, PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::ElasticLoadBalancing::LoadBalancer`

AWS Config rule: elb-predefined-security-policy-ssl-check

Schedule type: Change triggered

Parameters:

- `predefinedPolicyName`: `ELBSecurityPolicy-TLS-1-2-2017-01` (not customizable)

This control checks whether your Classic Load Balancer HTTPS/SSL listeners use the predefined policy `ELBSecurityPolicy-TLS-1-2-2017-01`. The control fails if the Classic Load Balancer HTTPS/SSL listeners do not use `ELBSecurityPolicy-TLS-1-2-2017-01`.

A security policy is a combination of SSL protocols, ciphers, and the Server Order Preference option. Predefined policies control the ciphers, protocols, and preference orders to support during SSL negotiations between a client and load balancer.

Using `ELBSecurityPolicy-TLS-1-2-2017-01` can help you to meet compliance and security standards that require you to disable specific versions of SSL and TLS. For more information, see Predefined SSL security policies for Classic Load Balancers in *User Guide for Classic Load Balancers*.

Remediation

For information on how to use the predefined security policy `ELBSecurityPolicy-TLS-1-2-2017-01` with a Classic Load Balancer, see Configure security settings in *User Guide for Classic Load Balancers*.

[ELB.9] Classic Load Balancers should have cross-zone load balancing enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::ElasticLoadBalancing::LoadBalancer`

AWS Config rule: elb-cross-zone-load-balancing-enabled

Schedule type: Change triggered

Parameters: None

This control checks if cross-zone load balancing is enabled for the Classic Load Balancers (CLBs). The control fails if cross-zone load balancing is not enabled for a CLB.

A load balancer node distributes traffic only across the registered targets in its Availability Zone. When cross-zone load balancing is disabled, each load balancer node distributes traffic only across the registered targets in its Availability Zone. If the number of registered targets is not same across the Availability Zones, traffic wont be distributed evenly and the instances in one zone may end up over utilized compared to the instances in another zone. With cross-zone load balancing enabled, each load balancer node for your Classic Load Balancer distributes requests evenly across the registered instances in all enabled Availability Zones. For details see Cross-zone load balancing in the Elastic Load Balancing User Guide.

Remediation

To enable cross-zone load balancing in a Classic Load Balancer, see Enable cross-zone load balancing in the *User Guide for Classic Load Balancers*.

[ELB.10] Classic Load Balancer should span multiple Availability Zones

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::ElasticLoadBalancing::LoadBalancer`

AWS Config rule: clb-multiple-az

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`minAvailabilityZones`	Minimum number of Availability Zones	Enum	`2, 3, 4, 5, 6`	2

This control checks whether a Classic Load Balancer has been configured to span at least the specified number of Availability Zones (AZs). The control fails if the Classic Load Balancer does not span at least the specified number of AZs. Unless you provide a custom parameter value for the minimum number of AZs, Security Hub uses a default value of two AZs.

A Classic Load Balancer can be set up to distribute incoming requests across Amazon EC2 instances in a single Availability Zone or multiple Availability Zones. A Classic Load Balancer that does not span multiple Availability Zones is unable to redirect traffic to targets in another Availability Zone if the sole configured Availability Zone becomes unavailable.

Remediation

To add Availability Zones to a Classic Load Balancer, see Add or remove subnets for your Classic Load Balancer in the *User Guide for Classic Load Balancers*.

[ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode

Related requirements: NIST.800-53.r5 AC-4(21), NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, PCI DSS v4.0.1/6.2.4

Category: Protect > Data Protection > Data integrity

Severity: Medium

Resource type: `AWS::ElasticLoadBalancingV2::LoadBalancer`

AWS Config rule: alb-desync-mode-check

Schedule type: Change triggered

Parameters:

- `desyncMode`: `defensive, strictest` (not customizable)

This control checks whether an Application Load Balancer is configured with defensive or strictest desync mitigation mode. The control fails if an Application Load Balancer is not configured with defensive or strictest desync mitigation mode.

HTTP Desync issues can lead to request smuggling and make applications vulnerable to request queue or cache poisoning. In turn, these vulnerabilities can lead to credential stuffing or execution of unauthorized commands. Application Load Balancers configured with defensive or strictest desync mitigation mode protect your application from security issues that may be caused by HTTP Desync.

Remediation

To update desync mitigation mode of an Application Load Balancer, see Desync mitigation mode in the *User Guide for Application Load Balancers*.

[ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::ElasticLoadBalancingV2::LoadBalancer`

AWS Config rule: elbv2-multiple-az

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`minAvailabilityZones`	Minimum number of Availability Zones	Enum	`2, 3, 4, 5, 6`	`2`

This control checks whether an Elastic Load Balancer V2 (Application, Network, or Gateway Load Balancer) has registered instances from at least the specified number of Availability Zones (AZs). The control fails if an Elastic Load Balancer V2 doesn't have instances registered in at least the specified number of AZs. Unless you provide a custom parameter value for the minimum number of AZs, Security Hub uses a default value of two AZs.

Elastic Load Balancing automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones. Elastic Load Balancing scales your load balancer as your incoming traffic changes over time. It is recommended to configure at least two availability zones to ensure availability of services, as the Elastic Load Balancer will be able to direct traffic to another availability zone if one becomes unavailable. Having multiple availability zones configured will help eliminate having a single point of failure for the application.

Remediation

To add an Availability Zone to an Application Load Balancer, see Availability Zones for your Application Load Balancer in the *User Guide for Application Load Balancers*. To add an Availability

Zone to an Network Load Balancer, see Network Load Balancers in the *User Guide for Network Load Balancers*. To add an Availability Zone to a Gateway Load Balancer, see Create a Gateway Load Balancer in the *User Guide for Gateway Load Balancers*.

[ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode

Related requirements: NIST.800-53.r5 AC-4(21), NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, PCI DSS v4.0.1/6.2.4

Category: Protect > Data Protection > Data integrity

Severity: Medium

Resource type: `AWS::ElasticLoadBalancing::LoadBalancer`

AWS Config rule: clb-desync-mode-check

Schedule type: Change triggered

Parameters:

- `desyncMode`: `defensive, strictest` (not customizable)

This control checks whether a Classic Load Balancer is configured with defensive or strictest desync mitigation mode. The control fails if the Classic Load Balancer isn't configured with defensive or strictest desync mitigation mode.

HTTP Desync issues can lead to request smuggling and make applications vulnerable to request queue or cache poisoning. In turn, these vulnerabilities can lead to credential hijacking or execution of unauthorized commands. Classic Load Balancers configured with defensive or strictest desync mitigation mode protect your application from security issues that may be caused by HTTP Desync.

Remediation

To update desync mitigation mode on a Classic Load Balancer, see Modify desync mitigation mode in the *User Guide for Classic Load Balancers*.

[ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL

Related requirements: NIST.800-53.r5 AC-4(21)

Category: Protect > Protective services

Severity: Medium

Resource type: `AWS::ElasticLoadBalancingV2::LoadBalancer`

AWS Config rule: alb-waf-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Application Load Balancer is associated with an AWS WAF Classic or AWS WAF web access control list (web ACL). The control fails if the `Enabled` field for the AWS WAF configuration is set to `false`.

AWS WAF is a web application firewall that helps protect web applications and APIs from attacks. With AWS WAF, you can configure a web ACL, which is a set of rules that allow, block, or count web requests based on customizable web security rules and conditions that you define. We recommend associating your Application Load Balancer with an AWS WAF web ACL to help protect it from malicious attacks.

Remediation

To associate an Application Load Balancer with a web ACL, see Associating or disassociating a web ACL with an AWS resource in the *AWS WAF Developer Guide*.

[ELB.17] Application and Network Load Balancers with listeners should use recommended security policies

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::ElasticLoadBalancingV2::Listener`

AWS Config rule: elbv2-predefined-security-policy-ssl-check

Schedule type: Change triggered

Parameters: `sslPolicies`: `ELBSecurityPolicy-TLS13-1-2-2021-06, ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04, ELBSecurityPolicy-TLS13-1-3-2021-06, ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04, ELBSecurityPolicy-TLS13-1-2-Res-2021-06, ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04` (not customizable)

This control checks whether the HTTPS listener for an Application Load Balancer or the TLS listener for a Network Load Balancer is configured to encrypt data in transit by using a recommended security policy. The control fails if the HTTPS or TLS listener for a load balancer isn't configured to use a recommended security policy.

Elastic Load Balancing uses an SSL negotiation configuration, known as a *security policy*, to negotiate connections between a client and a load balancer. The security policy specifies a combination of protocols and ciphers. The protocol establishes a secure connection between a client and a server. A cipher is an encryption algorithm that uses encryption keys to create a coded message. During the connection negotiation process, the client and the load balancer present a list of ciphers and protocols that they each support, in order of preference. Using a recommended security policy for a load balancer can help you meet compliance and security standards.

Remediation

For information about recommended security policies and how to update listeners, see the following sections of the *Elastic Load Balancing User Guides*: Security policies for Application Load Balancers, Security policies for Network Load Balancers, Update an HTTPS listener for your Application Load Balancer, and Update a listener for your Network Load Balancer.

[ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::ElasticLoadBalancingV2::Listener`

AWS Config rule: elbv2-listener-encryption-in-transit

Schedule type: Change triggered

Parameters: None

This control checks whether the listener for an Application Load Balancer or Network Load Balancer is configured to use a secure protocol for encryption of data in transit. The control fails if an Application Load Balancer listener isn't configured to use the HTTPS protocol, or a Network Load Balancer listener isn't configured to use the TLS protocol.

To encrypt data that's transmitted between a client and a load balancer, Elastic Load Balancer listeners should be configured to use industry-standard security protocols: HTTPS for Application Load Balancers, or TLS for Network Load Balancers. Otherwise, data that's transmitted between a client and a load balancer is vulnerable to interception, tampering, and unauthorized access. Use of HTTPS or TLS by a listener aligns with security best practices and helps ensure the confidentiality and integrity of data during transmission. This is particularly important for applications that handle sensitive information, or must comply with security standards that require encryption of data in transit.

Remediation

For information about configuring security protocols for listeners, see the following sections of the *Elastic Load Balancing User Guides*: Create an HTTPS listener for your Application Load Balancer and Create a listener for your Network Load Balancer.

Security Hub for Elasticsearch

These AWS Security Hub controls evaluate the Elasticsearch service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[ES.1] Elasticsearch domains should have encryption at-rest enabled

Related requirements: PCI DSS v3.2.1/3.4, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::Elasticsearch::Domain`

AWS Config rule: `elasticsearch-encrypted-at-rest`

Schedule type: Periodic

Parameters: None

This control checks whether Elasticsearch domains have encryption at rest configuration enabled. The check fails if encryption at rest is not enabled.

For an added layer of security for your sensitive data in OpenSearch, you should configure your OpenSearch to be encrypted at rest. Elasticsearch domains offer encryption of data at rest. The feature uses AWS KMS to store and manage your encryption keys. To perform the encryption, it uses the Advanced Encryption Standard algorithm with 256-bit keys (AES-256).

To learn more about OpenSearch encryption at rest, see Encryption of data at rest for Amazon OpenSearch Service in the *Amazon OpenSearch Service Developer Guide*.

Certain instance types, such as `t.small` and `t.medium`, don't support encryption of data at rest. For details, see Supported instance types in the *Amazon OpenSearch Service Developer Guide*.

Remediation

To enable encryption at rest for new and existing Elasticsearch domains, see Enabling encryption of data at rest in the *Amazon OpenSearch Service Developer Guide*.

[ES.2] Elasticsearch domains should not be publicly accessible

Related requirements: PCI DSS v3.2.1/1.2.1,PCI DSS v3.2.1/1.3.1,PCI DSS v3.2.1/1.3.2,PCI DSS v3.2.1/1.3.4,PCI DSS v3.2.1/1.3.6, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration > Resources within VPC

Severity: Critical

Resource type: `AWS::Elasticsearch::Domain`

AWS Config rule: `elasticsearch-in-vpc-only`

Schedule type: Periodic

Parameters: None

This control checks whether Elasticsearch domains are in a VPC. It does not evaluate the VPC subnet routing configuration to determine public access. You should ensure that Elasticsearch domains are not attached to public subnets. See Resource-based policies in the *Amazon OpenSearch Service Developer Guide*. You should also ensure that your VPC is configured according to the recommended best practices. See Security best practices for your VPC in the *Amazon VPC User Guide*.

Elasticsearch domains deployed within a VPC can communicate with VPC resources over the private AWS network, without the need to traverse the public internet. This configuration increases the security posture by limiting access to the data in transit. VPCs provide a number of network controls to secure access to Elasticsearch domains, including network ACL and security groups. Security Hub recommends that you migrate public Elasticsearch domains to VPCs to take advantage of these controls.

Remediation

If you create a domain with a public endpoint, you cannot later place it within a VPC. Instead, you must create a new domain and migrate your data. The reverse is also true. If you create a domain within a VPC, it cannot have a public endpoint. Instead, you must either create another domain or disable this control.

See Launching your Amazon OpenSearch Service domains within a VPC in the *Amazon OpenSearch Service Developer Guide*.

[ES.3] Elasticsearch domains should encrypt data sent between nodes

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::Elasticsearch::Domain`

AWS Config rule: `elasticsearch-node-to-node-encryption-check`

Schedule type: Change triggered

Parameters: None

This control checks whether an Elasticsearch domain has node-to-node encryption enabled. The control fails if the Elasticsearch domain doesn't have node-to-node encryption enabled. The control also produces failed findings if an Elasticsearch version doesn't support node-to-node encryption checks.

HTTPS (TLS) can be used to help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle or similar attacks. Only encrypted connections over HTTPS (TLS) should be allowed. Enabling node-to-node encryption for Elasticsearch domains ensures that intra-cluster communications are encrypted in transit.

There can be a performance penalty associated with this configuration. You should be aware of and test the performance trade-off before enabling this option.

Remediation

For information about enabling node-to-node encryption on new and existing domains, see Enabling node-to-node encryption in the *Amazon OpenSearch Service Developer Guide*.

[ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8)

Category: Identify - Logging

Severity: Medium

Resource type: `AWS::Elasticsearch::Domain`

AWS Config rule: `elasticsearch-logs-to-cloudwatch`

Schedule type: Change triggered

Parameters:

- `logtype = 'error'` (not customizable)

This control checks whether Elasticsearch domains are configured to send error logs to CloudWatch Logs.

You should enable error logs for Elasticsearch domains and send those logs to CloudWatch Logs for retention and response. Domain error logs can assist with security and access audits, and can help to diagnose availability issues.

Remediation

For information on how to enable log publishing, see Enabling log publishing (console) in the *Amazon OpenSearch Service Developer Guide*.

[ES.5] Elasticsearch domains should have audit logging enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::Elasticsearch::Domain`

AWS Config rule: `elasticsearch-audit-logging-enabled` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

- `cloudWatchLogsLogGroupArnList` (not customizable). Security Hub does not populate this parameter. Comma-separated list of CloudWatch Logs log groups that should be configured for audit logs.

  This rule is `NON_COMPLIANT` if the CloudWatch Logs log group of the Elasticsearch domain is not specified in this parameter list.

This control checks whether Elasticsearch domains have audit logging enabled. This control fails if an Elasticsearch domain does not have audit logging enabled.

Audit logs are highly customizable. They allow you to track user activity on your Elasticsearch clusters, including authentication successes and failures, requests to OpenSearch, index changes, and incoming search queries.

Remediation

For detailed instructions on enabling audit logs, see Enabling audit logs in the *Amazon OpenSearch Service Developer Guide*.

[ES.6] Elasticsearch domains should have at least three data nodes

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::Elasticsearch::Domain`

AWS Config rule: `elasticsearch-data-node-fault-tolerance` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether Elasticsearch domains are configured with at least three data nodes and `zoneAwarenessEnabled` is `true`.

An Elasticsearch domain requires at least three data nodes for high availability and fault-tolerance. Deploying an Elasticsearch domain with at least three data nodes ensures cluster operations if a node fails.

Remediation

To modify the number of data nodes in an Elasticsearch domain

1. Open the Amazon OpenSearch Service console at https://console.aws.amazon.com/aos/.
2. Under **Domains**, choose the name of the domain you want to edit.
3. Choose **Edit domain**.
4. Under **Data nodes**, set **Number of nodes** to a number greater than or equal to 3.

For three Availability Zone deployments, set to a multiple of three to ensure equal distribution across Availability Zones.

5. Choose **Submit**.

[ES.7] Elasticsearch domains should be configured with at least three dedicated master nodes

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::Elasticsearch::Domain`

AWS Configrule: `elasticsearch-primary-node-fault-tolerance` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether Elasticsearch domains are configured with at least three dedicated primary nodes. This control fails if the domain does not use dedicated primary nodes. This control passes if Elasticsearch domains have five dedicated primary nodes. However, using more than three primary nodes might be unnecessary to mitigate the availability risk, and will result in additional cost.

An Elasticsearch domain requires at least three dedicated primary nodes for high availability and fault-tolerance. Dedicated primary node resources can be strained during data node blue/green deployments because there are additional nodes to manage. Deploying an Elasticsearch domain with at least three dedicated primary nodes ensures sufficient primary node resource capacity and cluster operations if a node fails.

Remediation

To modify the number of dedicated primary nodes in an OpenSearch domain

1. Open the Amazon OpenSearch Service console at https://console.aws.amazon.com/aos/.
2. Under **Domains**, choose the name of the domain you want to edit.

3. Choose **Edit domain**.
4. Under **Dedicated master nodes**, set **Instance type** to the desired instance type.
5. Set **Number of master nodes** equal to three or greater.
6. Choose **Submit**.

[ES.8] Connections to Elasticsearch domains should be encrypted using the latest TLS security policy

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::Elasticsearch::Domain`

AWS Config rule: `elasticsearch-https-required` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This controls checks whether an Elasticsearch domain endpoint is configured to use the latest TLS security policy. The control fails if the Elasticsearch domain endpoint isn't configured to use the latest supported policy or if HTTPs isn't enabled. The current latest supported TLS security policy is `Policy-Min-TLS-1-2-PFS-2023-10`.

HTTPS (TLS) can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. Only encrypted connections over HTTPS (TLS) should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the impact of TLS. TLS 1.2 provides several security enhancements over previous versions of TLS.

Remediation

To enable TLS encryption, use the UpdateDomainConfig API operation to configure the DomainEndpointOptions object. This sets the `TLSSecurityPolicy`.

[ES.9] Elasticsearch domains should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Elasticsearch::Domain`

AWS Config rule: `tagged-elasticsearch-domain` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Elasticsearch domain has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the domain doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the domain isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals.

You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Elasticsearch domain, see Working with tags in the *Amazon OpenSearch Service Developer Guide*.

Security Hub controls for Amazon EMR

These AWS Security Hub controls evaluate the Amazon EMR (previously called Amazon Elastic MapReduce) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses

Related requirements: PCI DSS v3.2.1/1.2.1,PCI DSS v3.2.1/1.3.1,PCI DSS v3.2.1/1.3.2,PCI DSS v3.2.1/1.3.4,PCI DSS v3.2.1/1.3.6, PCI DSS v4.0.1/1.4.4, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::EMR::Cluster`

AWS Config rule: emr-master-no-public-ip

Schedule type: Periodic

Parameters: None

This control checks whether master nodes on Amazon EMR clusters have public IP addresses. The control fails if public IP addresses are associated with any of the master node instances.

Public IP addresses are designated in the `PublicIp` field of the `NetworkInterfaces` configuration for the instance. This control only checks Amazon EMR clusters that are in a `RUNNING` or `WAITING` state.

Remediation

During launch, you can control whether your instance in a default or nondefault subnet is assigned a public IPv4 address. By default, default subnets have this attribute set to `true`. Nondefault subnets have the IPv4 public addressing attribute set to `false`, unless it was created by the Amazon EC2 launch instance wizard. In that case, the attribute is set to `true`.

After launch, you can't manually disassociate a public IPv4 address from your instance.

To remediate a failed finding, you must launch a new cluster in a VPC with a private subnet that has the IPv4 public addressing attribute set to `false`. For instructions, see Launch clusters into a VPC in the *Amazon EMR Management Guide*.

[EMR.2] Amazon EMR block public access setting should be enabled

Related requirements: PCI DSS v4.0.1/1.4.4, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure access management > Resource not publicly accessible

Severity: Critical

Resource type: `AWS::::Account`

AWS Config rule: emr-block-public-access

Schedule type: Periodic

Parameters: None

This control checks whether your account is configured with Amazon EMR block public access. The control fails if the block public access setting isn't enabled or if any port other than port 22 is allowed.

Amazon EMR block public access prevents you from launching a cluster in a public subnet if the cluster has a security configuration that allows inbound traffic from public IP addresses on a port. When a user from your AWS account launches a cluster, Amazon EMR checks the port rules in the security group for the cluster and compares them with your inbound traffic rules. If the security group has an inbound rule that opens ports to the public IP addresses IPv4 0.0.0.0/0 or IPv6 ::/0, and those ports aren't specified as exceptions for your account, Amazon EMR doesn't let the user create the cluster.

> **Note**
>
> Block public access is enabled by default. To increase account protection, we recommend that you keep it enabled.

Remediation

To configure block public access for Amazon EMR, see Using Amazon EMR block public access in the *Amazon EMR Management Guide*.

[EMR.3] Amazon EMR security configurations should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CP-9(8), NIST.800-53.r5 SI-12

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::EMR::SecurityConfiguration`

AWS Config rule: emr-security-configuration-encryption-rest

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EMR security configuration has encryption at rest enabled. The control fails if the security configuration doesn't enable encryption at rest.

Data at rest refers to data that's stored in persistent, non-volatile storage for any duration. Encrypting data at rest helps you protect its confidentiality, which reduces the risk that an unauthorized user can access it.

Remediation

To enable encryption at rest in an Amazon EMR security configuration, see Configure data encryption in the *Amazon EMR Management Guide*.

[EMR.4] Amazon EMR security configurations should be encrypted in transit

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3)

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::EMR::SecurityConfiguration`

AWS Config rule: emr-security-configuration-encryption-transit

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EMR security configuration has encryption in transit enabled. The control fails if the security configuration doesn't enable encryption in transit.

Data in transit refers to data that moves from one location to another, such as between nodes in your cluster or between your cluster and your application. Data may move across the internet or within a private network. Encrypting data in transit reduces the risk that an unauthorized user can eavesdrop on network traffic.

Remediation

To enable encryption in transit in an Amazon EMR security configuration, see Configure data encryption in the *Amazon EMR Management Guide*.

Security Hub controls for EventBridge

These AWS Security Hub controls evaluate the Amazon EventBridge service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[EventBridge.2] EventBridge event buses should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Events::EventBus`

AWS Config rule:`tagged-events-eventbus` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon EventBridge event bus has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the event bus doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the event bus isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals.

You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an EventBridge event bus, see Amazon EventBridge tags in the *Amazon EventBridge User Guide*.

[EventBridge.3] EventBridge custom event buses should have a resource-based policy attached

Related requirements: NIST.800-53.r5 AC-2, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-6(3), PCI DSS v4.0.1/10.3.1

Category: Protect > Secure access management > Resource not publicly accessible

Severity: Low

Resource type: `AWS::Events::EventBus`

AWS Config rule: custom-eventbus-policy-attached

Schedule type: Change triggered

Parameters: None

This control checks if an Amazon EventBridge custom event bus has a resource-based policy attached. This control fails if the custom event bus doesn't have a resource-based policy.

By default, an EventBridge custom event bus doesn't have a resource-based policy attached. This allows principals in the account to access the event bus. By attaching a resource-based policy to the event bus, you can limit access to the event bus to specified accounts, as well as intentionally grant access to entities in another account.

Remediation

To attach a resource-based policy to an EventBridge custom event bus, see Using resource-based policies for Amazon EventBridge in the *Amazon EventBridge User Guide*.

[EventBridge.4] EventBridge global endpoints should have event replication enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::Events::Endpoint`

AWS Config rule: `global-endpoint-event-replication-enabled`

Schedule type: Change triggered

Parameters: None

This control checks if event replication is enabled for an Amazon EventBridge global endpoint. The control fails if event replication isn't enabled for a global endpoint.

Global endpoints help make your application Regional-fault tolerant. To start, you assign an Amazon Route 53 health check to the endpoint. When failover is initiated, the health check reports an "unhealthy" state. Within minutes of failover initiation, all custom events are routed to an event bus in the secondary Region and are processed by that event bus. When you use global endpoints, you can enable event replication. Event replication sends all custom events to the event buses in the primary and secondary Regions using managed rules. We recommend enabling event replication when setting up global endpoints. Event replication helps you verify that your global endpoints are configured correctly. Event replication is required to automatically recover from a failover event. If you don’t have event replication enabled, you’ll have to manually reset the Route 53 health check to "healthy" before events are rerouted back to the primary Region.

> **Note**
>
> If you're using custom event buses, you'll need a custom even bus in each Region with the same name and in the same account for failover to work properly. Enabling event

replication can increase your monthly cost. For information about pricing, see Amazon EventBridge pricing.

Remediation

To enable event replication for EventBridge global endpoints, see Create a global endpoint in the *Amazon EventBridge User Guide*. For **Event replication**, select **Event replication enabled**.

Security Hub controls for Amazon Fraud Detector

These Security Hub controls evaluate the Amazon Fraud Detector service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[FraudDetector.1] Amazon Fraud Detector entity types should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::FraudDetector::EntityType`

AWS Config rule: `frauddetector-entity-type-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Fraud Detector entity type has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the entity type doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the entity type isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an Amazon Fraud Detector entity type (console)

1. Open the Amazon Fraud Detector console at https://console.aws.amazon.com/frauddetector.
2. In the navigation pane, choose **Entities**.
3. Select an entity type from the list.
4. In the **entity type tags** section, choose **Manage tags**.
5. Choose **Add new tag**. Enter the key and value for the tag. Repeat for additional key-value pairs.
6. When you are finished adding tags, choose **Save**.

[FraudDetector.2] Amazon Fraud Detector labels should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::FraudDetector::Label`

AWS Config rule: `frauddetector-label-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Fraud Detector label has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the label doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the label isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM

principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an Amazon Fraud Detector label (console)

1. Open the Amazon Fraud Detector console at https://console.aws.amazon.com/frauddetector.
2. In the navigation pane, choose **Labels**.
3. Select a label from the list.
4. In the **labels tags** section, choose **Manage tags**.
5. Choose **Add new tag**. Enter the key and value for the tag. Repeat for additional key-value pairs.
6. When you are finished adding tags, choose **Save**.

[FraudDetector.3] Amazon Fraud Detector outcomes should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::FraudDetector::Outcome`

AWS Config rule: `frauddetector-outcome-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Fraud Detector outcome has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the outcome doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the outcome isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an Amazon Fraud Detector outcome (console)

1. Open the Amazon Fraud Detector console at https://console.aws.amazon.com/frauddetector.
2. In the navigation pane, choose **Outcomes**.
3. Select an outcome from the list.
4. In the **outcomes tags** section, choose **Manage tags**.
5. Choose **Add new tag**. Enter the key and value for the tag. Repeat for additional key-value pairs.
6. When you are finished adding tags, choose **Save**.

[FraudDetector.4] Amazon Fraud Detector variables should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::FraudDetector::Variable`

AWS Config rule: `frauddetector-variable-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Fraud Detector variable has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the variable doesn't have any

tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the variable isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an Amazon Fraud Detector variable (console)

1. Open the Amazon Fraud Detector console at https://console.aws.amazon.com/frauddetector.
2. In the navigation pane, choose **Variables**.
3. Select a variable from the list.
4. In the **variables tags** section, choose **Manage tags**.
5. Choose **Add new tag**. Enter the key and value for the tag. Repeat for additional key-value pairs.
6. When you are finished adding tags, choose **Save**.

Security Hub controls for Amazon FSx

These AWS Security Hub controls evaluate the Amazon FSx service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::FSx::FileSystem`

AWS Config rule: fsx-openzfs-copy-tags-enabled

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon FSx for OpenZFS file system is configured to copy tags to backups and volumes. The control fails if the OpenZFS file system isn't configured to copy tags to backups and volumes.

Identification and inventory of your IT assets is an important aspect of governance and security. Tags help you categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type because you can quickly identify a specific resource based on the tags that you assigned to it.

Remediation

For information about configuring an FSx for OpenZFS file system to copy tags to backups and volumes, see Updating a file system in the *Amazon FSx for OpenZFS User Guide*.

[FSx.2] FSx for Lustre file systems should be configured to copy tags to backups

Related requirements: NIST.800-53.r5 CP-9, NIST.800-53.r5 CM-8

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::FSx::FileSystem`

AWS Config rule: fsx-lustre-copy-tags-to-backups

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon FSx for Lustre file system is configured to copy tags to backups and volumes. The control fails if the Lustre file system isn't configured to copy tags to backups and volumes.

Identification and inventory of your IT assets is an important aspect of governance and security. Tags help you categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type because you can quickly identify a specific resource based on the tags that you assigned to it.

Remediation

For information about configuring an FSx for Lustre file system to copy tags to backups, see Copying backups within the same AWS account in the *Amazon FSx for Lustre User Guide*.

[FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::FSx::FileSystem`

AWS Config rule: fsx-openzfs-deployment-type-check

Schedule type: Periodic

Parameters: `deploymentTypes: MULTI_AZ_1` (not customizable)

This control checks whether an Amazon FSx for OpenZFS file system is configured to use the multiple Availability Zones (Multi-AZ) deployment type. The control fails if the file system isn't configured to use the Multi-AZ deployment type.

Amazon FSx for OpenZFS supports several deployment types for file systems: *Multi-AZ (HA)*, *Single-AZ (HA)*, and *Single-AZ (non-HA)*. The deployment types offer different levels of availability and durability. Multi-AZ (HA) file systems are composed of a high-availability (HA) pair of file servers that are spread across two Availability Zones (AZs). We recommend using the Multi-AZ (HA) deployment type for most production workloads due to the high availability and durability model that it provides.

Remediation

You can configure an Amazon FSx for OpenZFS file system to use the Multi-AZ deployment type when you create the file system. You can't change the deployment type for an existing FSx for OpenZFS file system.

For information about deployment types and options for FSx for OpenZFS file systems, see Availability and durability for Amazon FSx for OpenZFS and Managing file system resources in the *Amazon FSx for OpenZFS User Guide*.

[FSx.4] FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::FSx::FileSystem`

AWS Config rule: fsx-ontap-deployment-type-check

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`deploymentTypes`	A list of deployment types to include in the evaluation. The control generates a `FAILED` finding if a file system isn't configured to use	Enum	`MULTI_AZ_1` , `MULTI_AZ_2`	`MULTI_AZ_1` , `MULTI_AZ_2`

Parameter	Description	Type	Allowed custom values	Security Hub default value
	a deployment type specified in the list.			

This control checks whether an Amazon FSx for NetApp ONTAP file system is configured to use a multiple Availability Zones (Multi-AZ) deployment type. The control fails if the file system isn't configured to use a Multi-AZ deployment type. You can optionally specify a list of deployment types to include in the evaluation.

Amazon FSx for NetApp ONTAP supports several deployment types for file systems: *Single-AZ 1*, *Single-AZ 2*, *Multi-AZ 1*, and *Multi-AZ 2*. The deployment types offer different levels of availability and durability. We recommend using a Multi-AZ deployment type for most production workloads due to the high availability and durability model that Multi-AZ deployment types provide. Multi-AZ file systems support all the availability and durability features of Single-AZ file systems. In addition, they're designed to provide continuous availability to data even when an Availability Zone (AZ) is unavailable.

Remediation

You can't change the deployment type for an existing Amazon FSx for NetApp ONTAP file system. However, you can back up the data, and then restore it on a new file system that uses a Multi-AZ deployment type.

For information about deployment types and options for FSx for ONTAP file systems, see Availability, durability, and deployment options and Managing file systems in the *FSx for ONTAP User Guide*.

[FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::FSx::FileSystem`

AWS Config rule: fsx-windows-deployment-type-check

Schedule type: Periodic

Parameters: `deploymentTypes: MULTI_AZ_1` (not customizable)

This control checks whether an Amazon FSx for Windows File Server file system is configured to use the multiple Availability Zones (Multi-AZ) deployment type. The control fails if the file system isn't configured to use the Multi-AZ deployment type.

Amazon FSx for Windows File Server supports two deployment types for file systems: *Single-AZ* and *Multi-AZ*. The deployment types offer different levels of availability and durability. Single-AZ file systems are composed of a single Windows file server instance and a set of storage volumes within a single Availability Zone (AZ). Multi-AZ file systems are composed of a high-availability cluster of Windows file servers spread across two Availability Zones. We recommend using the Multi-AZ deployment type for most production workloads due to the high availability and durability model that it provides.

Remediation

You can configure an Amazon FSx for Windows File Server file system to use the Multi-AZ deployment type when you create the file system. You can't change the deployment type for an existing FSx for Windows File Server file system.

For information about deployment types and options for FSx for Windows File Server file systems, see Availability and durability: Single-AZ and Multi-AZ file systems and Getting started with Amazon FSx for Windows File Server in the *Amazon FSx for Windows File Server User Guide*.

Security Hub controls for Global Accelerator

These AWS Security Hub controls evaluate the AWS Global Accelerator service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[GlobalAccelerator.1] Global Accelerator accelerators should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::GlobalAccelerator::Accelerator`

AWS Config rule: `tagged-globalaccelerator-accelerator` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS Global Accelerator accelerator has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the accelerator doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the accelerator isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Global Accelerator global accelerator, see see Tagging in AWS Global Accelerator in the *AWS Global Accelerator Developer Guide*.

Security Hub controls for AWS Glue

These AWS Security Hub controls evaluate the AWS Glue service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Glue.1] AWS Glue jobs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Glue::Job`

AWS Config rule: `tagged-glue-job` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Glue job has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the job doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the job isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a AWS Glue job, see AWS tags in AWS Glue in the *AWS Glue User Guide*.

[Glue.3] AWS Glue machine learning transforms should be encrypted at rest

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::Glue::MLTransform`

AWS Config rule: glue-ml-transform-encrypted-at-rest

Schedule type: Change triggered

Parameters: No

This control checks whether an AWS Glue machine learning transform is encrypted at rest. The control fails if the machine learning transform isn't encrypted at rest.

Data at rest refers to data that's stored in persistent, non-volatile storage for any duration. Encrypting data at rest helps you protect its confidentiality, which reduces the risk that an unauthorized user can access it.

Remediation

To configure encryption for AWS Glue machine learning transforms, see Working with machine learning transforms in the *AWS Glue User Guide*.

[Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5)

Category: Identify > Vulnerability, patch, and version management

Severity: Medium

Resource type: `AWS::Glue::Job`

AWS Config rule: glue-spark-job-supported-version

Schedule type: Change triggered

Parameters: `minimumSupportedGlueVersion`: `3.0` (not customizable)

This control checks whether an AWS Glue for Spark job is configured to run on a supported version of AWS Glue. The control fails if the Spark job is configured to run on a version of AWS Glue that's earlier than the minimum supported version.

> **Note**
>
> This control also generates a `FAILED` finding for an AWS Glue for Spark job if the AWS Glue version (`GlueVersion`) property doesn't exist or is null in the configuration item (CI) for the job. In such cases, the finding includes the following annotation: `GlueVersion is null or missing in glueetl job configuration`. To address this type of `FAILED` finding, add the `GlueVersion` property to the job's configuration. For a list of supported versions and runtime environments, see AWS Glue Versions in the *AWS Glue User Guide*.

Running AWS Glue Spark jobs on current versions of AWS Glue can optimize performance, security, and access to the latest features of AWS Glue. It can also help safeguard against security vulnerabilities. For example, a new version might be released to provide security updates, address issues, or introduce new features.

Remediation

For information about migrating a Spark job to a supported version of AWS Glue, see Migrating AWS Glue for Spark jobs in the *AWS Glue User Guide*.

Security Hub controls for Amazon GuardDuty

These AWS Security Hub controls evaluate the Amazon GuardDuty service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[GuardDuty.1] GuardDuty should be enabled

Related requirements: NIST.800-53.r5 AC-2(12), NIST.800-53.r5 AU-6(1), NIST.800-53.r5 AU-6(5), NIST.800-53.r5 CA-7, NIST.800-53.r5 CM-8(3), NIST.800-53.r5 RA-3(4), NIST.800-53.r5 SA-11(1), NIST.800-53.r5 SA-11(6), NIST.800-53.r5 SA-15(2), NIST.800-53.r5 SA-15(8), NIST.800-53.r5 SA-8(19), NIST.800-53.r5 SA-8(21), NIST.800-53.r5 SA-8(25), NIST.800-53.r5 SC-5, NIST.800-53.r5 SC-5(1), NIST.800-53.r5 SC-5(3), NIST.800-53.r5 SI-20, NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4, NIST.800-53.r5 SI-4(1), NIST.800-53.r5 SI-4(13), NIST.800-53.r5 SI-4(2), NIST.800-53.r5 SI-4(22), NIST.800-53.r5 SI-4(25), NIST.800-53.r5 SI-4(4), NIST.800-53.r5 SI-4(5), NIST.800-171.r2 3.4.2, NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7, PCI DSS v3.2.1/11.4, PCI DSS v4.0.1/11.5.1

Category: Detect > Detection services

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: guardduty-enabled-centralized

Schedule type: Periodic

Parameters: None

This control checks whether Amazon GuardDuty is enabled in your GuardDuty account and Region.

It is highly recommended that you enable GuardDuty in all supported AWS Regions. Doing so allows GuardDuty to generate findings about unauthorized or unusual activity, even in Regions that you do not actively use. This also allows GuardDuty to monitor CloudTrail events for global AWS services such as IAM.

Remediation

To enable GuardDuty, see Getting started with GuardDuty in the *Amazon GuardDuty User Guide*.

[GuardDuty.2] GuardDuty filters should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::GuardDuty::Filter`

AWS Config rule: `tagged-guardduty-filter` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon GuardDuty filter has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the filter doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the filter isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a GuardDuty filter, see TagResource in the *Amazon GuardDuty API Reference*.

[GuardDuty.3] GuardDuty IPSets should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::GuardDuty::IPSet`

AWS Config rule: `tagged-guardduty-ipset` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon GuardDuty IPSet has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the IPSet doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the IPSet isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a GuardDuty IPSet, see TagResource in the *Amazon GuardDuty API Reference*.

[GuardDuty.4] GuardDuty detectors should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: `tagged-guardduty-detector` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon GuardDuty detector has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the detector doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the detector isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a GuardDuty detector, see TagResource in the *Amazon GuardDuty API Reference*.

[GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled

Category: Detect > Detection services

Severity: High

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: guardduty-eks-protection-audit-enabled

Schedule type: Periodic

Parameters: None

This control checks whether GuardDuty EKS Audit Log Monitoring is enabled. For a standalone account, the control fails if GuardDuty EKS Audit Log Monitoring is disabled in the account. In a multi-account environment, the control fails if the delegated GuardDuty administrator account and all member accounts don't have EKS Audit Log Monitoring enabled.

In a multi-account environment, the control generates findings in only the delegated GuardDuty administrator account. Only the delegated administrator can enable or disable the EKS Audit Log Monitoring feature for the member accounts in the organization. GuardDuty member accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated GuardDuty administrator has a suspended member account that doesn't have GuardDuty EKS Audit Log Monitoring enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in GuardDuty.

GuardDuty EKS Audit Log Monitoring helps you detect potentially suspicious activities in your Amazon Elastic Kubernetes Service (Amazon EKS) clusters. EKS Audit Log Monitoring uses Kubernetes audit logs to capture chronological activities from users, applications using the Kubernetes API, and the control plane.

Remediation

To enable GuardDuty EKS Audit Log Monitoring, see EKS Audit Log Monitoring in the *Amazon GuardDuty User Guide*.

[GuardDuty.6] GuardDuty Lambda Protection should be enabled

Related requirements: PCI DSS v4.0.1/11.5.1

Category: Detect > Detection services

Severity: High

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: guardduty-lambda-protection-enabled

Schedule type: Periodic

Parameters: None

This control checks whether GuardDuty Lambda Protection is enabled. For a standalone account, the control fails if GuardDuty Lambda Protection is disabled in the account. In a multi-account environment, the control fails if the delegated GuardDuty administrator account and all member accounts don't have Lambda Protection enabled.

In a multi-account environment, the control generates findings in only the delegated GuardDuty administrator account. Only the delegated administrator can enable or disable the Lambda Protection feature for the member accounts in the organization. GuardDuty member accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated GuardDuty administrator has a suspended member account that doesn't have GuardDuty Lambda Protection enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in GuardDuty.

GuardDuty Lambda Protection helps you identify potential security threats when an AWS Lambda function gets invoked. After your enable Lambda Protection, GuardDuty starts monitoring Lambda network activity logs associated with the Lambda functions in your AWS account. When a Lambda function gets invoked and GuardDuty identifies suspicious network traffic that indicates the presence of a potentially malicious piece of code in your Lambda function, GuardDuty generates a finding.

Remediation

To enable GuardDuty Lambda Protection, see Configuring Lambda Protection in the *Amazon GuardDuty User Guide*.

[GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled

Related requirements: PCI DSS v4.0.1/11.5.1

Category: Detect > Detection Services

Severity: Medium

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: guardduty-eks-protection-runtime-enabled

Schedule type: Periodic

Parameters: None

This control checks whether GuardDuty EKS Runtime Monitoring with automated agent management is enabled. For a standalone account, the control fails if GuardDuty EKS Runtime Monitoring with automated agent management is disabled in the account. In a multi-account environment, the control fails if the delegated GuardDuty administrator account and all member accounts don't have EKS Runtime Monitoring with automated agent management enabled.

In a multi-account environment, the control generates findings in only the delegated GuardDuty administrator account. Only the delegated administrator can enable or disable the EKS Runtime Monitoring feature with automated agent management for the member accounts in the organization. GuardDuty member accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated GuardDuty administrator has a suspended member account that doesn't have GuardDuty EKS Runtime Monitoring enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in GuardDuty.

EKS Protection in Amazon GuardDuty provides threat detection coverage to help you protect Amazon EKS clusters within your AWS environment. EKS Runtime Monitoring uses operating system-level events to help you detect potential threats in EKS nodes and containers within your EKS clusters.

Remediation

To enable EKS Runtime Monitoring with automated agent management, see Enabling GuardDuty Runtime Monitoring in the *Amazon GuardDuty User Guide*.

[GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled

Category: Detect > Detection services

Severity: High

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: guardduty-malware-protection-enabled

Schedule type: Periodic

Parameters: None

This control checks whether GuardDuty Malware Protection is enabled. For a standalone account, the control fails if GuardDuty Malware Protection is disabled in the account. In a multi-account environment, the control fails if the delegated GuardDuty administrator account and all member accounts don't have Malware Protection enabled.

In a multi-account environment, the control generates findings in only the delegated GuardDuty administrator account. Only the delegated administrator can enable or disable the Malware Protection feature for the member accounts in the organization. GuardDuty member accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated GuardDuty administrator has a suspended member account that doesn't have GuardDuty Malware Protection enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in GuardDuty.

GuardDuty Malware Protection for EC2 helps you detect the potential presence of malware by scanning the Amazon Elastic Block Store (Amazon EBS) volumes that are attached to Amazon Elastic Compute Cloud (Amazon EC2) instances and container workloads. Malware Protection provides scan options where you can decide if you want to include or exclude specific EC2 instances and container workloads at the time of scanning. It also provides an option to retain the snapshots of EBS volumes attached to the EC2 instances or container workloads, in your GuardDuty accounts. The snapshots get retained only when malware is found and Malware Protection findings are generated.

Remediation

To enable GuardDuty Malware Protection for EC2, see Configuring GuardDuty-initiated malware scan in the *Amazon GuardDuty User Guide*.

[GuardDuty.9] GuardDuty RDS Protection should be enabled

Related requirements: PCI DSS v4.0.1/11.5.1

Category: Detect > Detection services

Severity: High

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: guardduty-rds-protection-enabled

Schedule type: Periodic

Parameters: None

This control checks whether GuardDuty RDS Protection is enabled. For a standalone account, the control fails if GuardDuty RDS Protection is disabled in the account. In a multi-account environment, the control fails if the delegated GuardDuty administrator account and all member accounts don't have RDS Protection enabled.

In a multi-account environment, the control generates findings in only the delegated GuardDuty administrator account. Only the delegated administrator can enable or disable the RDS Protection feature for the member accounts in the organization. GuardDuty member accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated GuardDuty administrator has a suspended member account that doesn't have GuardDuty RDS Protection enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in GuardDuty.

RDS Protection in GuardDuty analyzes and profiles RDS login activity for potential access threats to your Amazon Aurora databases (Aurora MySQL-Compatible Edition and Aurora PostgreSQL-Compatible Edition). This feature allows you to identify potentially suspicious login behavior. RDS Protection doesn't require additional infrastructure; it is designed so as not to affect the performance of your database instances. When RDS Protection detects a potentially suspicious or anomalous login attempt that indicates a threat to your database, GuardDuty generates a new finding with details about the potentially compromised database.

Remediation

To enable GuardDuty RDS Protection, see GuardDuty RDS Protection in the *Amazon GuardDuty User Guide*.

[GuardDuty.10] GuardDuty S3 Protection should be enabled

Related requirements: PCI DSS v4.0.1/11.5.1

Category: Detect > Detection services

Severity: High

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: guardduty-s3-protection-enabled

Schedule type: Periodic

Parameters: None

This control checks whether GuardDuty S3 Protection is enabled. For a standalone account, the control fails if GuardDuty S3 Protection is disabled in the account. In a multi-account environment, the control fails if the delegated GuardDuty administrator account and all member accounts don't have S3 Protection enabled.

In a multi-account environment, the control generates findings in only the delegated GuardDuty administrator account. Only the delegated administrator can enable or disable the S3 Protection feature for the member accounts in the organization. GuardDuty member accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated GuardDuty administrator has a suspended member account that doesn't have GuardDuty S3 Protection enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in GuardDuty.

S3 Protection enables GuardDuty to monitor object-level API operations to identify potential security risks for data within your Amazon Simple Storage Service (Amazon S3) buckets. GuardDuty monitors threats against your S3 resources by analyzing AWS CloudTrail management events and CloudTrail S3 data events.

Remediation

To enable GuardDuty S3 Protection, see Amazon S3 Protection in Amazon GuardDuty in the *Amazon GuardDuty User Guide*.

[GuardDuty.11] GuardDuty Runtime Monitoring should be enabled

Category: Detect > Detection Services

Severity: High

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: guardduty-runtime-monitoring-enabled

Schedule type: Periodic

Parameters: None

This control checks whether Runtime Monitoring is enabled in Amazon GuardDuty. For a standalone account, the control fails if GuardDuty Runtime Monitoring is disabled for the account. In a multi-account environment, the control fails if GuardDuty Runtime Monitoring is disabled for the delegated GuardDuty administrator account and all member accounts.

In a multi-account environment, only the delegated GuardDuty administrator can enable or disable GuardDuty Runtime Monitoring for accounts in their organization. In addition, only the GuardDuty administrator can configure and manage the security agents that GuardDuty uses for runtime monitoring of AWS workloads and resources for accounts in the organization. GuardDuty member accounts can't enable, configure, or disable Runtime Monitoring for their own accounts.

GuardDuty Runtime Monitoring observes and analyzes operating system-level, networking, and file events to help you detect potential threats in specific AWS workloads in your environment. It uses GuardDuty security agents that add visibility into runtime behavior, such as file access, process execution, command line arguments, and network connections. You can enable and manage the security agent for each type of resource that you want to monitor for potential threats, such as Amazon EKS clusters and Amazon EC2 instances.

Remediation

For information about configuring and enabling GuardDuty Runtime Monitoring, see GuardDuty Runtime Monitoring and Enabling GuardDuty Runtime Monitoring in the *Amazon GuardDuty User Guide*.

[GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled

Category: Detect > Detection Services

Severity: Medium

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: guardduty-ecs-protection-runtime-enabled

Schedule type: Periodic

Parameters: None

This control checks whether the Amazon GuardDuty automated security agent is enabled for runtime monitoring of Amazon ECS clusters on AWS Fargate. For a standalone account, the control fails if the security agent is disabled for the account. In a multi-account environment, the control

fails if the security agent is disabled for the delegated GuardDuty administrator account and all member accounts.

In a multi-account environment, this control generates findings only in the delegated GuardDuty administrator account. This is because only the delegated GuardDuty administrator can enable or disable Runtime Monitoring of ECS-Fargate resources for accounts in their organization. GuardDuty member accounts can't do this for their own accounts. In addition, this control generates `FAILED` findings if GuardDuty is suspended for a member account and Runtime Monitoring of ECS-Fargate resources is disabled for the member account. To receive a `PASSED` finding, the GuardDuty administrator must disassociate the suspended member account from their administrator account by using GuardDuty.

GuardDuty Runtime Monitoring observes and analyzes operating system-level, networking, and file events to help you detect potential threats in specific AWS workloads in your environment. It uses GuardDuty security agents that add visibility into runtime behavior, such as file access, process execution, command line arguments, and network connections. You can enable and manage the security agent for each type of resource that you want to monitor for potential threats. This includes Amazon ECS clusters on AWS Fargate.

Remediation

To enable and manage the security agent for GuardDuty Runtime Monitoring of ECS-Fargate resources, you must use GuardDuty directly. You can't enable or manage it manually for ECS-Fargate resources. For information about enabling and managing the security agent, see Prerequisites for AWS Fargate (Amazon ECS only) support and Managing the automated security agent for AWS Fargate (Amazon ECS only) in the *Amazon GuardDuty User Guide*.

[GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled

Category: Detect > Detection Services

Severity: Medium

Resource type: `AWS::GuardDuty::Detector`

AWS Config rule: guardduty-ec2-protection-runtime-enabled

Schedule type: Periodic

Parameters: None

This control checks whether the Amazon GuardDuty automated security agent is enabled for runtime monitoring of Amazon EC2 instances. For a standalone account, the control fails if the security agent is disabled for the account. In a multi-account environment, the control fails if the security agent is disabled for the delegated GuardDuty administrator account and all member accounts.

In a multi-account environment, this control generates findings only in the delegated GuardDuty administrator account. This is because only the delegated GuardDuty administrator can enable or disable Runtime Monitoring of Amazon EC2 instances for accounts in their organization. GuardDuty member accounts can't do this for their own accounts. In addition, this control generates `FAILED` findings if GuardDuty is suspended for a member account and Runtime Monitoring of EC2 instances is disabled for the member account. To receive a `PASSED` finding, the GuardDuty administrator must disassociate the suspended member account from their administrator account by using GuardDuty.

GuardDuty Runtime Monitoring observes and analyzes operating system-level, networking, and file events to help you detect potential threats in specific AWS workloads in your environment. It uses GuardDuty security agents that add visibility into runtime behavior, such as file access, process execution, command line arguments, and network connections. You can enable and manage the security agent for each type of resource that you want to monitor for potential threats. This includes Amazon EC2 instances.

Remediation

For information about configuring and managing the automated security agent for GuardDuty Runtime Monitoring of EC2 instances, see Prerequisites for Amazon EC2 instance support and Enabling the automated security agent for Amazon EC2 instances in the *Amazon GuardDuty User Guide*.

Security Hub controls for AWS Identity and Access Management

These AWS Security Hub controls evaluate the AWS Identity and Access Management (IAM) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[IAM.1] IAM policies should not allow full "*" administrative privileges

Related requirements: CIS AWS Foundations Benchmark v1.2.0/1.22, CIS AWS Foundations Benchmark v1.4.0/1.16, NIST.800-53.r5 AC-2, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3,

NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-6(10), NIST.800-53.r5 AC-6(2), NIST.800-53.r5 AC-6(3), NIST.800-171.r2 3.1.4, PCI DSS v3.2.1/7.2.1

Category: Protect > Secure access management

Severity: High

Resource type: `AWS::IAM::Policy`

AWS Config rule: `iam-policy-no-statements-with-admin-access`

Schedule type: Change triggered

Parameters:

- `excludePermissionBoundaryPolicy: true` (not customizable)

This control checks whether the default version of IAM policies (also known as customer managed policies) has administrator access by including a statement with `"Effect": "Allow"` with `"Action": "*"` over `"Resource": "*"`. The control fails if you have IAM policies with such a statement.

The control only checks the customer managed policies that you create. It does not check inline and AWS managed policies.

IAM policies define a set of privileges that are granted to users, groups, or roles. Following standard security advice, AWS recommends that you grant least privilege, which means to grant only the permissions that are required to perform a task. When you provide full administrative privileges instead of the minimum set of permissions that the user needs, you expose the resources to potentially unwanted actions.

Instead of allowing full administrative privileges, determine what users need to do and then craft policies that let the users perform only those tasks. It is more secure to start with a minimum set of permissions and grant additional permissions as necessary. Do not start with permissions that are too lenient and then try to tighten them later.

You should remove IAM policies that have a statement with `"Effect": "Allow"` with `"Action": "*"` over `"Resource": "*"`.

> **Note**
>
> AWS Config should be enabled in all Regions in which you use Security Hub. However, global resource recording can be enabled in a single Region. If you only record global resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

Remediation

To modify your IAM policies so that they do not allow full "*" administrative privileges, see Editing IAM policies in the *IAM User Guide*.

[IAM.2] IAM users should not have IAM policies attached

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.15, CIS AWS Foundations Benchmark v1.2.0/1.16, NIST.800-53.r5 AC-2, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-6(3), NIST.800-171.r2 3.1.1, NIST.800-171.r2 3.1.2, NIST.800-171.r2 3.1.7, NIST.800-171.r2 3.3.9, NIST.800-171.r2 3.13.3, PCI DSS v3.2.1/7.2.1

Category: Protect > Secure access management

Severity: Low

Resource type: `AWS::IAM::User`

AWS Config rule: `iam-user-no-policies-check`

Schedule type: Change triggered

Parameters: None

This control checks whether your IAM users have policies attached. The control fails if your IAM users have policies attached. Instead, IAM users must inherit permissions from IAM groups or assume a role.

By default, IAM users, groups, and roles have no access to AWS resources. IAM policies grant privileges to users, groups, or roles. We recommend that you apply IAM policies directly to groups and roles but not to users. Assigning privileges at the group or role level reduces the complexity of

access management as the number of users grows. Reducing access management complexity might in turn reduce the opportunity for a principal to inadvertently receive or retain excessive privileges.

> **Note**
>
> AWS Config should be enabled in all Regions in which you use Security Hub. However, global resource recording can be enabled in a single Region. If you only record global resources in a single Region, you can disable this control in all Regions except the Region where you record global resources.

Remediation

To resolve this issue, create an IAM group, and attach the policy to the group. Then, add the users to the group. The policy is applied to each user in the group. To remove a policy attached directly to a user, see Adding and removing IAM identity permissions in the *IAM User Guide*.

[IAM.3] IAM users' access keys should be rotated every 90 days or less

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.14, CIS AWS Foundations Benchmark v1.4.0/1.14, CIS AWS Foundations Benchmark v1.2.0/1.4, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-2(3), NIST.800-53.r5 AC-3(15), PCI DSS v4.0.1/8.3.9, PCI DSS v4.0.1/8.6.3

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::IAM::User`

AWS Config rule: `access-keys-rotated`

Schedule type: Periodic

Parameters:

- `maxAccessKeyAge`: `90` (not customizable)

This control checks whether the active access keys are rotated within 90 days.

We highly recommend that you do not generate and remove all access keys in your account. Instead, the recommended best practice is to either create one or more IAM roles or to use

federation through AWS IAM Identity Center. You can use these methods to allow your users to access the AWS Management Console and AWS CLI.

Each approach has its use cases. Federation is generally better for enterprises that have an existing central directory or plan to need more than the current limit on IAM users. Applications that run outside of an AWS environment need access keys for programmatic access to AWS resources.

However, if the resources that need programmatic access run inside AWS, the best practice is to use IAM roles. Roles allow you to grant a resource access without hardcoding an access key ID and secret access key into the configuration.

To learn more about protecting your access keys and account, see Best practices for managing AWS access keys in the *AWS General Reference*. Also see the blog post Guidelines for protecting your AWS account while using programmatic access.

If you already have an access key, Security Hub recommends that you rotate the access keys every 90 days. Rotating access keys reduces the chance that an access key that is associated with a compromised or terminated account is used. It also ensures that data cannot be accessed with an old key that might have been lost, cracked, or stolen. Always update your applications after you rotate access keys.

Access keys consist of an access key ID and a secret access key. They are used to sign programmatic requests that you make to AWS. Users need their own access keys to make programmatic calls to AWS from the AWS CLI, Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the API operations for individual AWS services.

If your organization uses AWS IAM Identity Center (IAM Identity Center), your users can sign in to Active Directory, a built-in IAM Identity Center directory, or another identity provider (IdP) connected to IAM Identity Center. They can then be mapped to an IAM role that enables them to run AWS CLI commands or call AWS API operations without the need for access keys. To learn more, see Configuring the AWS CLI to use AWS IAM Identity Center in the *AWS Command Line Interface User Guide*.

Note

AWS Config should be enabled in all Regions in which you use Security Hub. However, global resource recording can be enabled in a single Region. If you only record global resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

Remediation

To rotate access keys that are older than 90 days, see Rotating access keys in the *IAM User Guide*. Follow the instructions for any user with an **Access key age** greater than 90 days.

[IAM.4] IAM root user access key should not exist

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.4, CIS AWS Foundations Benchmark v1.4.0/1.4, CIS AWS Foundations Benchmark v1.2.0/1.12, PCI DSS v3.2.1/2.1, PCI DSS v3.2.1/2.2, PCI DSS v3.2.1/7.2.1, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-6(10), NIST.800-53.r5 AC-6(2)

Category: Protect > Secure access management

Severity: Critical

Resource type: `AWS::::Account`

AWS Config rule: `iam-root-access-key-check`

Schedule type: Periodic

Parameters: None

This control checks whether the root user access key is present.

The root user is the most privileged user in an AWS account. AWS access keys provide programmatic access to a given account.

Security Hub recommends that you remove all access keys that are associated with the root user. This limits that vectors that can be used to compromise your account. It also encourages the creation and use of role-based accounts that are least privileged.

Remediation

To delete the root user access key, see Deleting access keys for the root user in the *IAM User Guide*. To delete the root user access keys from an AWS account in AWS GovCloud (US), see Deleting my AWS GovCloud (US) account root user access keys in the *AWS GovCloud (US) User Guide*.

[IAM.5] MFA should be enabled for all IAM users that have a console password

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.10, CIS AWS Foundations Benchmark v1.4.0/1.10, CIS AWS Foundations Benchmark v1.2.0/1.2, NIST.800-53.r5 AC-2(1),

NIST.800-53.r5 AC-3(15), NIST.800-53.r5 IA-2(1), NIST.800-53.r5 IA-2(2), NIST.800-53.r5 IA-2(6), NIST.800-53.r5 IA-2(8), PCI DSS v4.0.1/8.4.2

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::IAM::User`

AWS Config rule: `mfa-enabled-for-iam-console-access`

Schedule type: Periodic

Parameters: None

This control checks whether AWS multi-factor authentication (MFA) is enabled for all IAM users that use a console password.

Multi-factor authentication (MFA) adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they are prompted for their user name and password. In addition, they are prompted for an authentication code from their AWS MFA device.

We recommend that you enable MFA for all accounts that have a console password. MFA is designed to provide increased security for console access. The authenticating principal must possess a device that emits a time-sensitive key and must have knowledge of a credential.

> **Note**
>
> AWS Config should be enabled in all Regions in which you use Security Hub. However, global resource recording can be enabled in a single Region. If you only record global resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

Remediation

To add MFA for IAM users, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

We are offering a free MFA security key to eligible customers. See if you qualify, and order your free key.

[IAM.6] Hardware MFA should be enabled for the root user

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.6, CIS AWS Foundations Benchmark v1.4.0/1.6, CIS AWS Foundations Benchmark v1.2.0/1.14, PCI DSS v3.2.1/8.3.1, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3(15), NIST.800-53.r5 IA-2(1), NIST.800-53.r5 IA-2(2), NIST.800-53.r5 IA-2(6), NIST.800-53.r5 IA-2(8), PCI DSS v4.0.1/8.4.2

Category: Protect > Secure access management

Severity: Critical

Resource type: `AWS::::Account`

AWS Config rule: `root-account-hardware-mfa-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether your AWS account is enabled to use a hardware multi-factor authentication (MFA) device to sign in with root user credentials. The control fails if hardware MFA isn't enabled or virtual MFA devices are permitted for signing in with root user credentials.

Virtual MFA might not provide the same level of security as hardware MFA devices. We recommend that you use a virtual MFA device only while you wait for hardware purchase approval or for your hardware to arrive. To learn more, see Assign a virtual MFA device (console) in the *IAM User Guide*.

Note

Security Hub evaluates this control based on the presence of root user credentials (login profile) in an AWS account. The control generates `PASSED` findings in the following cases:

- Root user credentials are present in the account and hardware MFA is enabled for the root user.
- Root user credentials aren't present in the account.

The control generates a `FAILED` finding if root user credentials are present in the account and hardware MFA is not enabled for the root user.

Remediation

For information about enabling hardware MFA for the root user, see Multi-factor authentication for an AWS account root user in the *IAM User Guide*.

We are offering a free MFA security key to eligible customers. To determine whether you're eligible, see the MFA Security Key Program FAQs.

[IAM.7] Password policies for IAM users should have strong configurations

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-2(3), NIST.800-53.r5 AC-3(15), NIST.800-53.r5 IA-5(1), NIST.800-171.r2 3.5.2, NIST.800-171.r2 3.5.7, NIST.800-171.r2 3.5.8, PCI DSS v4.0.1/8.3.6, PCI DSS v4.0.1/8.3.7, PCI DSS v4.0.1/8.3.9, PCI DSS v4.0.1/8.3.10.1, PCI DSS v4.0.1/8.6.3

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`RequireUppercaseCharacters`	Require at least one uppercase character in password	Boolean	`true` or `false`	`true`
`RequireLowercaseCharacters`	Require at least one lowercase character in password	Boolean	`true` or `false`	`true`
`RequireSymbols`	Require at least one symbol in password	Boolean	`true` or `false`	`true`

Parameter	Description	Type	Allowed custom values	Security Hub default value
`RequireNumbers`	Require at least one number in password	Boolean	`true` or `false`	`true`
`MinimumPasswordLength`	Minimum number of characters in the password	Integer	8 to 128	8
`PasswordReusePrevention`	Number of password rotations before an old password can be reused	Integer	12 to 24	No default value
`MaxPasswordAge`	Number of days before password expiration	Integer	1 to 90	No default value

This control checks whether the account password policy for IAM users uses strong configurations. The control fails if the password policy doesn't use strong configurations. Unless you provide custom parameter values, Security Hub uses the default values mentioned in the preceding table. The `PasswordReusePrevention` and `MaxPasswordAge` parameters have no default value, so if you exclude these parameters, Security Hub ignores number of password rotations and password age when evaluating this control.

To access the AWS Management Console, IAM users need passwords. As a best practice, Security Hub highly recommends that instead of creating IAM users, you use federation. Federation allows users to use their existing corporate credentials to log into the AWS Management Console. Use AWS IAM Identity Center (IAM Identity Center) to create or federate the user, and then assume an IAM role into an account.

To learn more about identity providers and federation, see Identity providers and federation in the *IAM User Guide*. To learn more about IAM Identity Center, see the *AWS IAM Identity Center User Guide*.

If you need to use IAM users, Security Hub recommends that you enforce the creation of strong user passwords. You can set a password policy on your AWS account to specify complexity requirements and mandatory rotation periods for passwords. When you create or change a

password policy, most of the password policy settings are enforced the next time users change their passwords. Some of the settings are enforced immediately.

Remediation

To update your password policy, see Setting an account password policy for IAM users in the *IAM User Guide*.

[IAM.8] Unused IAM user credentials should be removed

Related requirements: CIS AWS Foundations Benchmark v1.2.0/1.3, NIST.800-53.r5 AC-2, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-2(3), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6, NIST.800-171.r2 3.1.2, PCI DSS v3.2.1/8.1.4, PCI DSS v4.0.1/8.2.6

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::IAM::User`

AWS Config rule: `iam-user-unused-credentials-check`

Schedule type: Periodic

Parameters:

- `maxCredentialUsageAge`: `90` (not customizable)

This control checks whether your IAM users have passwords or active access keys that have not been used for 90 days.

IAM users can access AWS resources using different types of credentials, such as passwords or access keys.

Security Hub recommends that you remove or deactivate all credentials that were unused for 90 days or more. Disabling or removing unnecessary credentials reduces the window of opportunity for credentials associated with a compromised or abandoned account to be used.

> **Note**
>
> AWS Config should be enabled in all Regions in which you use Security Hub. However, global resource recording can be enabled in a single Region. If you only record global

resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

Remediation

When you view user information in the IAM console, there are columns for **Access key age**, **Password age**, and **Last activity**. If the value in any of these columns is greater than 90 days, make the credentials for those users inactive.

You can also use credential reports to monitor users and identify those with no activity for 90 or more days. You can download credential reports in `.csv` format from the IAM console.

After you identify the inactive accounts or unused credentials, deactivate them. For instructions, see Creating, changing, or deleting an IAM user password (console) in the *IAM User Guide*.

[IAM.9] MFA should be enabled for the root user

Related requirements: PCI DSS v3.2.1/8.3.1, PCI DSS v4.0.1/8.4.2, CIS AWS Foundations Benchmark v3.0.0/1.5, CIS AWS Foundations Benchmark v1.4.0/1.5, CIS AWS Foundations Benchmark v1.2.0/1.13, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3(15), NIST.800-53.r5 IA-2(1), NIST.800-53.r5 IA-2(2), NIST.800-53.r5 IA-2(6), NIST.800-53.r5 IA-2(8)

Category: Protect > Secure access management

Severity: Critical

Resource type: `AWS::::Account`

AWS Config rule: `root-account-mfa-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether multi-factor authentication (MFA) is enabled for the IAM root user of an AWS account to sign in to the AWS Management Console. The control fails if MFA isn't enabled for the root user of the account.

The IAM root user of an AWS account has complete access to all the services and resources in the account. If MFA is enabled, the user must enter a username, a password, and an authentication

code from their AWS MFA device in order to sign in to the AWS Management Console. MFA adds an extra layer of protection on top of a username and password.

This control generates PASSED findings in the following cases:

- Root user credentials are present in the account and MFA is enabled for the root user.
- Root user credentials aren't present in the account.

The control generates FAILED findings if root user credentials are present in the account and MFA isn't enabled for the root user.

Remediation

For information about enabling MFA for the root user of an AWS account, see Multi-factor authentication for the AWS account root user in the *AWS Identity and Access Management User Guide*.

[IAM.10] Password policies for IAM users should have strong configurations

Related requirements: NIST.800-171.r2 3.5.2, NIST.800-171.r2 3.5.7, NIST.800-171.r2 3.5.8, PCI DSS v3.2.1/8.1.4, PCI DSS v3.2.1/8.2.3, PCI DSS v3.2.1/8.2.4, PCI DSS v3.2.1/8.2.5

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters: None

This control checks whether the account password policy for IAM users uses the following minimum PCI DSS configurations.

- `RequireUppercaseCharacters` – Require at least one uppercase character in password. (Default = `true`)
- `RequireLowercaseCharacters` – Require at least one lowercase character in password. (Default = `true`)

- `RequireNumbers` – Require at least one number in password. (Default = `true`)
- `MinimumPasswordLength` – Password minimum length. (Default = 7 or longer)
- `PasswordReusePrevention` – Number of passwords before allowing reuse. (Default = 4)
- `MaxPasswordAge` – Number of days before password expiration. (Default = 90)

> **Note**
>
> On May 30, 2025, Security Hub removed this control from the PCI DSS v4.0.1 standard. PCI DSS v4.0.1 now requires passwords to have a minimum of 8 characters. This control continues to apply to the PCI DSS v3.2.1 standard, which has different password requirements.
> To evaluate account password policies against PCI DSS v4.0.1 requirements, you can use the IAM.7 control. This control requires passwords to have a minimum of 8 characters. It also supports custom values for password length and other parameters. The IAM.7 control is part of the PCI DSS v4.0.1 standard in Security Hub.

Remediation

To update your password policy to use the recommended configuration, see Setting an account password policy for IAM users in the *IAM User Guide*.

[IAM.11] Ensure IAM password policy requires at least one uppercase letter

Related requirements: CIS AWS Foundations Benchmark v1.2.0/1.5, NIST.800-171.r2 3.5.7, PCI DSS v4.0.1/8.3.6, PCI DSS v4.0.1/8.6.3

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters: None

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords use different character sets.

CIS recommends that the password policy require at least one uppercase letter. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To change your password policy, see Setting an account password policy for IAM users in the *IAM User Guide*. For **Password strength**, select **Require at least one uppercase letter from the Latin alphabet (A–Z)**.

[IAM.12] Ensure IAM password policy requires at least one lowercase letter

Related requirements: CIS AWS Foundations Benchmark v1.2.0/1.6, NIST.800-171.r2 3.5.7, PCI DSS v4.0.1/8.3.6, PCI DSS v4.0.1/8.6.3

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters: None

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords use different character sets. CIS recommends that the password policy require at least one lowercase letter. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To change your password policy, see Setting an account password policy for IAM users in the *IAM User Guide*. For **Password strength**, select **Require at least one lowercase letter from the Latin alphabet (A–Z)**.

[IAM.13] Ensure IAM password policy requires at least one symbol

Related requirements: CIS AWS Foundations Benchmark v1.2.0/1.7, NIST.800-171.r2 3.5.7

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters: None

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords use different character sets.

CIS recommends that the password policy require at least one symbol. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To change your password policy, see Setting an account password policy for IAM users in the *IAM User Guide*. For **Password strength**, select **Require at least one nonalphanumeric character**.

[IAM.14] Ensure IAM password policy requires at least one number

Related requirements: CIS AWS Foundations Benchmark v1.2.0/1.8, NIST.800-171.r2 3.5.7, PCI DSS v4.0.1/8.3.6, PCI DSS v4.0.1/8.6.3

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters: None

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords use different character sets.

CIS recommends that the password policy require at least one number. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To change your password policy, see Setting an account password policy for IAM users in the *IAM User Guide*. For **Password strength**, select **Require at least one number**.

[IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.8, CIS AWS Foundations Benchmark v1.4.0/1.8, CIS AWS Foundations Benchmark v1.2.0/1.9, NIST.800-171.r2 3.5.7

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters: None

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords are at least a given length.

CIS recommends that the password policy require a minimum password length of 14 characters. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To change your password policy, see Setting an account password policy for IAM users in the *IAM User Guide*. For **Password minimum length**, enter **14** or a larger number.

[IAM.16] Ensure IAM password policy prevents password reuse

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.9, CIS AWS Foundations Benchmark v1.4.0/1.9, CIS AWS Foundations Benchmark v1.2.0/1.10, NIST.800-171.r2 3.5.8, PCI DSS v4.0.1/8.3.7

Category: Protect > Secure access management

Severity: Low

Resource type: `AWS::::Account`

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters: None

This control checks whether the number of passwords to remember is set to 24. The control fails if the value is not 24.

IAM password policies can prevent the reuse of a given password by the same user.

CIS recommends that the password policy prevent the reuse of passwords. Preventing password reuse increases account resiliency against brute force login attempts.

Remediation

To change your password policy, see Setting an account password policy for IAM users in the *IAM User Guide*. For **Prevent password reuse**, enter **24**.

[IAM.17] Ensure IAM password policy expires passwords within 90 days or less

Related requirements: CIS AWS Foundations Benchmark v1.2.0/1.11, PCI DSS v4.0.1/8.3.9, PCI DSS v4.0.1/8.3.10.1

Category: Protect > Secure access management

Severity: Low

Resource type: `AWS::::Account`

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters: None

IAM password policies can require passwords to be rotated or expired after a given number of days.

CIS recommends that the password policy expire passwords after 90 days or less. Reducing the password lifetime increases account resiliency against brute force login attempts. Requiring regular password changes also helps in the following scenarios:

- Passwords can be stolen or compromised without your knowledge. This can happen via a system compromise, software vulnerability, or internal threat.
- Certain corporate and government web filters or proxy servers can intercept and record traffic even if it's encrypted.
- Many people use the same password for many systems such as work, email, and personal.
- Compromised end-user workstations might have a keystroke logger.

Remediation

To change your password policy, see Setting an account password policy for IAM users in the *IAM User Guide*. For **Turn on password expiration**, enter **90** or a smaller number.

[IAM.18] Ensure a support role has been created to manage incidents with AWS Support

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.17, CIS AWS Foundations Benchmark v1.4.0/1.17, CIS AWS Foundations Benchmark v1.2.0/1.20, NIST.800-171.r2 3.1.2, PCI DSS v4.0.1/12.10.3

Category: Protect > Secure access management

Severity: Low

Resource type: `AWS::::Account`

AWS Config rule: `iam-policy-in-use`

Schedule type: Periodic

Parameters:

- `policyARN`: `arn:partition:iam::aws:policy/AWSSupportAccess` (not customizable)
- `policyUsageType`: ANY (not customizable)

AWS provides a support center that can be used for incident notification and response, as well as technical support and customer services.

Create an IAM role to allow authorized users to manage incidents with AWS Support. By implementing least privilege for access control, an IAM role will require an appropriate IAM policy to allow support center access in order to manage incidents with Support.

> **Note**
>
> AWS Config should be enabled in all Regions in which you use Security Hub. However, global resource recording can be enabled in a single Region. If you only record global resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

Remediation

To remediate this issue, create a role to allow authorized users to manage Support incidents.

To create the role to use for Support access

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the IAM navigation pane, choose **Roles**, then choose **Create role**.
3. For **Role type**, choose the **Another AWS account**.
4. For **Account ID**, enter the AWS account ID of the AWS account to which you want to grant access to your resources.

   If the users or groups that will assume this role are in the same account, then enter the local account number.

   > **Note**
   >
   > The administrator of the specified account can grant permission to assume this role to any user in that account. To do this, the administrator attaches a policy to the user or a group that grants permission for the `sts:AssumeRole` action. In that policy, the resource must be the role ARN.

5. Choose **Next: Permissions**.
6. Search for the managed policy `AWSSupportAccess`.

7. Select the check box for the `AWSSupportAccess` managed policy.
8. Choose **Next: Tags**.
9. (Optional) To add metadata to the role, attach tags as key-value pairs.

   For more information about using tags in IAM, see Tagging IAM users and roles in the *IAM User Guide*.
10. Choose **Next: Review**.
11. For **Role name**, enter a name for your role.

    Role names must be unique within your AWS account. They are not case sensitive.
12. (Optional) For **Role description**, enter a description for the new role.
13. Review the role, then choose **Create role**.

[IAM.19] MFA should be enabled for all IAM users

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3(15), NIST.800-53.r5 IA-2(1), NIST.800-53.r5 IA-2(2), NIST.800-53.r5 IA-2(6), NIST.800-53.r5 IA-2(8), NIST.800-171.r2 3.3.8, NIST.800-171.r2 3.5.3, NIST.800-171.r2 3.5.4, NIST.800-171.r2 3.7.5, PCI DSS v3.2.1/8.3.1, PCI DSS v4.0.1/8.4.2,

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::IAM::User`

AWS Config rule: `iam-user-mfa-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether the IAM users have multi-factor authentication (MFA) enabled.

> **Note**
>
> AWS Config should be enabled in all Regions in which you use Security Hub. However, global resource recording can be enabled in a single Region. If you only record global

resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

Remediation

To add MFA for IAM users, see Enabling MFA devices for users in AWS in the *IAM User Guide*.

[IAM.20] Avoid the use of the root user

Important

Security Hub retired this control in April 2024. For more information, see Change log for Security Hub CSPM controls.

Related requirements: CIS AWS Foundations Benchmark v1.2.0/1.1

Category: Protect > Secure access management

Severity: Low

Resource type: `AWS::IAM::User`

AWS Config rule: `use-of-root-account-test` (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

This control checks whether an AWS account has restrictions on the usage of the root user. The control evaluates the following resources:

- Amazon Simple Notification Service (Amazon SNS) topics
- AWS CloudTrail trails
- Metric filters associated with the CloudTrail trails
- Amazon CloudWatch alarms based on the filters

This check results in a `FAILED` finding if one or more of the following statements is true:

- No CloudTrail trails exist in the account.
- A CloudTrail trail is enabled, but not configured with at-least one multi-Region trail that includes read and write management events.
- A CloudTrail trail is enabled, but not associated with a CloudWatch Logs log group.
- The exact metric filter prescribed by the Center for Internet Security (CIS) is not used. The prescribed metric filter is `'{$.userIdentity.type="Root" && $.userIdentity.invokedBy NOT EXISTS && $.eventType !="AwsServiceEvent"}'`.
- No CloudWatch alarms based on the metric filter exist in the account.
- CloudWatch alarms configured to send notification to the associated SNS topic don't trigger based on the alarm condition.
- The SNS topic doesn't comply with the constraints for sending a message to an SNS topic.
- The SNS topic doesn't have at least one subscriber.

This check results in a control status of `NO_DATA` if one or more of the following statements is true:

- A multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- A multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

This check results in a control status of `WARNING` if one or more of the following statements is true:

- The current account doesn't own the SNS topic referenced in the CloudWatch alarm.
- The current account doesn't have access to the SNS topic when invoking the `ListSubscriptionsByTopic` SNS API.

Note

We recommend using organization trails to log events from many accounts in an organization. Organization trails are multi-Region trails by default and can only be managed by the AWS Organizations management account or the CloudTrail delegated administrator account. Using an organization trail results in a control status of NO_DATA for controls evaluated in organization member accounts. In member accounts, Security Hub only generates findings for member-owned resources. Findings that pertain to organization

trails are generated in the resource owner's account. You can see these findings in your Security Hub delegated administrator account by using cross-Region aggregation.

As a best practice, use your root user credentials only when required to perform account and service management tasks. Apply IAM policies directly to groups and roles but not to users. For instructions on setting up an administrator for daily use, see Creating your first IAM admin user and group in the *IAM User Guide*.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. Create an Amazon SNS topic that receives all CIS alarms.

   Create at least one subscriber to the topic. For more information, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in the section called "[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events".

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.

5. Under **Define pattern**, do the following:

   a. Copy the following pattern and then paste it into the **Filter Pattern** field.

   ```
   {$.userIdentity.type="Root" && $.userIdentity.invokedBy NOT EXISTS &&
    $.eventType !="AwsServiceEvent"}
   ```

   b. Choose **Next**.

6. Under **Assign Metric**, do the following:

   a. In **Filter name**, enter a name for your metric filter.

   b. For **Metric Namespace**, enter **`LogMetrics`**.

      If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.

   c. For **Metric Name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.

   d. For **Metric value**, enter **1**.

   e. Choose **Next**.

7. Under **Review and create**, verify the information that you provided for the new metric filter. Then, choose **Create metric filter**.

8. In the navigation pane, choose **Log groups**, and then choose the filter you created under **Metric filters**.

9. Select the check box for the filter. Choose **Create alarm**.

10. Under **Specify metric and conditions**, do the following:

    a. Under **Conditions**, for **Threshold**, choose **Static**.

    b. For **Define the alarm condition**, choose **Greater/Equal**.

    c. For **Define the threshold value**, enter **1**.

    d. Choose **Next**.

11. Under **Configure actions**, do the following:

    a. Under **Alarm state trigger**, choose **In alarm**.

    b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.

    c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.

d. Choose **Next**.

12. Under **Add name and description**, enter a **Name** and **Description** for the alarm, such as **`CIS-1.1-RootAccountUsage`**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

[IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services

Related requirements: NIST.800-53.r5 AC-2, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-6(10), NIST.800-53.r5 AC-6(2), NIST.800-53.r5 AC-6(3), NIST.800-171.r2 3.1.1, NIST.800-171.r2 3.1.2, NIST.800-171.r2 3.1.5, NIST.800-171.r2 3.1.7, NIST.800-171.r2 3.3.8, NIST.800-171.r2 3.3.9, NIST.800-171.r2 3.13.3, NIST.800-171.r2 3.13.4

Category: Detect > Secure access management

Severity: Low

Resource type: `AWS::IAM::Policy`

AWS Config rule: `iam-policy-no-statements-with-full-access`

Schedule type: Change triggered

Parameters:

- `excludePermissionBoundaryPolicy`: `True` (not customizable)

This control checks whether the IAM identity-based policies that you create have Allow statements that use the * wildcard to grant permissions for all actions on any service. The control fails if any policy statement includes `"Effect": "Allow"` with `"Action": "Service:*"`.

For example, the following statement in a policy results in a failed finding.

```
"Statement": [
{
  "Sid": "EC2-Wildcard",
  "Effect": "Allow",
  "Action": "ec2:*",
```

```
  "Resource": "*"
}
```

The control also fails if you use `"Effect": "Allow"` with `"NotAction": "service:*"`. In that case, the `NotAction` element provides access to all of the actions in an AWS service, except for the actions specified in `NotAction`.

This control only applies to customer managed IAM policies. It does not apply to IAM policies that are managed by AWS.

When you assign permissions to AWS services, it is important to scope the allowed IAM actions in your IAM policies. You should restrict IAM actions to only those actions that are needed. This helps you to provision least privilege permissions. Overly permissive policies might lead to privilege escalation if the policies are attached to an IAM principal that might not require the permission.

In some cases, you might want to allow IAM actions that have a similar prefix, such as `DescribeFlowLogs` and `DescribeAvailabilityZones`. In these authorized cases, you can add a suffixed wildcard to the common prefix. For example, `ec2:Describe*`.

This control passes if you use a prefixed IAM action with a suffixed wildcard. For example, the following statement in a policy results in a passed finding.

```
"Statement": [
{
  "Sid": "EC2-Wildcard",
  "Effect": "Allow",
  "Action": "ec2:Describe*",
  "Resource": "*"
}
```

When you group related IAM actions in this way, you can also avoid exceeding the IAM policy size limits.

> **Note**
>
> AWS Config should be enabled in all Regions in which you use Security Hub. However, global resource recording can be enabled in a single Region. If you only record global resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

Remediation

To remediate this issue, update your IAM policies so that they do not allow full "*" administrative privileges. For details about how to edit an IAM policy, see Editing IAM policies in the *IAM User Guide*.

[IAM.22] IAM user credentials unused for 45 days should be removed

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.12, CIS AWS Foundations Benchmark v1.4.0/1.12, NIST.800-171.r2 3.1.2

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::IAM::User`

AWS Config rule: `iam-user-unused-credentials-check`

Schedule type: Periodic

Parameters: None

This control checks whether your IAM users have passwords or active access keys that have not been used for 45 days or more. To do so, it checks whether the `maxCredentialUsageAge` parameter of the AWS Config rule is equal to 45 or more.

Users can access AWS resources using different types of credentials, such as passwords or access keys.

CIS recommends that you remove or deactivate all credentials that have been unused for 45 days or more. Disabling or removing unnecessary credentials reduces the window of opportunity for credentials associated with a compromised or abandoned account to be used.

The AWS Config rule for this control uses the `GetCredentialReport` and `GenerateCredentialReport` API operations, which are only updated every four hours. Changes to IAM users can take up to four hours to be visible to this control.

> **Note**
>
> AWS Config should be enabled in all Regions in which you use Security Hub. However, you can enable recording of global resources in a single Region. If you only record global

resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

Remediation

When you view user information in the IAM console, there are columns for **Access key age**, **Password age**, and **Last activity**. If the value in any of these columns is greater than 45 days, make the credentials for those users inactive.

You can also use credential reports to monitor users and identify those with no activity for 45 or more days. You can download credential reports in `.csv` format from the IAM console.

After you identify the inactive accounts or unused credentials, deactivate them. For instructions, see Creating, changing, or deleting an IAM user password (console) in the *IAM User Guide*.

[IAM.23] IAM Access Analyzer analyzers should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AccessAnalyzer::Analyzer`

AWS Config rule: `tagged-accessanalyzer-analyzer` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an analyzer managed by AWS Identity and Access Management Access Analyzer (IAM Access Analyzer) has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the analyzer doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the analyzer isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an analyzer, see TagResource in the *AWS IAM Access Analyzer API Reference*.

[IAM.24] IAM roles should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IAM::Role`

AWS Config rule: `tagged-iam-role` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS Identity and Access Management (IAM) role has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the role doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the role isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an IAM role, see Tagging IAM resources in the *IAM User Guide*.

[IAM.25] IAM users should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IAM::User`

AWS Config rule: `tagged-iam-user` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS Identity and Access Management (IAM) user has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the user doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the user isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an IAM user, see Tagging IAM resources in the *IAM User Guide*.

[IAM.26] Expired SSL/TLS certificates managed in IAM should be removed

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.19

Category: Identify > Compliance

Severity: Medium

Resource type: `AWS::IAM::ServerCertificate`

AWS Config rule: `iam-server-certificate-expiration-check`

Schedule type: Periodic

Parameters: None

This controls checks whether an active SSL/TLS server certificate that is managed in IAM has expired. The control fails if the expired SSL/TLS server certificate isn't removed.

To enable HTTPS connections to your website or application in AWS, you need an SSL/TLS server certificate. You can use IAM or AWS Certificate Manager (ACM) to store and deploy server certificates. Use IAM as a certificate manager only when you must support HTTPS connections in an AWS Region that isn't supported by ACM. IAM securely encrypts your private keys and stores

the encrypted version in IAM SSL certificate storage. IAM supports deploying server certificates in all Regions, but you must obtain your certificate from an external provider for use with AWS. You can't upload an ACM certificate to IAM. Additionally, you can't manage your certificates from the IAM console. Removing expired SSL/TLS certificates eliminates the risk that an invalid certificate is deployed accidentally to a resource, which can damage the credibility of the underlying application or website.

Remediation

To remove a server certificate from IAM, see Managing server certificates in IAM in the *IAM User Guide*.

[IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.22

Category: Protect > Secure access management > Secure IAM policies

Severity: Medium

Resource type: `AWS::IAM::Role`, `AWS::IAM::User`, `AWS::IAM::Group`

AWS Config rule: `iam-policy-blacklisted-check`

Schedule type: Change triggered

Parameters:

- "policyArns": "arn:aws:iam::aws:policy/AWSCloudShellFullAccess,arn:aws-cn:iam::aws:policy/AWSCloudShellFullAccess, arn:aws-us-gov:iam::aws:policy/AWSCloudShellFullAccess"

This control checks whether an IAM identity (user, role, or group) has the AWS managed policy `AWSCloudShellFullAccess` attached. The control fails if an IAM identity has the `AWSCloudShellFullAccess` policy attached.

AWS CloudShell provides a convenient way to run CLI commands against AWS services. The AWS managed policy `AWSCloudShellFullAccess` provides full access to CloudShell, which allows file upload and download capability between a user's local system and the CloudShell environment. Within the CloudShell environment, a user has sudo permissions, and can access the internet. As a result, atttaching this managed policy to an IAM identity gives them the ability to install file

transfer software and move data from CloudShell to external internet servers. We recommend following the principle of least privilege and attaching narrower permissions to your IAM identities.

Remediation

To detach the `AWSCloudShellFullAccess` policy from an IAM identity, see Adding and removing IAM identity permissions in the *IAM User Guide*.

[IAM.28] IAM Access Analyzer external access analyzer should be enabled

Related requirements: CIS AWS Foundations Benchmark v3.0.0/1.20

Category: Detect > Detection services > Privileged usage monitoring

Severity: High

Resource type: `AWS::AccessAnalyzer::Analyzer`

AWS Config rule: `iam-external-access-analyzer-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether an AWS account has an IAM Access Analyzer external access analyzer enabled. The control fails if the account doesn't have an external access analyzer enabled in your currently selected AWS Region.

IAM Access Analyzer external access analyzers help identify resources, such as Amazon Simple Storage Service (Amazon S3) buckets or IAM roles, that are shared with an external entity. This helps you avoid unintended access to your resources and data. IAM Access Analyzer is Regional and must be enabled in each Region. To identify resources that are shared with external principals, an access analyzer uses logic-based reasoning to analyze resource-based policies in your AWS environment. When you create an external access analyzer, you can create and enable it for your entire organization or individual accounts.

> **Note**
>
> If an account is part of an organization in AWS Organizations, this control doesn't factor external access analyzers that specify the organization as the zone of trust and are enabled for the organization in the current Region. If your organization uses this type

of configuration, consider disabling this control for individual member accounts in your organization in the Region.

Remediation

For information about enabling an external access analyzer in a specific Region, see Getting started with IAM Access Analyzer in the *IAM User Guide*. You must enable an analyzer in each Region in which you want to monitor access to your resources.

Security Hub controls for Amazon Inspector

These AWS Security Hub controls evaluate the Amazon Inspector service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Inspector.1] Amazon Inspector EC2 scanning should be enabled

Related requirements: PCI DSS v4.0.1/11.3.1

Category: Detect > Detection services

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: `inspector-ec2-scan-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Inspector EC2 scanning is enabled. For a standalone account, the control fails if Amazon Inspector EC2 scanning is disabled in the account. In a multi-account environment, the control fails if the delegated Amazon Inspector administrator account and all member accounts don't have EC2 scanning enabled.

In a multi-account environment, the control generates findings in only the delegated Amazon Inspector administrator account. Only the delegated administrator can enable or disable the EC2 scanning feature for the member accounts in the organization. Amazon Inspector member

accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated administrator has a suspended member account that doesn't have Amazon Inspector EC2 scanning enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in Amazon Inspector.

Amazon Inspector EC2 scanning extracts metadata from your Amazon Elastic Compute Cloud (Amazon EC2) instance, and then compares this metadata against rules collected from security advisories to produce findings. Amazon Inspector scans instances for package vulnerabilities and network reachability issues. For information about supported operating systems, including which operating system can be scanned without an SSM agent, see Supported operating systems: Amazon EC2 scanning.

Remediation

To enable Amazon Inspector EC2 scanning, see Activating scans in the *Amazon Inspector User Guide*.

[Inspector.2] Amazon Inspector ECR scanning should be enabled

Related requirements: PCI DSS v4.0.1/11.3.1

Category: Detect > Detection services

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: `inspector-ecr-scan-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Inspector ECR scanning is enabled. For a standalone account, the control fails if Amazon Inspector ECR scanning is disabled in the account. In a multi-account environment, the control fails if the delegated Amazon Inspector administrator account and all member accounts don't have ECR scanning enabled.

In a multi-account environment, the control generates findings in only the delegated Amazon Inspector administrator account. Only the delegated administrator can enable or disable the ECR scanning feature for the member accounts in the organization. Amazon Inspector member

accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated administrator has a suspended member account that doesn't have Amazon Inspector ECR scanning enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in Amazon Inspector.

Amazon Inspector scans container images stored in Amazon Elastic Container Registry (Amazon ECR) for software vulnerabilities to generate package vulnerability findings. When you activate Amazon Inspector scans for Amazon ECR, you set Amazon Inspector as your preferred scanning service for your private registry. This replaces basic scanning, which is provided at no charge by Amazon ECR, with enhanced scanning, which is provided and billed through Amazon Inspector. Enhanced scanning gives you the benefit of vulnerability scanning for both operating system and programming language packages at the registry level. You can review findings discovered using enhanced scanning at the image level, for each layer of the image, on the Amazon ECR console. Additionally, you can review and work with these findings in other services not available for basic scanning findings, including AWS Security Hub and Amazon EventBridge.

Remediation

To enable Amazon Inspector ECR scanning, see Activating scans in the *Amazon Inspector User Guide*.

[Inspector.3] Amazon Inspector Lambda code scanning should be enabled

Related requirements: PCI DSS v4.0.1/6.2.4, PCI DSS v4.0.1/6.3.1

Category: Detect > Detection services

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: `inspector-lambda-code-scan-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Inspector Lambda code scanning is enabled. For a standalone account, the control fails if Amazon Inspector Lambda code scanning is disabled in the account. In a multi-account environment, the control fails if the delegated Amazon Inspector administrator account and all member accounts don't have Lambda code scanning enabled.

In a multi-account environment, the control generates findings in only the delegated Amazon Inspector administrator account. Only the delegated administrator can enable or disable the Lambda code scanning feature for the member accounts in the organization. Amazon Inspector member accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated administrator has a suspended member account that doesn't have Amazon Inspector Lambda code scanning enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in Amazon Inspector.

Amazon Inspector Lambda code scanning scans the custom application code within an AWS Lambda function for code vulnerabilities based on AWS security best practices. Lambda code scanning can detect injection flaws, data leaks, weak cryptography, or missing encryption in your code. This feature is available in specific AWS Regions only. You can activate Lambda code scanning together with Lambda standard scanning (see [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled).

Remediation

To enable Amazon Inspector Lambda code scanning, see Activating scans in the *Amazon Inspector User Guide*.

[Inspector.4] Amazon Inspector Lambda standard scanning should be enabled

Related requirements: PCI DSS v4.0.1/6.2.4, PCI DSS v4.0.1/6.3.1

Category: Detect > Detection services

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: `inspector-lambda-standard-scan-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Inspector Lambda standard scanning is enabled. For a standalone account, the control fails if Amazon Inspector Lambda standard scanning is disabled in the account. In a multi-account environment, the control fails if the delegated Amazon Inspector administrator account and all member accounts don't have Lambda standard scanning enabled.

In a multi-account environment, the control generates findings in only the delegated Amazon Inspector administrator account. Only the delegated administrator can enable or disable the Lambda standard scanning feature for the member accounts in the organization. Amazon Inspector member accounts can't modify this configuration from their accounts. This control generates `FAILED` findings if the delegated administrator has a suspended member account that doesn't have Amazon Inspector Lambda standard scanning enabled. To receive a `PASSED` finding, the delegated administrator must disassociate these suspended accounts in Amazon Inspector.

Amazon Inspector Lambda standard scanning identifies software vulnerabilities in the application package dependencies you add to your AWS Lambda function code and layers. If Amazon Inspector detects a vulnerability in your Lambda function application package dependencies, Amazon Inspector produces a detailed `Package Vulnerability` type finding. You can activate Lambda code scanning together with Lambda standard scanning (see [Inspector.3] Amazon Inspector Lambda code scanning should be enabled).

Remediation

To enable Amazon Inspector Lambda standard scanning, see Activating scans in the *Amazon Inspector User Guide*.

Security Hub controls for AWS IoT

These AWS Security Hub controls evaluate the AWS IoT service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[IoT.1] AWS IoT Device Defender security profiles should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoT::SecurityProfile`

AWS Config rule: `tagged-iot-securityprofile` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS IoT Device Defender security profile has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the security profile doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the security profile isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS IoT Device Defender security profile, see Tagging your AWS IoT resources in the *AWS IoT Developer Guide*.

[IoT.2] AWS IoT Core mitigation actions should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoT::MitigationAction`

AWS Config rule: `tagged-iot-mitigationaction` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS IoT Core mitigation action has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the mitigation action doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the mitigation action isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS IoT Core mitigation action, see Tagging your AWS IoT resources in the *AWS IoT Developer Guide*.

[IoT.3] AWS IoT Core dimensions should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoT::Dimension`

AWS Config rule: `tagged-iot-dimension` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS IoT Core dimension has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the dimension doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the dimension isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS IoT Core dimension, see Tagging your AWS IoT resources in the *AWS IoT Developer Guide*.

[IoT.4] AWS IoT Core authorizers should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoT::Authorizer`

AWS Config rule: `tagged-iot-authorizer` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS IoT Core authorizer has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the authorizer doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the authorizer isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS IoT Core authorizer, see Tagging your AWS IoT resources in the *AWS IoT Developer Guide*.

[IoT.5] AWS IoT Core role aliases should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoT::RoleAlias`

AWS Config rule: `tagged-iot-rolealias` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS IoT Core role alias has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the role alias doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the role alias isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS IoT Core role alias, see Tagging your AWS IoT resources in the *AWS IoT Developer Guide*.

[IoT.6] AWS IoT Core policies should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoT::Policy`

AWS Config rule: `tagged-iot-policy` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS IoT Core policy has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the policy doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the policy isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an AWS IoT Core policy, see Tagging your AWS IoT resources in the *AWS IoT Developer Guide*.

Security Hub controls for AWS IoT Events

These AWS Security Hub controls evaluate the AWS IoT Events service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[IoTEvents.1] AWS IoT Events inputs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTEvents::Input`

AWS Config rule: `iotevents-input-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT Events input has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the input doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the input isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT Events input, see Tagging your AWS IoT Events resources in the *AWS IoT Events Developer Guide*.

[IoTEvents.2] AWS IoT Events detector models should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTEvents::DetectorModel`

AWS Config rule: `iotevents-detector-model-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT Events detector model has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the detector model doesn't have

any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the detector model isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT Events detector model, see Tagging your AWS IoT Events resources in the *AWS IoT Events Developer Guide*.

[IoTEvents.3] AWS IoT Events alarm models should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTEvents::AlarmModel`

AWS Config rule: `iotevents-alarm-model-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT Events alarm model has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the alarm model doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the alarm model isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT Events alarm model, see Tagging your AWS IoT Events resources in the *AWS IoT Events Developer Guide*.

Security Hub controls for AWS IoT SiteWise

These AWS Security Hub controls evaluate the AWS IoT SiteWise service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTSiteWise::AssetModel`

AWS Config rule: `iotsitewise-asset-model-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT SiteWise asset model has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the asset model doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the

parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the asset model isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT SiteWise asset model, see Tag your AWS IoT SiteWise resources in the *AWS IoT SiteWise User Guide*.

[IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTSiteWise::Dashboard`

AWS Config rule: `iotsitewise-dashboard-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT SiteWise dashboard has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the dashboard doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the dashboard isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT SiteWise dashboard, see Tag your AWS IoT SiteWise resources in the *AWS IoT SiteWise User Guide*.

[IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTSiteWise::Gateway`

AWS Config rule: `iotsitewise-gateway-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT SiteWise gateway has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the gateway doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the gateway isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT SiteWise gateway, see Tag your AWS IoT SiteWise resources in the *AWS IoT SiteWise User Guide*.

[IoTSiteWise.4] AWS IoT SiteWise portals should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTSiteWise::Portal`

AWS Config rule: `iotsitewise-portal-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource	StringList (maximum of 6 items)	1–6 tag keys that meet AWS	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
	must contain. Tag keys are case sensitive.		requirements.	

This control checks whether an AWS IoT SiteWise portal has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the portal doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the portal isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT SiteWise portal, see Tag your AWS IoT SiteWise resources in the *AWS IoT SiteWise User Guide*.

[IoTSiteWise.5] AWS IoT SiteWise projects should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTSiteWise::Project`

AWS Config rule: `iotsitewise-project-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT SiteWise project has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the project doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the project isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM

principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT SiteWise project, see Tag your AWS IoT SiteWise resources in the *AWS IoT SiteWise User Guide*.

Security Hub controls for AWS IoT TwinMaker

These AWS Security Hub controls evaluate the AWS IoT TwinMaker service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTTwinMaker::SyncJob`

AWS Config rule: `iottwinmaker-sync-job-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT TwinMaker sync job has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the sync job doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the sync job isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT TwinMaker sync job, see TagResource in the *AWS IoT TwinMaker User Guide*.

[IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTTwinMaker::Workspace`

AWS Config rule: `iottwinmaker-workspace-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT TwinMaker workspace has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the workspace doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the workspace isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT TwinMaker workspace, see TagResource in the *AWS IoT TwinMaker User Guide*.

[IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTTwinMaker::Scene`

AWS Config rule: `iottwinmaker-scene-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource	StringList (maximum of 6 items)	1–6 tag keys that meet AWS	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
	must contain. Tag keys are case sensitive.		requirements.	

This control checks whether an AWS IoT TwinMaker scene has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the scene doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the scene isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT TwinMaker scene, see TagResource in the *AWS IoT TwinMaker User Guide*.

[IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTTwinMaker::Entity`

AWS Config rule: `iottwinmaker-entity-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT TwinMaker entity has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the entity doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the entity isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM

principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT TwinMaker entity, see TagResource in the *AWS IoT TwinMaker User Guide*.

Security Hub controls for AWS IoT Wireless

These AWS Security Hub controls evaluate the AWS IoT Wireless service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[IoTWireless.1] AWS IoT Wireless multicast groups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTWireless::MulticastGroup`

AWS Config rule: `iotwireless-multicast-group-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT Wireless multicast group has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the multicast group doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the multicast group isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT Wireless multicast group, see Tagging your AWS IoT Wireless resources in the *AWS IoT Wireless Developer Guide*.

[IoTWireless.2] AWS IoT Wireless service profiles should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTWireless::ServiceProfile`

AWS Config rule: `iotwireless-service-profile-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS IoT Wireless service profile has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the service profile doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the service profile isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT Wireless service profile, see Tagging your AWS IoT Wireless resources in the *AWS IoT Wireless Developer Guide*.

[IoTWireless.3] AWS IoT FUOTA tasks should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IoTWireless::FuotaTask`

AWS Config rule: `iotwireless-fuota-task-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource	StringList (maximum of 6 items)	1–6 tag keys that meet AWS	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
	must contain. Tag keys are case sensitive.		requirements.	

This control checks whether an AWS IoT Wireless firmware update over-the-air (FUOTA) task has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the FUOTA task doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the FUOTA task isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS IoT Wireless FUOTA task, see Tagging your AWS IoT Wireless resources in the *AWS IoT Wireless Developer Guide*.

Security Hub controls for Amazon IVS

These AWS Security Hub controls evaluate the Amazon Interactive Video Service (IVS) service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[IVS.1] IVS playback key pairs should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IVS::PlaybackKeyPair`

AWS Config rule: `ivs-playback-key-pair-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon IVS playback key pair has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the playback key pair doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the playback key pair isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an IVS playback key pair, see TagResource in the *Amazon IVS Real-Time Streaming API Reference*.

[IVS.2] IVS recording configurations should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IVS::RecordingConfiguration`

AWS Config rule: `ivs-recording configuration-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon IVS recording configuration has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the recording configuration doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the recording configuration isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an IVS recording configuration, see TagResource in the *Amazon IVS Real-Time Streaming API Reference*.

[IVS.3] IVS channels should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::IVS::Channel`

AWS Config rule: `ivs-channel-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon IVS channel has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the channel doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the channel isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other

criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an IVS channel, see TagResource in the *Amazon IVS Real-Time Streaming API Reference*.

Security Hub controls for Amazon Keyspaces

These AWS Security Hub controls evaluate the Amazon Keyspaces service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Keyspaces.1] Amazon Keyspaces keyspaces should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Cassandra::Keyspace`

AWS Config rule: `cassandra-keyspace-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Keyspaces keyspace has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the keyspace doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the keyspace isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an Amazon Keyspaces keyspace, see Add tags to a keyspace in the *Amazon Keyspaces Developer Guide*.

Security Hub controls for Kinesis

These AWS Security Hub controls evaluate the Amazon Kinesis service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Kinesis.1] Kinesis streams should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::Kinesis::Stream`

AWS Config rule: kinesis-stream-encrypted

Schedule type: Change triggered

Parameters: None

This control checks if Kinesis Data Streams are encrypted at rest with server-side encryption. This control fails if a Kinesis stream is not encrypted at rest with server-side encryption.

Server-side encryption is a feature in Amazon Kinesis Data Streams that automatically encrypts data before it's at rest by using an AWS KMS key. Data is encrypted before it's written to the Kinesis stream storage layer, and decrypted after it's retrieved from storage. As a result, your data is encrypted at rest within the Amazon Kinesis Data Streams service.

Remediation

For information about enabling server-side encryption for Kinesis streams, see How do I get started with server-side encryption? in the *Amazon Kinesis Developer Guide*.

[Kinesis.2] Kinesis streams should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Kinesis::Stream`

AWS Configrule: `tagged-kinesis-stream` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Kinesis data stream has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the data stream doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the data stream isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals.

You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Kinesis data stream, see Tagging your streams in Amazon Kinesis Data Streams in the *Amazon Kinesis Developer Guide*.

[Kinesis.3] Kinesis streams should have an adequate data retention period

Severity: Medium

Resource type: `AWS::Kinesis::Stream`

AWS Configrule: kinesis-stream-backup-retention-check

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`minimumBackupRetentionPeriod`	Minimum number of hours that the data should be retained.	String	24 to 8760	168

This control checks whether an Amazon Kinesis data stream has a data retention period greater than or equal to the specified time frame. The control fails if the data retention period is less than

the specified time frame. Unless you provide a custom parameter value for the data retention period, Security Hub uses a default value of 168 hours.

In Kinesis Data Streams, a data stream is an ordered sequence of data records meant to be written to and read from in real time. Data records are stored in shards in your stream temporarily. The time period from when a record is added to when it is no longer accessible is called the retention period. Kinesis Data Streams almost immediately makes records older than the new retention period inaccessible after decreasing the retention period. For example, changing the retention period from 24 hours to 48 hours means that records added to the stream 23 hours 55 minutes prior are still available after 24 hours.

Remediation

To change the backup retention period for your Kinesis Data Streams, see Change the data retention period in the *Amazon Kinesis Data Streams Developer Guide*.

Security Hub controls for AWS KMS

These AWS Security Hub controls evaluate the AWS Key Management Service (AWS KMS) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys

Related requirements: NIST.800-53.r5 AC-2, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-6(3)

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::IAM::Policy`

AWS Config rule: iam-customer-policy-blocked-kms-actions

Schedule type: Change triggered

Parameters:

- `blockedActionsPatterns: kms:ReEncryptFrom, kms:Decrypt` (not customizable)

- `excludePermissionBoundaryPolicy`: `True` (not customizable)

Checks whether the default version of IAM customer managed policies allow principals to use the AWS KMS decryption actions on all resources. The control fails if the policy is open enough to allow `kms:Decrypt` or `kms:ReEncryptFrom` actions on all KMS keys.

The control only checks KMS keys in the Resource element and doesn't take into account any conditionals in the Condition element of a policy. In addition, the control evaluates both attached and unattached customer managed policies. It doesn't check inline policies or AWS managed policies.

With AWS KMS, you control who can use your KMS keys and gain access to your encrypted data. IAM policies define which actions an identity (user, group, or role) can perform on which resources. Following security best practices, AWS recommends that you allow least privilege. In other words, you should grant to identities only the `kms:Decrypt` or `kms:ReEncryptFrom` permissions and only for the keys that are required to perform a task. Otherwise, the user might use keys that are not appropriate for your data.

Instead of granting permissions for all keys, determine the minimum set of keys that users need to access encrypted data. Then design policies that allow users to use only those keys. For example, do not allow `kms:Decrypt` permission on all KMS keys. Instead, allow `kms:Decrypt` only on keys in a particular Region for your account. By adopting the principle of least privilege, you can reduce the risk of unintended disclosure of your data.

Remediation

To modify an IAM customer managed policy, see Editing customer managed policies in the *IAM User Guide*. When editing your policy, for the `Resource` field, provide the Amazon Resource Name (ARN) of the specific key or keys that you want to allow decryption actions on.

[KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys

Related requirements: NIST.800-53.r5 AC-2, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-6(3)

Category: Protect > Secure access management

Severity: Medium

Resource type:

- `AWS::IAM::Group`
- `AWS::IAM::Role`
- `AWS::IAM::User`

AWS Config rule: iam-inline-policy-blocked-kms-actions

Schedule type: Change triggered

Parameters:

- `blockedActionsPatterns: kms:ReEncryptFrom, kms:Decrypt` (not customizable)

This control checks whether the inline policies that are embedded in your IAM identities (role, user, or group) allow the AWS KMS decryption and re-encryption actions on all KMS keys. The control fails if the policy is open enough to allow `kms:Decrypt` or `kms:ReEncryptFrom` actions on all KMS keys.

The control only checks KMS keys in the Resource element and doesn't take into account any conditionals in the Condition element of a policy.

With AWS KMS, you control who can use your KMS keys and gain access to your encrypted data. IAM policies define which actions an identity (user, group, or role) can perform on which resources. Following security best practices, AWS recommends that you allow least privilege. In other words, you should grant to identities only the permissions they need and only for keys that are required to perform a task. Otherwise, the user might use keys that are not appropriate for your data.

Instead of granting permission for all keys, determine the minimum set of keys that users need to access encrypted data. Then design policies that allow the users to use only those keys. For example, do not allow `kms:Decrypt` permission on all KMS keys. Instead, allow the permission only on specific keys in a specific Region for your account. By adopting the principle of least privilege, you can reduce the risk of unintended disclosure of your data.

Remediation

To modify an IAM inline policy, see Editing inline policies in the *IAM User Guide*. When editing your policy, for the `Resource` field, provide the Amazon Resource Name (ARN) of the specific key or keys that you want to allow decryption actions on.

[KMS.3] AWS KMS keys should not be deleted unintentionally

Related requirements: NIST.800-53.r5 SC-12, NIST.800-53.r5 SC-12(2)

Category: Protect > Data protection > Data deletion protection

Severity: Critical

Resource type: `AWS::KMS::Key`

AWS Config rule: `kms-cmk-not-scheduled-for-deletion-2` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether KMS keys are scheduled for deletion. The control fails if a KMS key is scheduled for deletion.

KMS keys cannot be recovered once deleted. Data encrypted under a KMS key is also permanently unrecoverable if the KMS key is deleted. If meaningful data has been encrypted under a KMS key scheduled for deletion, consider decrypting the data or re-encrypting the data under a new KMS key unless you are intentionally performing a *cryptographic erasure*.

When a KMS key is scheduled for deletion, a mandatory waiting period is enforced to allow time to reverse the deletion, if it was scheduled in error. The default waiting period is 30 days, but it can be reduced to as short as 7 days when the KMS key is scheduled for deletion. During the waiting period, the scheduled deletion can be canceled and the KMS key will not be deleted.

For additional information regarding deleting KMS keys, see Deleting KMS keys in the *AWS Key Management Service Developer Guide*.

Remediation

To cancel a scheduled KMS key deletion, see **To cancel key deletion** under Scheduling and canceling key deletion (console) in the *AWS Key Management Service Developer Guide*.

[KMS.4] AWS KMS key rotation should be enabled

Related requirements: CIS AWS Foundations Benchmark v3.0.0/3.6, CIS AWS Foundations Benchmark v1.4.0/3.8, CIS AWS Foundations Benchmark v1.2.0/2.8, NIST.800-53.r5 SC-12, NIST.800-53.r5 SC-12(2), NIST.800-53.r5 SC-28(3), PCI DSS v3.2.1/3.6.4, PCI DSS v4.0.1/3.7.4

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::KMS::Key`

AWS Config rule: cmk-backing-key-rotation-enabled

Schedule type: Periodic

Parameters: None

AWS KMS enables customers to rotate the backing key, which is key material stored in AWS KMS and is tied to the key ID of the KMS key. It's the backing key that is used to perform cryptographic operations such as encryption and decryption. Automated key rotation currently retains all previous backing keys so that decryption of encrypted data can take place transparently.

CIS recommends that you enable KMS key rotation. Rotating encryption keys helps reduce the potential impact of a compromised key because data encrypted with a new key can't be accessed with a previous key that might have been exposed.

Remediation

To enable KMS key rotation, see How to enable and disable automatic key rotation in the *AWS Key Management Service Developer Guide*.

[KMS.5] KMS keys should not be publicly accessible

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: Critical

Resource type: `AWS::KMS::Key`

AWS Config rule: kms-key-policy-no-public-access

Schedule type: Change triggered

Parameters: None

This controls checks whether an AWS KMS key is publicly accessible. The control fails if the KMS key is publicly accessible.

Implementing least privilege access is fundamental to reducing security risk and the impact of errors or malicious intent. If the key policy for an AWS KMS key allows access from external accounts, third parties might be able to encrypt and decrypt data by using the key. This could result in an internal or external threat exfiltrating data from AWS services that use the key.

Note

This control also returns a `FAILED` finding for an AWS KMS key if your configurations prevent AWS Config from recording the key policy in the Configuration Item (CI) for the KMS key. For AWS Config to populate the key policy in the CI for the KMS key, the AWS Config role must have access to read the key policy by using the GetKeyPolicy API call. To resolve this type of `FAILED` finding, check policies that can prevent the AWS Config role from having read access to the key policy for the KMS key. For example, check the following:

- The key policy for the KMS key.
- Service control policies (SCPs) and resource control policies (RCPs) in AWS Organizations that apply to your account.
- Permissions for the AWS Config role, if you are not using the AWS Config service-linked role.

In addition, this control doesn't evaluate policy conditions that use wildcard characters or variables. To produce a `PASSED` finding, conditions in the key policy must only use fixed values, which are values that don't contain wildcard characters or policy variables. For information about policy variables, see Variables and tags in the *AWS Identity and Access Management User Guide*.

Remediation

For information about updating the key policy for an AWS KMS key, see Key policies in AWS KMS in the *AWS Key Management Service Developer Guide*.

Security Hub controls for AWS Lambda

These AWS Security Hub controls evaluate the AWS Lambda service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Lambda.1] Lambda function policies should prohibit public access

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.2, PCI DSS v3.2.1/1.3.4, PCI DSS v3.2.1/7.2.1, PCI DSS v4.0.1/7.2.1

Category: Protect > Secure network configuration

Severity: Critical

Resource type: `AWS::Lambda::Function`

AWS Config rule: lambda-function-public-access-prohibited

Schedule type: Change triggered

Parameters: None

This control checks whether the Lambda function resource-based policy prohibits public access outside of your account. The control fails if public access is permitted. The control also fails if a Lambda function is invoked from Amazon S3, and the policy doesn't include a condition to limit public access, such as `AWS:SourceAccount`. We recommend using other S3 conditions along with `AWS:SourceAccount` in your bucket policy for more refined access.

> **Note**
>
> This control doesn't evaluate policy conditions that use wildcard characters or variables. To produce a `PASSED` finding, conditions in the policy for the Lambda function must only use fixed values, which are values that don't contain wildcard characters or policy variables. For information about policy variables, see Variables and tags in the *AWS Identity and Access Management User Guide*.

The Lambda function should not be publicly accessible, as this may allow unintended access to your function code.

Remediation

To remediate this issue, you must update your function's resource-based policy to remove permissions or to add the `AWS:SourceAccount` condition. You can only update the resource-based policy from the Lambda API or AWS CLI.

To start, review the resource-based policy on the Lambda console. Identify the policy statement that has `Principal` field values that make the policy public, such as `"*"` or `{ "AWS": "*" }`.

You cannot edit the policy from the console. To remove permissions from the function, run the remove-permission command from the AWS CLI.

```
$ aws lambda remove-permission --function-name <function-name> --statement-
id <statement-id>
```

Replace *<function-name>* with the name of the Lambda function, and *<statement-id>* with the statement ID (`Sid`) of the statement that you want to remove.

[Lambda.2] Lambda functions should use supported runtimes

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5), PCI DSS v4.0.1/12.3.4

Category: Protect > Secure development

Severity: Medium

Resource type: `AWS::Lambda::Function`

AWS Config rule: lambda-function-settings-check

Schedule type: Change triggered

Parameters:

- `runtime`: `dotnet8, java21, java17, java11, java8.al2, nodejs22.x, nodejs20.x, nodejs18.x, python3.13, python3.12, python3.11, python3.10, python3.9, ruby3.4, ruby3.3, ruby3.2` (not customizable)

This control checks whether AWS Lambda function runtime settings match the expected values set for the supported runtimes in each language. The control fails if the Lambda function doesn't use a

supported runtime, as noted in the Parameters section. Security Hub ignores functions that have a package type of `Image`.

Lambda runtimes are built around a combination of operating system, programming language, and software libraries that are subject to maintenance and security updates. When a runtime component is no longer supported for security updates, Lambda deprecates the runtime. Even though you can't create functions that use the deprecated runtime, the function is still available to process invocation events. We recommend ensuring that your Lambda functions are current and don't use deprecated runtime environments. For a list of supported runtimes, see Lambda runtimes in the *AWS Lambda Developer Guide*.

Remediation

For more information about supported runtimes and deprecation schedules, see Runtime deprecation policy in the *AWS Lambda Developer Guide*. When you migrate your runtimes to the latest version, follow the syntax and guidance from the publishers of the language. We also recommend applying runtime updates to help reduce the risk of impact to your workloads in the rare event of a runtime version incompatibility.

[Lambda.3] Lambda functions should be in a VPC

Related requirements: PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.2, PCI DSS v3.2.1/1.3.4, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration

Severity: Low

Resource type: `AWS::Lambda::Function`

AWS Config rule: lambda-inside-vpc

Schedule type: Change triggered

Parameters: None

This control checks whether a Lambda function is deployed in a virtual private cloud (VPC). The control fails if the Lambda function isn't deployed in a VPC. Security Hub doesn't evaluate the VPC

subnet routing configuration to determine public reachability. You might see failed findings for Lambda@Edge resources.

Deploying resources in a VPC strengthens security and control over network configurations. Such deployments also offer scalability and high fault tolerance across multiple Availability Zones. You can customize VPC deployments to meet diverse application requirements.

Remediation

To configure an existing function to connect to private subnets in your VPC, see Configuring VPC access in the *AWS Lambda Developer Guide*. We recommend choosing at least two private subnets for high availability and at least one security group that meets the connectivity requirements of the function.

[Lambda.5] VPC Lambda functions should operate in multiple Availability Zones

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::Lambda::Function`

AWS Config rule: lambda-vpc-multi-az-check

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`availabilityZones`	Minimum number of Availability Zones	Enum	`2, 3, 4, 5, 6`	`2`

This control checks if an AWS Lambda function that connects to a virtual private cloud (VPC) operates in at least the specified number of Availability Zone (AZs). The control fails if the function

doesn't operate in at least the specified number of AZs. Unless you provide a custom parameter value for the minimum number of AZs, Security Hub uses a default value of two AZs.

Deploying resources across multiple AZs is an AWS best practice to ensure high availability within your architecture. Availability is a core pillar in the confidentiality, integrity, and availability triad security model. All Lambda functions that connect to a VPC should have a multi-AZ deployment to ensure that a single zone of failure doesn't cause a total disruption of operations.

Remediation

If you configure your function to connect to a VPC in your account, specify subnets in multiple AZs to ensure high availability. For instructions, see Configuring VPC access in the *AWS Lambda Developer Guide*.

Lambda automatically runs other functions in multiple AZs to ensure that it is available to process events in case of a service interruption in a single zone.

[Lambda.6] Lambda functions should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Lambda::Function`

AWS Config rule: `tagged-lambda-function` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS Lambda function has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the function doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the function isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Lambda function, see Using tags on Lambda functions in the *AWS Lambda Developer Guide*.

[Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled

Related requirements: NIST.800-53.r5 CA-7

Category: Identify > Logging

Severity: Low

Resource type: `AWS::Lambda::Function`

AWS Config rule: lambda-function-xray-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether active tracing with AWS X-Ray is enabled for an AWS Lambda function. The control fails if active tracing with X-Ray is disabled for the Lambda function.

AWS X-Ray can provide tracing and monitoring capabilities for AWS Lambda functions, which can save time and effort debugging and operating Lambda functions. It can help you diagnose errors and identify performance bottlenecks, slowdowns, and timeouts by breaking down latency for Lambda functions. It can also help with data privacy and compliance requirements. If you enable active tracing for a Lambda function, X-Ray provides a holistic view of data flow and processing within the Lambda function, which can help you identify potential security vulnerabilities or non-compliant data handling practices. This visibility can help you maintain data integrity, confidentiality, and compliance with relevant regulations.

> **Note**
>
> AWS X-Ray tracing is currently not supported for Lambda functions with Amazon Managed Streaming for Apache Kafka (Amazon MSK), self-managed Apache Kafka, Amazon MQ with ActiveMQ and RabbitMQ, or Amazon DocumentDB event source mappings.

Remediation

For information about enabling active tracing for an AWS Lambda function, see Visualize Lambda function invocations using AWS X-Ray in the *AWS Lambda Developer Guide*.

Security Hub controls for Macie

These AWS Security Hub controls evaluate the Amazon Macie service.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Macie.1] Amazon Macie should be enabled

Related requirements: NIST.800-53.r5 CA-7, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 RA-5, NIST.800-53.r5 SA-8(19), NIST.800-53.r5 SI-4

Category: Detect > Detection services

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: `macie-status-check`

Schedule type: Periodic

This control checks whether Amazon Macie is enabled for an account. The control fails if Macie isn't enabled for the account.

Amazon Macie discovers sensitive data using machine learning and pattern matching, provides visibility into data security risks, and enables automated protection against those risks. Macie automatically and continually evaluates your Amazon Simple Storage Service (Amazon S3) buckets for security and access control, and generates findings to notify you of potential issues with the security or privacy of your Amazon S3 data. Macie also automates discovery and reporting of sensitive data, such as personally identifiable information (PII), to provide you with a better understanding of the data that you store in Amazon S3. To learn more, see the *Amazon Macie User Guide*.

Remediation

To enable Macie, see Enable Macie in the *Amazon Macie User Guide*.

[Macie.2] Macie automated sensitive data discovery should be enabled

Related requirements: NIST.800-53.r5 CA-7, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 RA-5, NIST.800-53.r5 SA-8(19), NIST.800-53.r5 SI-4

Category: Detect > Detection services

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: `macie-auto-sensitive-data-discovery-check`

Schedule type: Periodic

This control checks whether automated sensitive data discovery is enabled for an Amazon Macie administrator account. The control fails if automated sensitive data discovery isn't enabled for a Macie administrator account. This control applies only to administrator accounts.

Macie automates discovery and reporting of sensitive data, such as personally identifiable information (PII), in Amazon Simple Storage Service (Amazon S3) buckets. With automated sensitive data discovery, Macie continually evaluates your bucket inventory and uses sampling techniques to identify and select representative S3 objects from your buckets. Macie then analyzes the selected objects, inspecting them for sensitive data. As the analyses progress, Macie updates statistics, inventory data, and other information that it provides about your S3 data. Macie also generates findings to report sensitive data that it finds.

Remediation

To create and configure automated sensitive data discovery jobs to analyze objects in S3 buckets, see Configuring automated sensitive data discovery for your account in the *Amazon Macie User Guide*.

Security Hub controls for Amazon MSK

These AWS Security Hub controls evaluate the Amazon Managed Streaming for Apache Kafka (Amazon MSK) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[MSK.1] MSK clusters should be encrypted in transit among broker nodes

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::MSK::Cluster`

AWS Config rule: msk-in-cluster-node-require-tls

Schedule type: Change triggered

Parameters: None

This controls checks whether an Amazon MSK cluster is encrypted in transit with HTTPS (TLS) among the broker nodes of the cluster. The control fails if plain text communication is enabled for a cluster broker node connection.

HTTPS offers an extra layer of security as it uses TLS to move data and can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. By default, Amazon MSK encrypts data in transit with TLS. However, you can override this default at the time that you create the cluster. We recommend using encrypted connections over HTTPS (TLS) for-broker node connections.

Remediation

For information about updating the encryption settings for an Amazon MSK cluster, see Updating security settings of a cluster in the *Amazon Managed Streaming for Apache Kafka Developer Guide*.

[MSK.2] MSK clusters should have enhanced monitoring configured

Related requirements: NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-2

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::MSK::Cluster`

AWS Config rule: msk-enhanced-monitoring-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon MSK cluster has enhanced monitoring configured, specified by a monitoring level of at least `PER_TOPIC_PER_BROKER`. The control fails if the monitoring level for the cluster is set to `DEFAULT` or `PER_BROKER`.

The `PER_TOPIC_PER_BROKER` monitoring level provides more granular insights into the performance of your MSK cluster, and also provides metrics related to resource utilization, such as CPU and memory usage. This helps you identify performance bottlenecks and resource utilization patterns for individual topics and brokers. This visibility, in turn, can optimize the performance of your Kafka brokers.

Remediation

To configure enhanced monitoring for an MSK cluster, complete the following steps:

1. Open the Amazon MSK console at https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.

2. In the navigation pane, choose **Clusters**. Then, choose a cluster.
3. For **Action**, select **Edit monitoring**.
4. Select the option for **Enhanced topic-level monitoring**.
5. Choose **Save changes**.

For more information about monitoring levels, see Amazon MSK metrics for monitoring Standard brokers with CloudWatch in the *Amazon Managed Streaming for Apache Kafka Developer Guide*.

[MSK.3] MSK Connect connectors should be encrypted in transit

Related requirements: PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::KafkaConnect::Connector`

AWS Config rule: `msk-connect-connector-encrypted` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon MSK Connect connector is encrypted in transit. This control fails if the connector isn't encrypted in transit.

Data in transit refers to data that moves from one location to another, such as between nodes in your cluster or between your cluster and your application. Data may move across the internet or within a private network. Encrypting data in transit reduces the risk that an unauthorized user can eavesdrop on network traffic.

Remediation

You can enable encryption in transit when you create an MSK Connect connector. You can't change encryption settings after creating a connector. For more information, see Create a connector in the *Amazon Managed Streaming for Apache Kafka Developer Guide*.

[MSK.4] MSK clusters should have public access disabled

Category: Protect > Secure access management > Resource not publicly accessible

Severity: Critical

Resource type: `AWS::MSK::Cluster`

AWS Config rule: msk-cluster-public-access-disabled

Schedule type: Change triggered

Parameters: None

This control checks whether public access is disabled for an Amazon MSK cluster. The control fails if public access is enabled for the MSK cluster.

By default, clients can access an Amazon MSK cluster only if they're in the same VPC as the cluster. All communication between Kafka clients and an MSK cluster are private by default and streaming data doesn't traverse the internet. However, if an MSK cluster is configured to allow public access, anyone on the internet can establish a connection to Apache Kafka brokers that are running within the cluster. This can lead to issues such as unauthorized access, data breaches, or exploitation of vulnerabilities. If you restrict access to a cluster by requiring authentication and authorization measures, you can help protect sensitive information and maintain the integrity of your resources.

Remediation

For information about managing public access to an Amazon MSK cluster, see Turn on public access to an MSK Provisioned cluster in the *Amazon Managed Streaming for Apache Kafka Developer Guide*.

[MSK.5] MSK connectors should have logging enabled

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::KafkaConnect::Connector`

AWS Config rule: msk-connect-connector-logging-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether logging is enabled for an Amazon MSK connector. The control fails if logging is disabled for the MSK connector.

Amazon MSK connectors integrate external systems and Amazon services with Apache Kafka by continuously copying streaming data from a data source into an Apache Kafka cluster, or

continuously copying data from a cluster into a data sink. MSK Connect can write log events that can help debug a connector. When you create a connector, you can specify zero or more of the following log destinations: Amazon CloudWatch Logs, Amazon S3, and Amazon Data Firehose.

> **Note**
>
> Sensitive configuration values can appear in connector logs if a plugin does not define those values as secret. Kafka Connect treats undefined configuration values the same as any other plaintext value.

Remediation

To enable logging for an existing Amazon MSK connector, you have to re-create the connector with the appropriate logging configuration. For information about configuration options, see Logging for MSK Connect in the *Amazon Managed Streaming for Apache Kafka Developer Guide*.

[MSK.6] MSK clusters should disable unauthenticated access

Category: Protect > Secure access management > Passwordless authentication

Severity: Medium

Resource type: `AWS::MSK::Cluster`

AWS Config rule: msk-unrestricted-access-check

Schedule type: Change triggered

Parameters: None

This control checks whether unauthenticated access is enabled for an Amazon MSK cluster. The control fails if unauthenticated access is enabled for the MSK cluster.

Amazon MSK supports client authentication and authorization mechanisms to control access to a cluster. These mechanisms verify the identity of clients connecting to the cluster and determine which actions clients can perform. An MSK cluster can be configured to allow unauthenticated access, which allows any client with network connectivity to publish and subscribe to Kafka topics without providing credentials. Running an MSK cluster without requiring authentication violates the principle of least privilege and can expose the cluster to unauthorized access. It can allow any client to access, modify, or delete data in Kafka topics, potentially resulting in data breaches, unauthorized data modifications, or service disruptions. We recommend enabling authentication

mechanisms such as IAM authentication, SASL/SCRAM, or mutual TLS to ensure proper access control and maintain security compliance.

Remediation

For information about changing the authentication settings for an Amazon MSK cluster, see the following sections of the *Amazon Managed Streaming for Apache Kafka Developer Guide*: Update security settings of an Amazon MSK cluster and Authentication and authorization for Apache Kafka APIs.

Security Hub controls for Amazon MQ

These AWS Security Hub controls evaluate the Amazon MQ service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch

Related requirements: NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-12, NIST.800-53.r5 SI-4, PCI DSS v4.0.1/10.3.3

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::AmazonMQ::Broker`

AWS Config rule: `mq-cloudwatch-audit-log-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon MQ ActiveMQ broker streams audit logs to Amazon CloudWatch Logs. The control fails if the broker doesn't stream audit logs to CloudWatch Logs.

By publishing ActiveMQ broker logs to CloudWatch Logs, you can create CloudWatch alarms and metrics that increase the visibility of security-related information.

Remediation

To stream ActiveMQ broker logs to CloudWatch Logs, see Configuring Amazon MQ for ActiveMQ logs in the *Amazon MQ Developer Guide*.

[MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled

Related requirements: NIST.800-53.r5 CM-3, NIST.800-53.r5 SI-2, PCI DSS v4.0.1/6.3.3

Category: Identify > Vulnerability, patch, and version management

Severity: Low

Resource type: `AWS::AmazonMQ::Broker`

AWS Config rule: `mq-auto-minor-version-upgrade-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon MQ broker has automatic minor version upgrade enabled. The control fails if the broker doesn't have automatic minor version upgrade enabled.

As Amazon MQ releases and supports new broker engine versions, the changes are backward-compatible with an existing application and don't deprecate existing functionality. Automatic broker engine version updates protect you against security risks, help fix bugs, and improve functionality.

> **Note**
>
> When the broker associated with automatic minor version upgrade is on its latest patch and becomes unsupported, you must take manual action to upgrade.

Remediation

To enable automatic minor version upgrade for an MQ broker, see Automatically upgrading the minor engine version in the *Amazon MQ Developer Guide*.

[MQ.4] Amazon MQ brokers should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::AmazonMQ::Broker`

AWS Config rule: `tagged-amazonmq-broker` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon MQ broker has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the broker doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the broker isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Amazon MQ broker, see Tagging resources in the *Amazon MQ Developer Guide*.

[MQ.5] ActiveMQ brokers should use active/standby deployment mode

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Low

Resource type: `AWS::AmazonMQ::Broker`

AWS Config rule: `mq-active-deployment-mode`

Schedule type: Change triggered

Parameters: None

This control checks whether the deployment mode for an Amazon MQ ActiveMQ broker is set to active/standby. The control fails if a single-instance broker (enabled by default) is set as the deployment mode.

Active/standby deployment provides high availability for your Amazon MQ ActiveMQ brokers in an AWS Region. The active/standby deployment mode includes two broker instances in two different Availability Zones, configured in a redundant pair. These brokers communicate synchronously with your application, which can reduce downtime and loss of data in the event of a failure.

Remediation

To create a new ActiveMQ broker with active/standby deployment mode, see Creating and configuring an ActiveMQ broker in the *Amazon MQ Developer Guide*. For **Deployment mode**, choose **Active/standby broker**. You can't change the deployment mode for an existing broker. Instead, you must create a new broker and copy the settings over from the old broker.

[MQ.6] RabbitMQ brokers should use cluster deployment mode

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5

Category: Recover > Resilience > High availability

Severity: Low

Resource type: `AWS::AmazonMQ::Broker`

AWS Config rule: `mq-rabbit-deployment-mode`

Schedule type: Change triggered

Parameters: None

This control checks whether the deployment mode for an Amazon MQ RabbitMQ broker is set to cluster deployment. The control fails if a single-instance broker (enabled by default) is set as the deployment mode.

Cluster deployment provides high availability for your Amazon MQ RabbitMQ brokers in an AWS Region. The cluster deployment is a logical grouping of three RabbitMQ broker nodes, each with its own Amazon Elastic Block Store (Amazon EBS) volume and a shared state. The cluster deployment ensures that data is replicated to all nodes in the cluster, which can reduce downtime and loss of data in the event of a failure.

Remediation

To create a new RabbitMQ broker with cluster deployment mode, see Creating and connecting to a RabbitMQ broker in the *Amazon MQ Developer Guide*. For **Deployment mode**, choose **Cluster deployment**. You can't change the deployment mode for an existing broker. Instead, you must create a new broker and copy the settings over from the old broker.

Security Hub controls for Neptune

These AWS Security Hub controls evaluate the Amazon Neptune service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Neptune.1] Neptune DB clusters should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `neptune-cluster-encrypted`

Schedule type: Change triggered

Parameters: None

This control checks whether a Neptune DB cluster is encrypted at rest. The control fails if a Neptune DB cluster isn't encrypted at rest.

Data at rest refers to any data that's stored in persistent, non-volatile storage for any duration. Encryption helps you protect the confidentiality of such data, reducing the risk that an unauthorized user can access it. Encrypting your Neptune DB clusters protects your data and metadata against unauthorized access. It also fulfills compliance requirements for data-at-rest encryption of production file systems.

Remediation

You can enable encryption at rest when you create a Neptune DB cluster. You can't change encryption settings after creating a cluster. For more information, see Encrypting Neptune resources at rest in the *Neptune User Guide*.

[Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(1), NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-6(5), NIST.800-53.r5 AU-7(1), NIST.800-53.r5 AU-9(7), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-20, NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-4(5), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.3.3

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `neptune-cluster-cloudwatch-log-export-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether a Neptune DB cluster publishes audit logs to Amazon CloudWatch Logs. The control fails if a Neptune DB cluster doesn't publish audit logs to CloudWatch Logs. `EnableCloudWatchLogsExport` should be set to `Audit`.

Amazon Neptune and Amazon CloudWatch are integrated so that you can gather and analyze performance metrics. Neptune automatically sends metrics to CloudWatch and also supports CloudWatch Alarms. Audit logs are highly customizable. When you audit a database, each operation on the data can be monitored and logged to an audit trail, including information about which database cluster is accessed and how. We recommend sending these logs to CloudWatch to help you monitor your Neptune DB clusters.

Remediation

To publish Neptune audit logs to CloudWatch Logs, see Publishing Neptune logs to Amazon CloudWatch Logs in the *Neptune User Guide*. In the **Log exports** section, choose **Audit**.

[Neptune.3] Neptune DB cluster snapshots should not be public

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: Critical

Resource type: `AWS::RDS::DBClusterSnapshot`

AWS Config rule: `neptune-cluster-snapshot-public-prohibited`

Schedule type: Change triggered

Parameters: None

This control checks whether a Neptune manual DB cluster snapshot is public. The control fails if a Neptune manual DB cluster snapshot is public.

A Neptune DB cluster manual snapshot should not be public unless intended. If you share an unencrypted manual snapshot as public, the snapshot is available to all AWS accounts. Public snapshots may result in unintended data exposure.

Remediation

To remove public access for Neptune manual DB cluster snapshots, see Sharing a DB cluster snapshot in the *Neptune User Guide*.

[Neptune.4] Neptune DB clusters should have deletion protection enabled

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-3, NIST.800-53.r5 SC-5(2)

Category: Protect > Data protection > Data deletion protection

Severity: Low

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `neptune-cluster-deletion-protection-enabled`

Schedule type: Change triggered

Parameters: None

This control checks if a Neptune DB cluster has deletion protection enabled. The control fails if a Neptune DB cluster doesn't have deletion protection enabled.

Enabling cluster deletion protection offers an additional layer of protection against accidental database deletion or deletion by an unauthorized user. A Neptune DB cluster can't be deleted while deletion protection is enabled. You must first disable deletion protection before a delete request can succeed.

Remediation

To enable deletion protection for an existing Neptune DB cluster, see Modifying the DB cluster by using the console, CLI, and API in the *Amazon Aurora User Guide*.

[Neptune.5] Neptune DB clusters should have automated backups enabled

Related requirements: NIST.800-53.r5 SI-12

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: neptune-cluster-backup-retention-check

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value	
minimumBackupRetentionPeriod	Minimum backup retention period in days	Integer	7 to 35	7	

This control checks whether a Neptune DB cluster has automated backups enabled, and a backup retention period greater than or equal to the specified time frame. The control fails if backups aren't enabled for the Neptune DB cluster, or if the retention period is less than the specified time frame. Unless you provide a custom parameter value for the backup retention period, Security Hub uses a default value of 7 days.

Backups help you recover more quickly from a security incident and strengthen the resilience of your systems. By automating backups for your Neptune DB clusters, you'll be able to restore your systems to a point in time and minimize downtime and data loss.

Remediation

To enable automated backups and set a backup retention period for your Neptune DB clusters, see Enabling automated backups in the *Amazon RDS User Guide*. For **Backup retention period**, choose a value greater than or equal to 7.

[Neptune.6] Neptune DB cluster snapshots should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SC-7(18)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::RDS::DBClusterSnapshot`

AWS Config rule: `neptune-cluster-snapshot-encrypted`

Schedule type: Change triggered

Parameters: None

This control checks whether a Neptune DB cluster snapshot is encrypted at rest. The control fails if a Neptune DB cluster isn't encrypted at rest.

Data at rest refers to any data that's stored in persistent, non-volatile storage for any duration. Encryption helps you protect the confidentiality of such data, reducing the risk that an unauthorized user gets access to it. Data in Neptune DB clusters snapshots should be encrypted at rest for an added layer of security.

Remediation

You can't encrypt an existing Neptune DB cluster snapshot. Instead, you must restore the snapshot to a new DB cluster and enable encryption on the cluster. You can create an encrypted snapshot from the encrypted cluster. For instructions, see Restoring from a DB cluster snapshot and Creating a DB cluster snapshot in Neptune in the *Neptune User Guide*.

[Neptune.7] Neptune DB clusters should have IAM database authentication enabled

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6

Category: Protect > Secure access management > Passwordless authentication

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `neptune-cluster-iam-database-authentication`

Schedule type: Change triggered

Parameters: None

This control checks if a Neptune DB cluster has IAM database authentication enabled. The control fails if IAM database authentication isn't enabled for a Neptune DB cluster.

IAM database authentication for Amazon Neptune database clusters removes the need to store user credentials within the database configuration because authentication is managed externally using IAM. When IAM database authentication is enabled, each request needs to be signed using AWS Signature Version 4.

Remediation

By default, IAM database authentication is disabled when you create a Neptune DB cluster. To enable it, see Enabling IAM database authentication in Neptune in the *Neptune User Guide*.

[Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `neptune-cluster-copy-tags-to-snapshot-enabled`

Schedule type: Change triggered

Parameters: None

This control checks if a Neptune DB cluster is configured to copy all tags to snapshots when the snapshots are created. The control fails if a Neptune DB cluster isn't configured to copy tags to snapshots.

Identification and inventory of your IT assets is a crucial aspect of governance and security. You should tag snapshots in the same way as their parent Amazon RDS database clusters. Copying tags ensures that the metadata for the DB snapshots matches that of the parent database clusters, and that access policies for the DB snapshot also match those of the parent DB instance.

Remediation

To copy tags to snapshots for Neptune DB clusters, see Copying tags in Neptune in the *Neptune User Guide*.

[Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `neptune-cluster-multi-az-enabled`

Schedule type: Change triggered

Parameters: None

This control checks if an Amazon Neptune DB cluster has read-replica instances in multiple Availability Zones (AZs). The control fails if the cluster is deployed in only one AZ.

If an AZ is unavailable and during regular maintenance events, read-replicas serve as failover targets for the primary instance. That is, if the primary instance fails, Neptune promotes a read-replica instance to become the primary instance. By contrast, if your DB cluster doesn't include any read-replica instances, your DB cluster remains unavailable when the primary instance fails until it has been re-created. Re-creating the primary instance takes considerably longer than promoting a read-replica. To ensure high availability, we recommend that you create one or more read-replica instances that have the same DB instance class as the primary instance and are located in different AZs than the primary instance.

Remediation

To deploy a Neptune DB cluster in multiple AZs,, see Read-replica DB instances in a Neptune DB cluster in the *Neptune User Guide*.

Security Hub controls for AWS Network Firewall

These AWS Security Hub controls evaluate the AWS Network Firewall service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::NetworkFirewall::Firewall`

AWS Config rule: netfw-multi-az-enabled

Schedule type: Change triggered

Parameters: None

This control evaluates whether a firewall managed through AWS Network Firewall is deployed across multiple Availability Zones (AZs). The control fails if a firewall is deployed in only one AZ.

AWS global infrastructure includes multiple AWS Regions. AZs are physically separated, isolated locations within each Region that are connected by low-latency, high-throughput, and highly redundant networking. By deploying a Network Firewall firewall across multiple AZs, you can balance and shift traffic among AZs, which helps you design highly available solutions.

Remediation

Deploying a Network Firewall firewall across multiple AZs

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, under **Network Firewall**, choose **Firewalls**.
3. On the **Firewalls** page, select the firewall that you want to edit.
4. On the firewall details page, choose the **Firewall details** tab.
5. In the **Associated policy and VPC** section, choose **Edit**
6. To add a new AZ, choose **Add New Subnet**. Select the AZ and subnet that you would like to use. Ensure that you select at least two AZs.
7. Choose **Save**.

[NetworkFirewall.2] Network Firewall logging should be enabled

Related requirements: NIST.800-53.r5 AC-2(12), NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-9(7),

NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4, NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), NIST.800-171.r2 3.1.20, NIST.800-171.r2 3.13.1

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::NetworkFirewall::LoggingConfiguration`

AWS Config rule: netfw-logging-enabled

Schedule type: Periodic

Parameters: None

This control checks whether logging is enabled for an AWS Network Firewall firewall. The control fails if logging isn't enabled for at least one log type or if the logging destination doesn't exist.

Logging helps you maintain the reliability, availability, and performance of your firewalls. In Network Firewall, logging gives you detailed information about network traffic, including the time that the stateful engine received a packet flow, detailed information about the packet flow, and any stateful rule action taken against the packet flow.

Remediation

To enable logging for a firewall, see Updating a firewall's logging configuration in the *AWS Network Firewall Developer Guide*.

[NetworkFirewall.3] Network Firewall policies should have at least one rule group associated

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-171.r2 3.1.3, NIST.800-171.r2 3.13.1

Category: Protect > Secure Network Configuration

Severity: Medium

Resource type: `AWS::NetworkFirewall::FirewallPolicy`

AWS Config rule: netfw-policy-rule-group-associated

Schedule type: Change triggered

Parameters: None

This control checks whether a Network Firewall policy has any stateful or stateless rule groups associated. The control fails if stateless or stateful rule groups are not assigned.

A firewall policy defines how your firewall monitors and handles traffic in Amazon Virtual Private Cloud (Amazon VPC). Configuration of stateless and stateful rule groups helps to filter packets and traffic flows, and defines default traffic handling.

Remediation

To add a rule group to a Network Firewall policy, see Updating a firewall policy in the *AWS Network Firewall Developer Guide*. For information about creating and managing rule groups, see Rule groups in AWS Network Firewall.

[NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Protect > Secure Network Configuration

Severity: Medium

Resource type: `AWS::NetworkFirewall::FirewallPolicy`

AWS Config rule: netfw-policy-default-action-full-packets

Schedule type: Change triggered

Parameters:

- `statelessDefaultActions: aws:drop,aws:forward_to_sfe` (not customizable)

This control checks whether the default stateless action for full packets for a Network Firewall policy is drop or forward. The control passes if `Drop` or `Forward` is selected, and fails if `Pass` is selected.

A firewall policy defines how your firewall monitors and handles traffic in Amazon VPC. You configure stateless and stateful rule groups to filter packets and traffic flows. Defaulting to `Pass` can allow unintended traffic.

Remediation

To change your firewall policy, see Updating a firewall policy in the *AWS Network Firewall Developer Guide*. For **Stateless default actions**, choose **Edit**. Then, choose **Drop** or **Forward to stateful rule groups** as the **Action**.

[NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-171.r2 3.1.3, NIST.800-171.r2 3.1.14, NIST.800-171.r2 3.13.1, NIST.800-171.r2 3.13.6

Category: Protect > Secure Network Configuration

Severity: Medium

Resource type: `AWS::NetworkFirewall::FirewallPolicy`

AWS Config rule: netfw-policy-default-action-fragment-packets

Schedule type: Change triggered

Parameters:

- `statelessFragDefaultActions (Required) : aws:drop, aws:forward_to_sfe` (not customizable)

This control checks whether the default stateless action for fragmented packets for a Network Firewall policy is drop or forward. The control passes if `Drop` or `Forward` is selected, and fails if `Pass` is selected.

A firewall policy defines how your firewall monitors and handles traffic in Amazon VPC. You configure stateless and stateful rule groups to filter packets and traffic flows. Defaulting to `Pass` can allow unintended traffic.

Remediation

To change your firewall policy, see Updating a firewall policy in the *AWS Network Firewall Developer Guide*. For **Stateless default actions**, choose **Edit**. Then, choose **Drop** or **Forward to stateful rule groups** as the **Action**.

[NetworkFirewall.6] Stateless Network Firewall rule group should not be empty

Related requirements: NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(5), NIST.800-171.r2 3.1.3, NIST.800-171.r2 3.1.14, NIST.800-171.r2 3.13.1, NIST.800-171.r2 3.13.6

Category: Protect > Secure Network Configuration

Severity: Medium

Resource type: `AWS::NetworkFirewall::RuleGroup`

AWS Config rule: netfw-stateless-rule-group-not-empty

Schedule type: Change triggered

Parameters: None

This control checks if a stateless rule group in AWS Network Firewall contains rules. The control fails if there are no rules in the rule group.

A rule group contains rules that define how your firewall processes traffic in your VPC. An empty stateless rule group, when present in a firewall policy, might give the impression that the rule group will process traffic. However, when the stateless rule group is empty, it does not process traffic.

Remediation

To add rules to your Network Firewall rule group, see Updating a stateful rule group in the *AWS Network Firewall Developer Guide*. On the firewall details page, for **Stateless rule group**, choose **Edit** to add rules.

[NetworkFirewall.7] Network Firewall firewalls should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::NetworkFirewall::Firewall`

AWS Config rule: `tagged-networkfirewall-firewall` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS Network Firewall firewall has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the firewall doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the firewall isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Network Firewall firewall, see Tagging AWS Network Firewall resources in the *AWS Network Firewall Developer Guide*.

[NetworkFirewall.8] Network Firewall firewall policies should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::NetworkFirewall::FirewallPolicy`

AWS Config rule: `tagged-networkfirewall-firewallpolicy` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS Network Firewall firewall policy has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the firewall policy doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the firewall policy isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps

you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Network Firewall policy, see Tagging AWS Network Firewall resources in the *AWS Network Firewall Developer Guide*.

[NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-3, NIST.800-53.r5 SC-5(2)

Category: Protect > Network Security

Severity: Medium

Resource type: `AWS::NetworkFirewall::Firewall`

AWS Config rule: netfw-deletion-protection-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS Network Firewall firewall has deletion protection enabled. The control fails if deletion protection isn't enabled for a firewall.

AWS Network Firewall is a stateful, managed network firewall and intrusion detection service that enables you to inspect and filter traffic to, from, or between your Virtual Private Clouds (VPCs). The deletion protection setting protects against accidental deletion of the firewall.

Remediation

To enable delete protection on an existing Network Firewall firewall, see Updating a firewall in the *AWS Network Firewall Developer Guide*. For **Change protections**, select **Enable**. You can also enable deletion protection by invoking the UpdateFirewallDeleteProtection API and setting the `DeleteProtection` field to `true`.

[NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-3, NIST.800-53.r5 SC-5(2)

Category: Protect > Network Security

Severity: Medium

Resource type: `AWS::NetworkFirewall::Firewall`

AWS Config rule: netfw-subnet-change-protection-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether subnet change protection is enabled for an AWS Network Firewall firewall. The control fails if subnet change protection isn't enabled for the firewall.

AWS Network Firewall is a stateful, managed network firewall and intrusion detection service that you can use to inspect and filter traffic to, from, or between your Virtual Private Clouds (VPCs). If you enable subnet change protection for a Network Firewall firewall, you can protect the firewall against accidental changes to the firewall's subnet associations.

Remediation

For information about enabling subnet change protection for an existing Network Firewall firewall, see Updating a firewall in the *AWS Network Firewall Developer Guide*.

Security Hub controls for Amazon OpenSearch Service

These AWS Security Hub controls evaluate the Amazon OpenSearch Service (OpenSearch Service) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Opensearch.1] OpenSearch domains should have encryption at rest enabled

Related requirements: PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.4, PCI DSS v3.2.1/7.2.1, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-encrypted-at-rest`

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains have encryption-at-rest configuration enabled. The check fails if encryption at rest is not enabled.

For an added layer of security for sensitive data, you should configure your OpenSearch Service domain to be encrypted at rest. When you configure encryption of data at rest, AWS KMS stores and manages your encryption keys. To perform the encryption, AWS KMS uses the Advanced Encryption Standard algorithm with 256-bit keys (AES-256).

To learn more about OpenSearch Service encryption at rest, see Encryption of data at rest for Amazon OpenSearch Service in the *Amazon OpenSearch Service Developer Guide*.

Remediation

To enable encryption at rest for new and existing OpenSearch domains, see Enabling encryption of data at rest in the *Amazon OpenSearch Service Developer Guide*.

[Opensearch.2] OpenSearch domains should not be publicly accessible

Related requirements: PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.2, PCI DSS v3.2.1/1.3.4, PCI DSS v3.2.1/1.3.6, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration > Resources within VPC

Severity: Critical

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-in-vpc-only`

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains are in a VPC. It does not evaluate the VPC subnet routing configuration to determine public access.

You should ensure that OpenSearch domains are not attached to public subnets. See Resource-based policies in the Amazon OpenSearch Service Developer Guide. You should also ensure that your VPC is configured according to the recommended best practices. See Security best practices for your VPC in the Amazon VPC User Guide.

OpenSearch domains deployed within a VPC can communicate with VPC resources over the private AWS network, without the need to traverse the public internet. This configuration increases the security posture by limiting access to the data in transit. VPCs provide a number of network controls to secure access to OpenSearch domains, including network ACL and security groups. Security Hub recommends that you migrate public OpenSearch domains to VPCs to take advantage of these controls.

Remediation

If you create a domain with a public endpoint, you cannot later place it within a VPC. Instead, you must create a new domain and migrate your data. The reverse is also true. If you create a domain within a VPC, it cannot have a public endpoint. Instead, you must either create another domain or disable this control.

For instructions, see Launching your Amazon OpenSearch Service domains within a VPC in the *Amazon OpenSearch Service Developer Guide*.

[Opensearch.3] OpenSearch domains should encrypt data sent between nodes

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2)

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-node-to-node-encryption-check`

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains have node-to-node encryption enabled. This control fails if node-to-node encryption is disabled on the domain.

HTTPS (TLS) can be used to help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle or similar attacks. Only encrypted connections over HTTPS (TLS) should be allowed. Enabling node-to-node encryption for OpenSearch domains ensures that intra-cluster communications are encrypted in transit.

There can be a performance penalty associated with this configuration. You should be aware of and test the performance trade-off before enabling this option.

Remediation

To enable node-to-node encryption on an OpenSearch domain, see Enabling node-to-node encryption in the *Amazon OpenSearch Service Developer Guide*.

[Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-logs-to-cloudwatch`

Schedule type: Change triggered

Parameters:

- `logtype = 'error'` (not customizable)

This control checks whether OpenSearch domains are configured to send error logs to CloudWatch Logs. This control fails if error logging to CloudWatch is not enabled for a domain.

You should enable error logs for OpenSearch domains and send those logs to CloudWatch Logs for retention and response. Domain error logs can assist with security and access audits, and can help to diagnose availability issues.

Remediation

To enable log publishing, see Enabling log publishing (console) in the *Amazon OpenSearch Service Developer Guide*.

[Opensearch.5] OpenSearch domains should have audit logging enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-audit-logging-enabled`

Schedule type: Change triggered

Parameters:

- `cloudWatchLogsLogGroupArnList` (not customizable) – Security Hub does not populate this parameter. Comma-separated list of CloudWatch Logs log groups that should be configured for audit logs.

This control checks whether OpenSearch domains have audit logging enabled. This control fails if an OpenSearch domain does not have audit logging enabled.

Audit logs are highly customizable. They allow you to track user activity on your OpenSearch clusters, including authentication successes and failures, requests to OpenSearch, index changes, and incoming search queries.

Remediation

For instructions on enabling audit logs, see Enabling audit logs in the *Amazon OpenSearch Service Developer Guide*.

[Opensearch.6] OpenSearch domains should have at least three data nodes

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-data-node-fault-tolerance`

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains are configured with at least three data nodes and `zoneAwarenessEnabled` is `true`. This control fails for an OpenSearch domain if `instanceCount` is less than 3 or `zoneAwarenessEnabled` is `false`.

To achieve cluster-level high availability and fault tolerance, an OpenSearch domain should have at least three data nodes. Deploying an OpenSearch domain with at least three data nodes ensures cluster operations if a node fails.

Remediation

To modify the number of data nodes in an OpenSearch domain

1. Sign in to the AWS console and open the Amazon OpenSearch Service console at https://console.aws.amazon.com/aos/.
2. Under **My domains**, choose the name of the domain to edit, and choose **Edit**.

3. Under **Data nodes** set **Number of nodes** to a number greater than 3. If you are deploying to three Availability Zones, set the number to a multiple of three to ensure equal distribution across Availability Zones.
4. Choose **Submit**.

[Opensearch.7] OpenSearch domains should have fine-grained access control enabled

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-5, NIST.800-53.r5 AC-6

Category: Protect > Secure Access Management > Sensitive API actions restricted

Severity: High

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-access-control-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains have fine-grained access control enabled. The control fails if the fine-grained access control is not enabled. Fine-grained access control requires `advanced-security-options`in the OpenSearch parameter `update-domain-config` to be enabled.

Fine-grained access control offers additional ways of controlling access to your data on Amazon OpenSearch Service.

Remediation

To enable fine-grained access control, see Fine-grained access control in Amazon OpenSearch Service in the *Amazon OpenSearch Service Developer Guide*.

[Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy

Related requirements: NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3),

NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-https-required`

Schedule type: Change triggered

Parameters:

- `tlsPolicies: Policy-Min-TLS-1-2-PFS-2023-10` (not customizable)

This controls checks whether an Amazon OpenSearch Service domain endpoint is configured to use the latest TLS security policy. The control fails if the OpenSearch domain endpoint isn't configured to use the latest supported policy or if HTTPs isn't enabled.

HTTPS (TLS) can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. Only encrypted connections over HTTPS (TLS) should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the impact of TLS. TLS 1.2 provides several security enhancements over previous versions of TLS.

Remediation

To enable TLS encryption, use the UpdateDomainConfig API operation. Configure the DomainEndpointOptions field to specify the value for `TLSSecurityPolicy`. For more information, see Node-to-node encryption in the *Amazon OpenSearch Service Developer Guide*.

[Opensearch.9] OpenSearch domains should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `tagged-opensearch-domain` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon OpenSearch Service domain has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the domain doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the domain isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an OpenSearch Service domain, see Working with tags in the *Amazon OpenSearch Service Developer Guide*.

[Opensearch.10] OpenSearch domains should have the latest software update installed

Related requirements: NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5), PCI DSS v4.0.1/6.3.3

Category: Identify > Vulnerability, patch, and version management

Severity: Low

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-update-check`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon OpenSearch Service domain has the latest software update installed. The control fails if a software update is available but not installed for the domain.

OpenSearch Service software updates provide the latest platform fixes, updates, and features available for the environment. Keeping up-to-date with patch installation helps maintain domain security and availability. If no action is taken on required updates, the service software is updated automatically (typically after 2 weeks). We recommend scheduling updates during a time of low traffic to the domain to minimize service disruption.

Remediation

To install software updates for an OpenSearch domain, see Starting an update in the *Amazon OpenSearch Service Developer Guide*.

[Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-2, NIST.800-53.r5 SC-5, NIST.800-53.r5 SC-36, NIST.800-53.r5 SI-13

Category: Recover > Resilience > High availability

Severity: Low

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-primary-node-fault-tolerance`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon OpenSearch Service domain is configured with at least three dedicated primary nodes. The control fails if the domain has fewer than three dedicated primary nodes.

OpenSearch Service uses dedicated primary nodes to increase cluster stability. A dedicated primary node performs cluster management tasks, but doesn't hold data or respond to data upload requests. We recommend that you use multi-AZ with standby, which adds three dedicated primary nodes to each production OpenSearch domain.

Remediation

To change the number of primary nodes for an OpenSearch domain, see Creating and managing Amazon OpenSearch Service domains in the *Amazon OpenSearch Service Developer Guide*.

Security Hub controls for AWS Private CA

These AWS Security Hub controls evaluate the AWS Private Certificate Authority (AWS Private CA) service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[PCA.1] AWS Private CA root certificate authority should be disabled

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Protect > Secure network configuration

Severity: Low

Resource type: `AWS::ACMPCA::CertificateAuthority`

AWS Config rule: `acm-pca-root-ca-disabled`

Schedule type: Periodic

Parameters: None

This control checks if AWS Private CA has a root certificate authority (CA) that is disabled. The control fails if the root CA is enabled.

With AWS Private CA, you can create a CA hierarchy that includes a root CA and subordinate CAs. You should minimize the use of the root CA for daily tasks, especially in production environments. The root CA should only be used to issue certificates for intermediate CAs. This allows the root CA to be stored out of harm's way while the intermediate CAs perform the daily task of issuing end-entity certificates.

Remediation

To disable the root CA, see Update CA status in the *AWS Private Certificate Authority User Guide*.

[PCA.2] AWS Private CA certificate authorities should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::ACMPCA::CertificateAuthority`

AWS Config rule: `acmpca-certificate-authority-tagged`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
			requirements.	

This control checks whether an AWS Private CA certificate authority has tags with the specific keys defined in the parameter `requiredKeyTags`. The control fails if the certificate authority doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredKeyTags`. If the parameter `requiredKeyTags` isn't provided, the control only checks for the existence of a tag key and fails if the certificate authority isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Best practices and strategies in the *Tagging AWS Resources and Tag Editor User Guide*.

Remediation

To add tags to an AWS Private CA authority, see Add tags for your private CA in the *AWS Private Certificate Authority User Guide*.

Security Hub controls for Amazon RDS

These AWS Security Hub controls evaluate the Amazon Relational Database Service (Amazon RDS) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[RDS.1] RDS snapshot should be private

Related requirements: PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.4, PCI DSS v3.2.1/1.3.6, PCI DSS v3.2.1/7.2.1, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration

Severity: Critical

Resource type: `AWS::RDS::DBClusterSnapshot`, `AWS::RDS::DBSnapshot`

AWS Config rule: `rds-snapshots-public-prohibited`

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon RDS snapshots are public. The control fails if RDS snapshots are public. This control evaluates RDS instances, Aurora DB instances, Neptune DB instances, and Amazon DocumentDB clusters.

RDS snapshots are used to back up the data on your RDS instances at a specific point in time. They can be used to restore previous states of RDS instances.

An RDS snapshot must not be public unless intended. If you share an unencrypted manual snapshot as public, this makes the snapshot available to all AWS accounts. This may result in unintended data exposure of your RDS instance.

Note that if the configuration is changed to allow public access, the AWS Config rule may not be able to detect the change for up to 12 hours. Until the AWS Config rule detects the change, the check passes even though the configuration violates the rule.

To learn more about sharing a DB snapshot, see Sharing a DB snapshot in the *Amazon RDS User Guide*.

Remediation

To remove public access from RDS snapshots, see Sharing a snapshot in the *Amazon RDS User Guide*. For **DB snapshot visibility**, we choose **Private**.

[RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration

Related requirements: CIS AWS Foundations Benchmark v3.0.0/2.3.3, NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(5), PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.2, PCI DSS v3.2.1/1.3.4, PCI DSS v3.2.1/1.3.6, PCI DSS v3.2.1/7.2.1, PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration

Severity: Critical

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-instance-public-access-check`

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon RDS instances are publicly accessible by evaluating the `PubliclyAccessible` field in the instance configuration item.

Neptune DB instances and Amazon DocumentDB clusters do not have the `PubliclyAccessible` flag and cannot be evaluated. However, this control can still generate findings for these resources. You can suppress these findings.

The `PubliclyAccessible` value in the RDS instance configuration indicates whether the DB instance is publicly accessible. When the DB instance is configured with `PubliclyAccessible`, it is an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. When the DB instance isn't publicly accessible, it is an internal instance with a DNS name that resolves to a private IP address.

Unless you intend for your RDS instance to be publicly accessible, the RDS instance should not be configured with `PubliclyAccessible` value. Doing so might allow unnecessary traffic to your database instance.

Remediation

To remove public access from RDS DB instances, see Modifying an Amazon RDS DB instance in the *Amazon RDS User Guide*. For **Public access**, choose **No**.

[RDS.3] RDS DB instances should have encryption at-rest enabled

Related requirements: CIS AWS Foundations Benchmark v3.0.0/2.3.1, CIS AWS Foundations Benchmark v1.4.0/2.3.1, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-storage-encrypted`

Schedule type: Change triggered

Parameters: None

This control checks whether storage encryption is enabled for your Amazon RDS DB instances.

This control is intended for RDS DB instances. However, it can also generate findings for Aurora DB instances, Neptune DB instances, and Amazon DocumentDB clusters. If these findings are not useful, then you can suppress them.

For an added layer of security for your sensitive data in RDS DB instances, you should configure your RDS DB instances to be encrypted at rest. To encrypt your RDS DB instances and snapshots at rest, enable the encryption option for your RDS DB instances. Data that is encrypted at rest includes the underlying storage for DB instances, its automated backups, read replicas, and snapshots.

RDS encrypted DB instances use the open standard AES-256 encryption algorithm to encrypt your data on the server that hosts your RDS DB instances. After your data is encrypted, Amazon RDS handles authentication of access and decryption of your data transparently with a minimal impact on performance. You do not need to modify your database client applications to use encryption.

Amazon RDS encryption is currently available for all database engines and storage types. Amazon RDS encryption is available for most DB instance classes. To learn about DB instance classes that

do not support Amazon RDS encryption, see Encrypting Amazon RDS resources in the *Amazon RDS User Guide*.

Remediation

For information about encrypting DB instances in Amazon RDS, see Encrypting Amazon RDS resources in the *Amazon RDS User Guide*.

[RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::RDS::DBClusterSnapshot`, `AWS::RDS::DBSnapshot`

AWS Config rule: `rds-snapshot-encrypted`

Schedule type: Change triggered

Parameters: None

This control checks whether an RDS DB snapshot is encrypted. The control fails if an RDS DB snapshot isn't encrypted.

This control is intended for RDS DB instances. However, it can also generate findings for snapshots of Aurora DB instances, Neptune DB instances, and Amazon DocumentDB clusters. If these findings are not useful, then you can suppress them.

Encrypting data at rest reduces the risk that an unauthenticated user gets access to data that is stored on disk. Data in RDS snapshots should be encrypted at rest for an added layer of security.

Remediation

To encrypt an RDS snapshot, see Encrypting Amazon RDS resources in the *Amazon RDS User Guide*. When you encrypt an RDS DB instance, the encrypted data includes the underlying storage for the instance, its automated backups, read replicas, and snapshots.

You can only encrypt an RDS DB instance when you create it, not after the DB instance is created. However, because you can encrypt a copy of an unencrypted snapshot, you can effectively add encryption to an unencrypted DB instance. That is, you can create a snapshot of your DB instance,

and then create an encrypted copy of that snapshot. You can then restore a DB instance from the encrypted snapshot, and thus you have an encrypted copy of your original DB instance.

[RDS.5] RDS DB instances should be configured with multiple Availability Zones

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-multi-az-support`

Schedule type: Change triggered

Parameters: None

This control checks whether high availability is enabled for your RDS DB instances. The control fails if an RDS DB instance isn't configured with multiple Availability Zones (AZs). This control doesn't apply to RDS DB instances that are part of a Multi-AZ DB cluster deployment.

Configuring Amazon RDS DB instances with AZs helps ensure the availability of stored data. Multi-AZ deployments allow for automated failover if there is an issue with AZ availability and during regular RDS maintenance.

Remediation

To deploy your DB instances in multiple AZs, Modifying a DB instance to be a Multi-AZ DB instance deployment in the *Amazon RDS User Guide*.

[RDS.6] Enhanced monitoring should be configured for RDS DB instances

Related requirements: NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-2

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-enhanced-monitoring-enabled`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value	
`monitoringInterval`	Number of seconds between monitoring metric collection intervals	Enum	`1`, `5`, `10`, `15`, `30`, `60`	No default value	

This control checks whether enhanced monitoring is enabled for an Amazon Relational Database Service (Amazon RDS) DB instance. The control fails if enhanced monitoring isn't enabled for the instance. If you provide a custom value for the `monitoringInterval` parameter, the control passes only if enhanced monitoring metrics are collected for the instance at the specified interval.

In Amazon RDS, Enhanced Monitoring enables a more rapid response to performance changes in underlying infrastructure. These performance changes could result in a lack of availability of the data. Enhanced Monitoring provides real-time metrics of the operating system that your RDS DB instance runs on. An agent is installed on the instance. The agent can obtain metrics more accurately than is possible from the hypervisor layer.

Enhanced Monitoring metrics are useful when you want to see how different processes or threads on a DB instance use the CPU. For more information, see Enhanced Monitoring in the *Amazon RDS User Guide*.

Remediation

For detailed instructions on enabling Enhanced Monitoring for your DB instance, see Setting up for and enabling Enhanced Monitoring in the *Amazon RDS User Guide*.

[RDS.7] RDS clusters should have deletion protection enabled

Related requirements: NIST.800-53.r5 CM-3, NIST.800-53.r5 SC-5(2)

Category: Protect > Data protection > Data deletion protection

Severity: Low

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `rds-cluster-deletion-protection-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an RDS DB cluster has deletion protection enabled. The control fails if an RDS DB cluster doesn't have deletion protection enabled.

This control is intended for RDS DB instances. However, it can also generate findings for Aurora DB instances, Neptune DB instances, and Amazon DocumentDB clusters. If these findings are not useful, then you can suppress them.

Enabling cluster deletion protection is an additional layer of protection against accidental database deletion or deletion by an unauthorized entity.

When deletion protection is enabled, an RDS cluster cannot be deleted. Before a deletion request can succeed, deletion protection must be disabled.

Remediation

To enable deletion protection for an RDS DB cluster, see Modifying the DB cluster by using the console, CLI, and API in the *Amazon RDS User Guide*. For **Deletion protection**, choose **Enable deletion protection**.

[RDS.8] RDS DB instances should have deletion protection enabled

Related requirements: NIST.800-53.r5 CM-3, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Protect > Data protection > Data deletion protection

Severity: Low

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-instance-deletion-protection-enabled`

Schedule type: Change triggered

Parameters:

- `databaseEngines`: `mariadb,mysql,custom-oracle-ee,oracle-ee-cdb,oracle-se2-cdb,oracle-ee,oracle-se2,oracle-se1,oracle-se,postgres,sqlserver-ee,sqlserver-se,sqlserver-ex,sqlserver-web` (not customizable)

This control checks whether your RDS DB instances that use one of the listed database engines have deletion protection enabled. The control fails if an RDS DB instance doesn't have deletion protection enabled.

Enabling instance deletion protection is an additional layer of protection against accidental database deletion or deletion by an unauthorized entity.

While deletion protection is enabled, an RDS DB instance cannot be deleted. Before a deletion request can succeed, deletion protection must be disabled.

Remediation

To enable deletion protection for an RDS DB instance, see Modifying an Amazon RDS DB instance in the *Amazon RDS User Guide*. For **Deletion protection**, choose **Enable deletion protection**.

[RDS.9] RDS DB instances should publish logs to CloudWatch Logs

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-logging-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS DB instance is configured to publish the following logs to Amazon CloudWatch Logs. The control fails if the instance isn’t configured to publish the following logs to CloudWatch Logs:

- Oracle: Alert, Audit, Trace, Listener
- PostgreSQL: Postgresql, Upgrade
- MySQL: Audit, Error, General, SlowQuery
- MariaDB: Audit, Error, General, SlowQuery
- SQL Server: Error, Agent
- Aurora: Audit, Error, General, SlowQuery
- Aurora-MySQL: Audit, Error, General, SlowQuery
- Aurora-PostgreSQL: Postgresql

RDS databases should have relevant logs enabled. Database logging provides detailed records of requests made to RDS. Database logs can assist with security and access audits and can help to diagnose availability issues.

Remediation

For information about publishing RDS database logs to CloudWatch Logs, see Specifying the logs to publish to CloudWatch Logs in the *Amazon RDS User Guide*.

[RDS.10] IAM authentication should be configured for RDS instances

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6

Category: Protect > Secure access management > Passwordless authentication

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-instance-iam-authentication-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an RDS DB instance has IAM database authentication enabled. The control fails if IAM authentication is not configured for RDS DB instances. This control only evaluates RDS instances with the following engine types: `mysql`, `postgres`, `aurora`, `aurora-mysql`, `aurora-postgresql`, and `mariadb`. An RDS instance must also be in one of the following

states for a finding to be generated: `available`, `backing-up`, `storage-optimization`, or `storage-full`.

IAM database authentication allows authentication to database instances with an authentication token instead of a password. Network traffic to and from the database is encrypted using SSL. For more information, see IAM database authentication in the *Amazon Aurora User Guide*.

Remediation

To activate IAM database authentication on an RDS DB instance, see Enabling and disabling IAM database authentication in the *Amazon RDS User Guide*.

[RDS.11] RDS instances should have automatic backups enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-12, NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `db-instance-backup-enabled`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value	
`backupRetentionMinimum`	Minimum backup retention period in days	Integer	7 to 35	7	
`checkReadReplicas`	Checks whether RDS DB instances have backups enabled for read replicas	Boolean	Not customizable	`false`	

This control checks whether an Amazon Relational Database Service instance has automated backups enabled, and a backup retention period greater than or equal to the specified time frame. Read replicas are excluded from evaluation. The control fails if backups aren't enabled for the instance, or if the retention period is less than the specified time frame. Unless you provide a custom parameter value for the backup retention period, Security Hub uses a default value of 7 days.

Backups help you more quickly recover from a security incident and strengthens the resilience of your systems. Amazon RDS lets you configure daily full instance volume snapshots. For more information about Amazon RDS automated backups, see Working with Backups in the *Amazon RDS User Guide*.

Remediation

To enable automated backups on an RDS DB instance, see Enabling automated backups in the *Amazon RDS User Guide*.

[RDS.12] IAM authentication should be configured for RDS clusters

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6

Category: Protect > Secure access management > Passwordless authentication

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `rds-cluster-iam-authentication-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS DB cluster has IAM database authentication enabled.

IAM database authentication allows for password-free authentication to database instances. The authentication uses an authentication token. Network traffic to and from the database is encrypted using SSL. For more information, see IAM database authentication in the *Amazon Aurora User Guide*.

Remediation

To enable IAM authentication for a DB cluster, see Enabling and disabling IAM database authentication in the *Amazon Aurora User Guide*.

[RDS.13] RDS automatic minor version upgrades should be enabled

Related requirements: CIS AWS Foundations Benchmark v3.0.0/2.3.2, NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5), PCI DSS v4.0.1/6.3.3

Category: Identify > Vulnerability, patch, and version management

Severity: High

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-automatic-minor-version-upgrade-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether automatic minor version upgrades are enabled for the RDS database instance.

Automatic minor version upgrades periodically update a database to recent database engine versions. However, the upgrade might not always include the latest database engine version. If you need to keep your databases on specific versions at particular times, we recommend that you manually upgrade to the database versions that you need according to your required schedule. In cases of critical security issues or when a version reaches its end-of-support date, Amazon RDS might apply a minor version upgrade even if you haven't enabled the **Auto minor version upgrade** option. For more information, see the Amazon RDS upgrade documentation for your specific database engine:

- Automatic minor version upgrades for RDS for MariaDB
- Automatic minor version upgrades for RDS for MySQL
- Automatic minor version upgrades for RDS for PostgreSQL
- Db2 on Amazon RDS versions
- Oracle minor version upgrades
- Upgrades of the Microsoft SQL Server DB engine

Remediation

To enable automatic minor version upgrades for an existing DB instance, see Modifying an Amazon RDS DB instance in the *Amazon RDS User Guide*. For **Auto minor version upgrade**, select **Yes**.

[RDS.14] Amazon Aurora clusters should have backtracking enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `aurora-mysql-backtracking-enabled`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value	
`BacktrackWindowInHours`	Number of hours to backtrack an Aurora MySQL cluster	Double	`0.1` to `72`	No default value	

This control checks whether an Amazon Aurora cluster has backtracking enabled. The control fails if the cluster doesn't have backtracking enabled. If you provide a custom value for the `BacktrackWindowInHours` parameter, the control passes only if the cluster is backtracked for the specified length of time.

Backups help you to recover more quickly from a security incident. They also strengthens the resilience of your systems. Aurora backtracking reduces the time to recover a database to a point in time. It does not require a database restore to do so.

Remediation

To enable Aurora backtracking, see Configuring backtracking in the *Amazon Aurora User Guide*.

Note that you cannot enable backtracking on an existing cluster. Instead, you can create a clone that has backtracking enabled. For more information about the limitations of Aurora backtracking, see the list of limitations in Overview of backtracking.

[RDS.15] RDS DB clusters should be configured for multiple Availability Zones

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 SC-36, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `rds-cluster-multi-az-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether high availability is enabled for your RDS DB clusters. The control fails if an RDS DB cluster isn't deployed in multiple Availability Zones (AZs).

RDS DB clusters should be configured for multiple AZs to ensure availability of stored data. Deployment to multiple AZs allows for automated failover in the event of an AZ availability issue and during regular RDS maintenance events.

Remediation

To deploy your DB clusters in multiple AZs, Modifying a DB instance to be a Multi-AZ DB instance deployment in the *Amazon RDS User Guide*.

Remediation steps differ for Aurora global databases. To configure multiple Availability Zones for an Aurora global database, select your DB cluster. Then, choose **Actions** and **Add reader**, and specify multiple AZs. For more information, see Adding Aurora Replicas to a DB cluster in the *Amazon Aurora User Guide*.

[RDS.16] Aurora DB clusters should be configured to copy tags to DB snapshots

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Identify > Inventory

Severity: Low

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `rds-cluster-copy-tags-to-snapshots-enabled` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Aurora DB cluster is configured to automatically copy tags to snapshots of the DB cluster when the snapshots are created. The control fails if the Aurora DB cluster isn't configured to automatically copy tags to snapshots of the cluster when the snapshots are created.

Identification and inventory of your IT assets is a crucial aspect of governance and security. You need to have visibility of all your Amazon Aurora DB clusters so that you can assess their security posture and take action on potential areas of weakness. Aurora DB snapshots should have the same tags as their parent DB clusters. In Amazon Aurora, you can configure a DB cluster to automatically copy all the tags for the cluster to snapshots of the cluster. Enabling this setting ensures that DB snapshots inherit the same tags as their parent DB clusters.

> **Note**
>
> On June 30, 2025, Security Hub changed the title of this control. Previously, the title of this control was: *RDS DB clusters should be configured to copy tags to snapshots*. The new title more accurately reflects that the control checks only Amazon Aurora DB clusters.

Remediation

For information about configuring an Amazon Aurora DB cluster to automatically copy tags to DB snapshots, see Modifying an Amazon Aurora DB cluster in the *Amazon Aurora User Guide*.

[RDS.17] RDS DB instances should be configured to copy tags to snapshots

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2)

Category: Identify > Inventory

Severity: Low

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-instance-copy-tags-to-snapshots-enabled` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether RDS DB instances are configured to copy all tags to snapshots when the snapshots are created.

Identification and inventory of your IT assets is a crucial aspect of governance and security. You need to have visibility of all your RDS DB instances so that you can assess their security posture and take action on potential areas of weakness. Snapshots should be tagged in the same way as their parent RDS database instances. Enabling this setting ensures that snapshots inherit the tags of their parent database instances.

Remediation

To automatically copy tags to snapshots for an RDS DB instance, see Modifying an Amazon RDS DB instance in the *Amazon RDS User Guide*. Select **Copy tags to snapshots**.

[RDS.18] RDS instances should be deployed in a VPC

Category: Protect > Secure network configuration > Resources within VPC

Severity: High

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-deployed-in-vpc` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS instance is deployed on an EC2-VPC.

VPCs provide a number of network controls to secure access to RDS resources. These controls include VPC Endpoints, network ACLs, and security groups. To take advantage of these controls, we recommend that you create your RDS instances on an EC2-VPC.

Remediation

For instructions on moving RDS instances to a VPC, see Updating the VPC for a DB instance in the *Amazon RDS User Guide*.

[RDS.19] Existing RDS event notification subscriptions should be configured for critical cluster events

Related requirements: NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-2

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: `AWS::RDS::EventSubscription`

AWS Config rule: `rds-cluster-event-notifications-configured` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an existing Amazon RDS event subscription for database clusters has notifications enabled for the following source type and event category key-value pairs:

```
DBCluster: ["maintenance","failure"]
```

The control passes if there are no existing event subscriptions in your account.

RDS event notifications uses Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for rapid response. For additional information about RDS event notifications, see Using Amazon RDS event notification in the *Amazon RDS User Guide*.

Remediation

To subscribe to RDS cluster event notifications, see Subscribing to Amazon RDS event notification in the *Amazon RDS User Guide*. Use the following values:

Field	Value
Source type	Clusters
Clusters to include	All clusters
Event categories to include	Select specific event categories or All event categories

[RDS.20] Existing RDS event notification subscriptions should be configured for critical database instance events

Related requirements: NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-2, PCI DSS v4.0.1/11.5.2

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: `AWS::RDS::EventSubscription`

AWS Config rule: `rds-instance-event-notifications-configured` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an existing Amazon RDS event subscription for database instances has notifications enabled for the following source type and event category key-value pairs:

```
DBInstance: ["maintenance","configuration change","failure"]
```

The control passes if there are no existing event subscriptions in your account.

RDS event notifications use Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for rapid response. For additional

information about RDS event notifications, see Using Amazon RDS event notification in the *Amazon RDS User Guide*.

Remediation

To subscribe to RDS instance event notifications, see Subscribing to Amazon RDS event notification in the *Amazon RDS User Guide*. Use the following values:

Field	Value
Source type	Instances
Instances to include	All instances
Event categories to include	Select specific event categories or All event categories

[RDS.21] An RDS event notifications subscription should be configured for critical database parameter group events

Related requirements: NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-2, PCI DSS v4.0.1/11.5.2

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: `AWS::RDS::EventSubscription`

AWS Config rule: `rds-pg-event-notifications-configured` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS event subscription exists with notifications enabled for the following source type, event category key-value pairs. The control passes if there are no existing event subscriptions in your account.

```
DBParameterGroup: ["configuration change"]
```

RDS event notifications use Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for rapid response. For additional information about RDS event notifications, see Using Amazon RDS event notification in the *Amazon RDS User Guide*.

Remediation

To subscribe to RDS database parameter group event notifications, see Subscribing to Amazon RDS event notification in the *Amazon RDS User Guide*. Use the following values:

Field	Value
Source type	Parameter groups
Parameter groups to include	All parameter groups
Event categories to include	Select specific event categories or All event categories

[RDS.22] An RDS event notifications subscription should be configured for critical database security group events

Related requirements: NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-2, PCI DSS v4.0.1/11.5.2

Category: Detect > Detection Services > Application monitoring

Severity: Low

Resource type: `AWS::RDS::EventSubscription`

AWS Config rule: `rds-sg-event-notifications-configured` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS event subscription exists with notifications enabled for the following source type, event category key-value pairs. The control passes if there are no existing event subscriptions in your account.

```
DBSecurityGroup: ["configuration change","failure"]
```

RDS event notifications use Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for a rapid response. For additional information about RDS event notifications, see Using Amazon RDS event notification in the *Amazon RDS User Guide*.

Remediation

To subscribe to RDS instance event notifications, see Subscribing to Amazon RDS event notification in the *Amazon RDS User Guide*. Use the following values:

Field	Value
Source type	Security groups
Security groups to include	All security groups
Event categories to include	Select specific event categories or All event categories

[RDS.23] RDS instances should not use a database engine default port

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(5)

Category: Protect > Secure network configuration

Severity: Low

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-no-default-ports` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an RDS cluster or instance uses a port other than the default port of the database engine. The control fails if the RDS cluster or instance uses the default port. This control doesn't apply to RDS instances that are part of a cluster.

If you use a known port to deploy an RDS cluster or instance, an attacker can guess information about the cluster or instance. The attacker can use this information in conjunction with other information to connect to an RDS cluster or instance or gain additional information about your application.

When you change the port, you must also update the existing connection strings that were used to connect to the old port. You should also check the security group of the DB instance to ensure that it includes an ingress rule that allows connectivity on the new port.

Remediation

To modify the default port of an existing RDS DB instance, see Modifying an Amazon RDS DB instance in the *Amazon RDS User Guide*. To modify the default port of an existing RDS DB cluster, see Modifying the DB cluster by using the console, CLI, and API in the *Amazon Aurora User Guide*. For **Database port**, change the port value to a non-default value.

[RDS.24] RDS Database clusters should use a custom administrator username

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, PCI DSS v4.0.1/2.2.2

Category: Identify > Resource Configuration

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `rds-cluster-default-admin-check`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS database cluster has changed the admin username from its default value. The control does not apply to engines of the type neptune (Neptune DB) or docdb (DocumentDB). This rule will fail if the admin username is set to the default value.

When creating an Amazon RDS database, you should change the default admin username to a unique value. Default usernames are public knowledge and should be changed during RDS database creation. Changing the default usernames reduces the risk of unintended access.

Remediation

For changing the admin username associated with the Amazon RDS database cluster, create a new RDS database cluster and change the default admin username while creating the database.

[RDS.25] RDS database instances should use a custom administrator username

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, PCI DSS v4.0.1/2.2.2

Category: Identify > Resource Configuration

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-instance-default-admin-check`

Schedule type: Change triggered

Parameters: None

This control checks whether you've changed the administrative username for Amazon Relational Database Service (Amazon RDS) database instances from the default value. The control fails if the administrative username is set to the default value. The control doesn't apply to engines of the type neptune (Neptune DB) or docdb (DocumentDB), and to RDS instances that are part of a cluster.

Default administrative usernames on Amazon RDS databases are public knowledge. When creating an Amazon RDS database, you should change the default administrative username to a unique value to reduce the risk of unintended access.

Remediation

To change the administrative username associated with an RDS database instance, first create a new RDS database instance. Change the default administrative username while creating the database.

[RDS.26] RDS DB instances should be protected by a backup plan

Category: Recover > Resilience > Backups enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-12, NIST.800-53.r5 SI-13(5)

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: `rds-resources-protected-by-backup-plan`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value	
`backupVaultLockCheck`	The control produces a PASSED finding if the parameter is set to true and the resource uses AWS Backup Vault Lock.	Boolean	`true` or `false`	No default value	

This control evaluates if Amazon RDS DB instances are covered by a backup plan. This control fails if the RDS DB instance isn't covered by a backup plan. If you set the `backupVaultLockCheck` parameter equal to `true`, the control passes only if the instance is backed up in an AWS Backup locked vault.

AWS Backup is a fully managed backup service that centralizes and automates the backing up of data across AWS services. With AWS Backup, you can create backup policies called backup plans. You can use these plans to define your backup requirements, such as how frequently to back up your data and how long to retain those backups. Including RDS DB instances in a backup plan helps you protect your data from unintended loss or deletion.

Remediation

To add an RDS DB instance to an AWS Backup backup plan, see Assigning resources to a backup plan in the *AWS Backup Developer Guide*.

[RDS.27] RDS DB clusters should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `rds-cluster-encrypted-at-rest`

Schedule type: Change triggered

Parameters: None

This control checks if an RDS DB cluster is encrypted at rest. The control fails if an RDS DB cluster isn't encrypted at rest.

Data at rest refers to any data that's stored in persistent, non-volatile storage for any duration. Encryption helps you protect the confidentiality of such data, reducing the risk that an unauthorized user can access it. Encrypting your RDS DB clusters protects your data and metadata against unauthorized access. It also fulfills compliance requirements for data-at-rest encryption of production file systems.

Remediation

You can enable encryption at rest when you create an RDS DB cluster. You can't change encryption settings after creating a cluster. For more information, see Encrypting an Amazon Aurora DB cluster in the *Amazon Aurora User Guide*.

[RDS.28] RDS DB clusters should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::RDS::DBCluster`

AWS Config rule:`tagged-rds-dbcluster` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon RDS DB cluster has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the DB cluster doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the DB cluster isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an RDS DB cluster, see Tagging Amazon RDS resources in the *Amazon RDS User Guide*.

[RDS.29] RDS DB cluster snapshots should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::RDS::DBClusterSnapshot`

AWS Config rule:`tagged-rds-dbclustersnapshot` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon RDS DB cluster snapshot has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the DB cluster snapshot doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the DB cluster snapshot isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an RDS DB cluster snapshot, see Tagging Amazon RDS resources in the *Amazon RDS User Guide*.

[RDS.30] RDS DB instances should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::RDS::DBInstance`

AWS Config rule:`tagged-rds-dbinstance` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon RDS DB instance has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the DB instance doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the DB instance isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an RDS DB instance, see Tagging Amazon RDS resources in the *Amazon RDS User Guide*.

[RDS.31] RDS DB security groups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::RDS::DBSecurityGroup`

AWS Config rule:`tagged-rds-dbsecuritygroup` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon RDS DB security group has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the DB security group doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the DB security group isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an RDS DB security group, see Tagging Amazon RDS resources in the *Amazon RDS User Guide*.

[RDS.32] RDS DB snapshots should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::RDS::DBSnapshot`

AWS Config rule:`tagged-rds-dbsnapshot` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon RDS DB snapshot has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the DB snapshot doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the DB snapshot isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an RDS DB snapshot, see Tagging Amazon RDS resources in the *Amazon RDS User Guide*.

[RDS.33] RDS DB subnet groups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::RDS::DBSubnetGroup`

AWS Config rule:`tagged-rds-dbsubnetgroups` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon RDS DB subnet group has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the DB subnet group doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the DB subnet group isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an RDS DB subnet group, see Tagging Amazon RDS resources in the *Amazon RDS User Guide*.

[RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `rds-aurora-mysql-audit-logging-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Aurora MySQL DB cluster is configured to publish audit logs to Amazon CloudWatch Logs. The control fails if the cluster isn't configured to publish audit logs to CloudWatch Logs. The control doesn't generate findings for Aurora Serverless v1 DB clusters.

Audit logs capture a record of database activity, including login attempts, data modifications, schema changes, and other events that can be audited for security and compliance purposes. When you configure an Aurora MySQL DB cluster to publish audit logs to a log group in Amazon CloudWatch Logs, you can perform real-time analysis of the log data. CloudWatch Logs retains logs in highly durable storage. You can also create alarms and view metrics in CloudWatch.

> **Note**
>
> An alternative way to publish audit logs to CloudWatch Logs is by enabling advanced auditing and setting the cluster-level DB parameter `server_audit_logs_upload` to 1. The default for the `server_audit_logs_upload parameter` is `0`. However, we recommend you use the following remediation instructions instead to pass this control.

Remediation

To publish Aurora MySQL DB cluster audit logs to CloudWatch Logs, see Publishing Amazon Aurora MySQL logs to Amazon CloudWatch Logs in the *Amazon Aurora User Guide*.

[RDS.35] RDS DB clusters should have automatic minor version upgrade enabled

Related requirements: NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5), PCI DSS v4.0.1/6.3.3

Category: Identify > Vulnerability, patch, and version management

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `rds-cluster-auto-minor-version-upgrade-enable`

Schedule type: Change triggered

Parameters: None

This control checks if automatic minor version upgrade is enabled for an Amazon RDS Multi-AZ DB cluster. The control fails if automatic minor version upgrade isn't enabled for the Multi-AZ DB cluster.

RDS provides automatic minor version upgrade so that you can keep your Multi-AZ DB cluster up to date. Minor versions can introduce new software features, bug fixes, security patches, and performance improvements. By enabling automatic minor version upgrade on RDS database clusters, the cluster, along with the instances in the cluster, will receive automatic updates to the minor version when new versions are available. The updates are applied automatically during the maintenance window.

Remediation

To enable automatic minor version upgrade on Multi-AZ DB clusters, see Modifying a Multi-AZ DB cluster in the *Amazon RDS User Guide*.

[RDS.36] RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs

Related requirements: PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: rds-postgresql-logs-to-cloudwatch

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`logTypes`	Comma-separated list of log types to be published to CloudWatch Logs	StringList	Not customizable	`postgresql`

This control checks whether an Amazon RDS for PostgreSQL DB instance is configured to publish logs to Amazon CloudWatch Logs. The control fails if the PostgreSQL DB instance isn't configured to publish the log types mentioned in the `logTypes` parameter to CloudWatch Logs.

Database logging provides detailed records of requests made to an RDS instance. PostgreSQL generates event logs that contain useful information for administrators. Publishing these logs to CloudWatch Logs centralizes log management and helps you perform real-time analysis of the log data. CloudWatch Logs retains logs in highly durable storage. You can also create alarms and view metrics in CloudWatch.

Remediation

To publish PostgreSQL DB instance logs to CloudWatch Logs, see Publishing PostgreSQL logs to Amazon CloudWatch Logs in the *Amazon RDS User Guide*.

[RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs

Related requirements: PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: rds-aurora-postgresql-logs-to-cloudwatch

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Aurora PostgreSQL DB cluster is configured to publish logs to Amazon CloudWatch Logs. The control fails if the Aurora PostgreSQL DB cluster isn't configured to publish PostgreSQL logs to CloudWatch Logs.

Database logging provides detailed records of requests made to an RDS cluster. Aurora PostgreSQL generates event logs that contain useful information for administrators. Publishing these logs to CloudWatch Logs centralizes log management and helps you perform real-time analysis of the log data. CloudWatch Logs retains logs in highly durable storage. You can also create alarms and view metrics in CloudWatch.

Remediation

To publish Aurora PostgreSQL DB cluster logs to CloudWatch Logs, see Publishing Aurora PostgreSQL logs to Amazon CloudWatch Logs in the *Amazon RDS User Guide*.

[RDS.38] RDS for PostgreSQL DB instances should be encrypted in transit

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: rds-postgres-instance-encrypted-in-transit

Schedule type: Periodic

Parameters: None

This control checks whether a connection to an Amazon RDS for PostgreSQL database (DB) instance is encrypted in transit. The control fails if the `rds.force_ssl` parameter for the parameter group associated with the instance is set to `0` (off). This control doesn't evaluate RDS DB instances that are part of a DB cluster.

Data in transit refers to data that moves from one location to another, such as between nodes in your cluster or between your cluster and your application. Data may move across the internet or within a private network. Encrypting data in transit reduces the risk that an unauthorized user can eavesdrop on network traffic.

Remediation

To require all connections to your RDS for PostgreSQL DB instance to use SSL, see Using SSL with a PostgreSQL DB instance in the *Amazon RDS User Guide*.

[RDS.39] RDS for MySQL DB instances should be encrypted in transit

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: rds-mysql-instance-encrypted-in-transit

Schedule type: Periodic

Parameters: None

This control checks whether a connection to an Amazon RDS for MySQL database (DB) instance is encrypted in transit. The control fails if the `rds.require_secure_transport` parameter for the parameter group associated with the instance is set to `0` (off). This control doesn't evaluate RDS DB instances that are part of a DB cluster.

Data in transit refers to data that moves from one location to another, such as between nodes in your cluster or between your cluster and your application. Data may move across the internet or within a private network. Encrypting data in transit reduces the risk that an unauthorized user can eavesdrop on network traffic.

Remediation

To require all connections to your RDS for MySQL DB instance to use SSL, see SSL/TLS support for MySQL DB instances on Amazon RDS in the *Amazon RDS User Guide*.

[RDS.40] RDS for SQL Server DB instances should publish logs to CloudWatch Logs

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: rds-sql-server-logs-to-cloudwatch

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`logTypes`	A list of the types of logs that an RDS for SQL Server DB instance should be configure d to publish to CloudWatch Logs. This control fails if a DB instance isn't configured to publish a type of log specified in the list.	EnumList (maximum of 2 items)	`agent`, `error`	`agent`, `error`

This control checks whether an Amazon RDS for Microsoft SQL Server DB instance is configured to publish logs to Amazon CloudWatch Logs. The control fails if the RDS for SQL Server DB instance isn't configured to publish logs to CloudWatch Logs. You can optionally specify the types of logs that a DB instance should be configured to publish.

Database logging provides detailed records of requests made to an Amazon RDS DB instance. Publishing logs to CloudWatch Logs centralizes log management and helps you perform real-time analysis of log data. CloudWatch Logs retains logs in highly durable storage. In addition, you can use it to create alarms for specific errors that can occur, such as frequent restarts that are recorded in an error log. Similarly, you can create alarms for errors or warnings that are recorded in SQL Server agent logs related to SQL agent jobs.

Remediation

For information about publishing logs to CloudWatch Logs for an RDS for SQL Server DB instance, see Amazon RDS for Microsoft SQL Server database log files in the *Amazon Relational Database Service User Guide*.

[RDS.41] RDS for SQL Server DB instances should be encrypted in transit

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: rds-sqlserver-encrypted-in-transit

Schedule type: Periodic

Parameters: None

This control checks whether a connection to an Amazon RDS for Microsoft SQL Server DB instance is encrypted in transit. The control fails if the `rds.force_ssl` parameter of the parameter group associated with the DB instance is set to `0 (off)`.

Data in transit refers to data that moves from one location to another, such as between nodes in a DB cluster or between a DB cluster and a client application. Data can move across the internet or within a private network. Encrypting data in transit reduces the risk of unauthorized users eavesdropping on network traffic.

Remediation

For information about enabling SSL/TLS for connections to Amazon RDS DB instances running Microsoft SQL Server, see Using SSL with a Microsoft SQL Server DB Instance in the *Amazon Relational Database Service User Guide*.

[RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: mariadb-publish-logs-to-cloudwatch-logs

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`logTypes`	A list of the types of logs that a MariaDB DB instance should be configured to publish to CloudWatch Logs. The control generates a `FAILED` finding if a DB instance isn't configured to publish a log type specified in the list.	EnumList (maximum of 4 items)	`audit`, `error`, `general`, `slowquery`	`audit`, `error`

This control checks whether an Amazon RDS for MariaDB DB instance is configured to publish certain types of logs to Amazon CloudWatch Logs. The control fails if the MariaDB DB instance isn't configured to publish the logs to CloudWatch Logs. You can optionally specify which types of logs a MariaDB DB instance should be configured to publish.

Database logging provides detailed records of requests made to an Amazon RDS for MariaDB DB instance. Publishing logs to Amazon CloudWatch Logs centralizes log management and helps you perform real-time analysis of the log data. In addition, CloudWatch Logs retains the logs in durable storage, which can support security, access, and availability reviews and audits. With CloudWatch Logs, you can also create alarms and review metrics.

Remediation

For information about configuring an Amazon RDS for MariaDB DB instance to publish logs to Amazon CloudWatch Logs, see Publishing MariaDB logs to Amazon CloudWatch Logs in the *Amazon Relational Database Service User Guide*.

[RDS.44] RDS for MariaDB DB instances should be encrypted in transit

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: rds-mariadb-instance-encrypted-in-transit

Schedule type: Periodic

Parameters: None

This control checks whether connections to an Amazon RDS for MariaDB DB instance are encrypted in transit. The control fails if the DB parameter group associated with the DB instance is not in sync, or the `require_secure_transport` parameter of the parameter group is not set to `ON`.

> **Note**
>
> This control doesn't evaluate Amazon RDS DB instances that use MariaDB versions earlier than version 10.5. The `require_secure_transport` parameter is supported only for MariaDB versions 10.5 and later.

Data in transit refers to data that moves from one location to another, such as between nodes in a DB cluster or between a DB cluster and a client application. Data can move across the internet or within a private network. Encrypting data in transit reduces the risk of unauthorized users eavesdropping on network traffic.

Remediation

For information about enabling SSL/TLS for connections to an Amazon RDS for MariaDB DB instance, see Requiring SSL/TLS for all connections to a MariaDB DB instance in the *Amazon Relational Database Service User Guide*.

[RDS.45] Aurora MySQL DB clusters should have audit logging enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: aurora-mysql-cluster-audit-logging

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon Aurora MySQL DB cluster has audit logging enabled. The control fails if the DB parameter group associated with the DB cluster is not in sync, the `server_audit_logging` parameter is not set to 1, or the `server_audit_events` parameter is set to an empty value.

Database logs can assist with security and access audits and help diagnose availability issues. Audit logs capture a record of database activity, including login attempts, data modifications, schema changes, and other events that can be audited for security and compliance purposes.

Remediation

For information about enabling logging for an Amazon Aurora MySQL DB cluster, see Publishing Amazon Aurora MySQL logs to Amazon CloudWatch Logs in the *Amazon Aurora User Guide*.

[RDS.46] RDS DB instances should not be deployed in public subnets with routes to internet gateways

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::RDS::DBInstance`

AWS Config rule: rds-instance-subnet-igw-check

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon RDS DB instance is deployed in a public subnet that has a route to an internet gateway. The control fails if the RDS DB instance is deployed in a subnet that has a route to an internet gateway and the destination is set to `0.0.0.0/0` or `::/0`.

By provisioning your Amazon RDS resources in private subnets, you can prevent your RDS resources from receiving inbound traffic from the public internet, which can prevent unintended access to your RDS DB instances. If RDS resources are provisioned in a public subnet that is open to the internet, they might be vulnerable to risks such as data exfiltration.

Remediation

For information about provisioning a private subnet for an Amazon RDS DB instance, see Working with a DB instance in a VPC in the *Amazon Relational Database Service User Guide*.

Security Hub controls for Amazon Redshift

These AWS Security Hub controls evaluate the Amazon Redshift service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Redshift.1] Amazon Redshift clusters should prohibit public access

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.2, PCI DSS v3.2.1/1.3.4, PCI DSS v3.2.1/1.3.6, PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: Critical

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: redshift-cluster-public-access-check

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon Redshift clusters are publicly accessible. It evaluates the `PubliclyAccessible` field in the cluster configuration item.

The `PubliclyAccessible` attribute of the Amazon Redshift cluster configuration indicates whether the cluster is publicly accessible. When the cluster is configured with `PubliclyAccessible` set to `true`, it is an Internet-facing instance that has a publicly resolvable DNS name, which resolves to a public IP address.

When the cluster is not publicly accessible, it is an internal instance with a DNS name that resolves to a private IP address. Unless you intend for your cluster to be publicly accessible, the cluster should not be configured with `PubliclyAccessible` set to `true`.

Remediation

To update an Amazon Redshift cluster to disable public access, see Modifying a cluster in the *Amazon Redshift Management Guide*. Set **Publicly accessible** to **No**.

[Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::Redshift::Cluster` `AWS::Redshift::ClusterParameterGroup`

AWS Config rule: redshift-require-tls-ssl

Schedule type: Change triggered

Parameters: None

This control checks whether connections to Amazon Redshift clusters are required to use encryption in transit. The check fails if the Amazon Redshift cluster parameter `require_SSL` isn't set to `True`.

TLS can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. Only encrypted connections over TLS should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the impact of TLS.

Remediation

To update an Amazon Redshift parameter group to require encryption, see Modifying a parameter group in the *Amazon Redshift Management Guide*. Set `require_ssl` to **True**.

[Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-13(5)

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: redshift-backup-enabled

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value	
`MinRetentionPeriod`	Minimum snapshot retention period in days	Integer	7 to 35	7	

This control checks whether an Amazon Redshift cluster has automated snapshots enabled, and a retention period greater than or equal to the specified time frame. The control fails if automated snapshots aren't enabled for the cluster, or if the retention period is less than the specified time frame. Unless you provide a custom parameter value for the snapshot retention period, Security Hub uses a default value of 7 days.

Backups help you to recover more quickly from a security incident. They strengthen the resilience of your systems. Amazon Redshift takes periodic snapshots by default. This control checks whether automatic snapshots are enabled and retained for at least seven days. For more details on Amazon Redshift automated snapshots, see Automated snapshots in the *Amazon Redshift Management Guide*.

Remediation

To update the snapshot retention period for an Amazon Redshift cluster, see Modifying a cluster in the *Amazon Redshift Management Guide*. For **Backup**, set **Snapshot retention** to a value of 7 or greater.

[Redshift.4] Amazon Redshift clusters should have audit logging enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: `redshift-cluster-audit-logging-enabled` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

- `loggingEnabled = true` (not customizable)

This control checks whether an Amazon Redshift cluster has audit logging enabled.

Amazon Redshift audit logging provides additional information about connections and user activities in your cluster. This data can be stored and secured in Amazon S3 and can be helpful in security audits and investigations. For more information, see Database audit logging in the *Amazon Redshift Management Guide*.

Remediation

To configure audit logging for an Amazon Redshift cluster, see Configuring auditing using the console in the *Amazon Redshift Management Guide*.

[Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5)

Category: Identify > Vulnerability, patch, and version management

Severity: Medium

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: redshift-cluster-maintenancesettings-check

Schedule type: Change triggered

Parameters:

- `allowVersionUpgrade = true` (not customizable)

This control checks whether automatic major version upgrades are enabled for the Amazon Redshift cluster.

Enabling automatic major version upgrades ensures that the latest major version updates to Amazon Redshift clusters are installed during the maintenance window. These updates might include security patches and bug fixes. Keeping up to date with patch installation is an important step in securing systems.

Remediation

To remediate this issue from the AWS CLI, use the Amazon Redshift `modify-cluster` command, and set the `--allow-version-upgrade` attribute. *`clustername`* is the name of your Amazon Redshift cluster.

```
aws redshift modify-cluster --cluster-identifier clustername --allow-version-upgrade
```

[Redshift.7] Redshift clusters should use enhanced VPC routing

Related requirements: NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration > API private access

Severity: Medium

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: redshift-enhanced-vpc-routing-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Redshift cluster has `EnhancedVpcRouting` enabled.

Enhanced VPC routing forces all `COPY` and `UNLOAD` traffic between the cluster and data repositories to go through your VPC. You can then use VPC features such as security groups and network access control lists to secure network traffic. You can also use VPC Flow Logs to monitor network traffic.

Remediation

For detailed remediation instructions, see Enabling enhanced VPC routing in the *Amazon Redshift Management Guide*.

[Redshift.8] Amazon Redshift clusters should not use the default Admin username

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Identify > Resource Configuration

Severity: Medium

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: redshift-default-admin-check

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Redshift cluster has changed the admin username from its default value. This control will fail if the admin username for a Redshift cluster is set to `awsuser`.

When creating a Redshift cluster, you should change the default admin username to a unique value. Default usernames are public knowledge and should be changed upon configuration. Changing the default usernames reduces the risk of unintended access.

Remediation

You can't change the admin username for your Amazon Redshift cluster after creating it. To create a new cluster with a non-default username, see Step 1: Create a sample Amazon Redshift cluster in the *Amazon Redshift Getting Started Guide*.

[Redshift.9] Redshift clusters should not use the default database name

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Identify > Resource Configuration

Severity: Medium

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: redshift-default-db-name-check

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Redshift cluster has changed the database name from its default value. The control will fail if the database name for a Redshift cluster is set to `dev`.

When creating a Redshift cluster, you should change the default database name to a unique value. Default names are public knowledge and should be changed upon configuration. As an example, a well-known name could lead to inadvertent access if it was used in IAM policy conditions.

Remediation

You can't change the database name for your Amazon Redshift cluster after it is created. For instructions on creating a new cluster, see Getting started with Amazon Redshift in the *Amazon Redshift Getting Started Guide*.

[Redshift.10] Redshift clusters should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: redshift-cluster-kms-enabled

Schedule type: Change triggered

Parameters: None

This control checks if Amazon Redshift clusters are encrypted at rest. The control fails if a Redshift cluster isn't encrypted at rest or if the encryption key is different from the provided key in the rule parameter.

In Amazon Redshift, you can turn on database encryption for your clusters to help protect data at rest. When you turn on encryption for a cluster, the data blocks and system metadata are encrypted for the cluster and its snapshots. Encryption of data at rest is a recommended best practice because it adds a layer of access management to your data. Encrypting Redshift clusters at rest reduces the risk that an unauthorized user can access the data stored on disk.

Remediation

To modify a Redshift cluster to use KMS encryption, see Changing cluster encryption in the *Amazon Redshift Management Guide*.

[Redshift.11] Redshift clusters should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: `tagged-redshift-cluster` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon Redshift cluster has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the cluster doesn't have any tag keys

or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the cluster isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Redshift cluster, see Tagging resources in Amazon Redshift in the *Amazon Redshift Management Guide*.

[Redshift.12] Redshift event notification subscriptions should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Redshift::EventSubscription`

AWS Config rule: `tagged-redshift-eventsubscription` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon Redshift cluster snapshot has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the cluster snapshot doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the cluster snapshot isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Redshift event notification subscription, see Tagging resources in Amazon Redshift in the *Amazon Redshift Management Guide*.

[Redshift.13] Redshift cluster snapshots should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Redshift::ClusterSnapshot`

AWS Config rule: `tagged-redshift-clustersnapshot` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon Redshift cluster snapshot has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the cluster snapshot doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the cluster snapshot isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging,

you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Redshift cluster snapshot, see Tagging resources in Amazon Redshift in the *Amazon Redshift Management Guide*.

[Redshift.14] Redshift cluster subnet groups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Redshift::ClusterSubnetGroup`

AWS Config rule: `tagged-redshift-clustersubnetgroup` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon Redshift cluster subnet group has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the cluster subnet group doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the cluster subnet group isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Redshift cluster subnet group, see Tagging resources in Amazon Redshift in the *Amazon Redshift Management Guide*.

[Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins

Related requirements: PCI DSS v4.0.1/1.3.1

Category: Protect > Secure network configuration > Security group configuration

Severity: High

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: redshift-unrestricted-port-access

Schedule type: Periodic

Parameters: None

This control checks whether a security group associated with an Amazon Redshift cluster has ingress rules that permit access to the cluster port from the internet (0.0.0.0/0 or ::/0). The control fails if the security group ingress rules permit access to the cluster port from the internet.

Permitting unrestricted inbound access to the Redshift cluster port (IP address with a /0 suffix) can result in unauthorized access or security incidents. We recommend applying the principal of least privilege access when creating security groups and configuring inbound rules.

Remediation

To restrict ingress on the Redshift cluster port to restricted origins, see Work with security group rules in the *Amazon VPC User Guide*. Update rules where the port range matches the Redshift cluster port and the IP port range is 0.0.0.0/0.

[Redshift.16] Redshift cluster subnet groups should have subnets from multiple Availability Zones

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::Redshift::ClusterSubnetGroup`

AWS Config rule: redshift-cluster-subnet-group-multi-az

Schedule type: Change triggered

Parameters: None

The control checks whether an Amazon Redshift cluster subnet group has subnets from more than one Availability Zone (AZ). The control fails if the cluster subnet group doesn't have subnets from at least two different AZs.

Configuring subnets across multiple AZs help ensure that your Redshift data warehouse can continue operating even when failure events occur.

Remediation

To modify a Redshift cluster subnet group to span multiple AZs, see Modifying a cluster subnet group in the *Amazon Redshift Management Guide*.

[Redshift.17] Redshift cluster parameter groups should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Redshift::ClusterParameterGroup`

AWS Config rule: redshift-cluster-parameter-group-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Redshift cluster parameter group has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the parameter group doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the parameter group doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use

tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an Amazon Redshift cluster parameter group, see Tag resources in Amazon Redshift in the *Amazon Redshift Management Guide*.

[Redshift.18] Redshift clusters should have Multi-AZ deployments enabled

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::Redshift::Cluster`

AWS Config rule: redshift-cluster-multi-az-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether multiple Availability Zones (Multi-AZ) deployments are enabled for an Amazon Redshift cluster. The control fails if Multi-AZ deployments aren't enabled for the Amazon Redshift cluster.

Amazon Redshift supports multiple Availability Zones (Multi-AZ) deployments for provisioned clusters. If Multi-AZ deployments are enabled for a cluster, an Amazon Redshift data warehouse can continue operating in failure scenarios when an unexpected event happens in an Availability Zone (AZ). A Multi-AZ deployment deploys compute resources in more than one AZ and these compute resources can be accessed through a single endpoint. In the event of an entire AZ failure, the remaining compute resources in another AZ are available to continue processing workloads.

You can convert an existing Single-AZ data warehouse to a Multi-AZ data warehouse. Additional compute resources are then provisioned in a second AZ.

Remediation

For information about configuring Multi-AZ deployments for an Amazon Redshift cluster, see Converting a Single-AZ data warehouse to a Multi-AZ data warehouse in the *Amazon Redshift Management Guide*.

Security Hub controls for Amazon Redshift Serverless

These AWS Security Hub controls evaluate the Amazon Redshift Serverless service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing

Category: Protect > Secure network configuration > Resources within VPC

Severity: High

Resource type: `AWS::RedshiftServerless::Workgroup`

AWS Config rule: redshift-serverless-workgroup-routes-within-vpc

Schedule type: Periodic

Parameters: None

This control checks whether enhanced VPC routing is enabled for an Amazon Redshift Serverless workgroup. The control fails if enhanced VPC routing is disabled for the workgroup.

If enhanced VPC routing is disabled for an Amazon Redshift Serverless workgroup, Amazon Redshift routes traffic through the internet, including traffic to other services within the AWS network. If you enable enhanced VPC routing for a workgroup, Amazon Redshift forces all `COPY` and `UNLOAD` traffic between your cluster and your data repositories through your virtual private cloud (VPC) based on the Amazon VPC service. With enhanced VPC routing, you can use standard VPC features to control the flow of data between your Amazon Redshift cluster and other resources. This includes features such as VPC security groups and endpoint policies, network access control lists (ACLs), and Domain Name System (DNS) servers. You can also use VPC flow logs to monitor `COPY` and `UNLOAD` traffic.

Remediation

For more information about enhanced VPC routing and how to enable it for a workgroup, see Controlling network traffic with Redshift enhanced VPC routing in the *Amazon Redshift Management Guide*.

[RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::RedshiftServerless::Workgroup`

AWS Config rule: redshift-serverless-workgroup-encrypted-in-transit

Schedule type: Periodic

Parameters: None

This control checks whether connections to an Amazon Redshift Serverless workgroup are required to encrypt data in transit. The control fails if the `require_ssl` configuration parameter for the workgroup is set to `false`.

An Amazon Redshift Serverless workgroup is a collection of compute resources that groups together compute resources like RPUs, VPC subnet groups, and security groups. Properties of a workgroup include network and security settings. These settings specify whether connections to a workgroup should be required to use SSL to encrypt data in transit.

Remediation

For information about updating the settings for an Amazon Redshift Serverless workgroup to require SSL connections, see Connecting to Amazon Redshift Serverless in the *Amazon Redshift Management Guide*.

[RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::RedshiftServerless::Workgroup`

AWS Config rule: redshift-serverless-workgroup-no-public-access

Schedule type: Periodic

Parameters: None

This control checks whether public access is disabled for an Amazon Redshift Serverless workgroup. It evaluates the `publiclyAccessible` property of a Redshift Serverless workgroup. The control fails if public access is enabled (`true`) for the workgroup.

The public access (`publiclyAccessible`) setting for an Amazon Redshift Serverless workgroup specifies whether the workgroup can be accessed from a public network. If public access is enabled (`true`) for a workgroup, Amazon Redshift creates an Elastic IP address that makes the workgroup publicly accessible from outside the VPC. If you don't want a workgroup to be publicly accessible, disable public access for it.

Remediation

For information about changing the public access setting for an Amazon Redshift Serverless workgroup, see Viewing the properties for a workgroup in the *Amazon Redshift Management Guide*.

[RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys

Related requirements: NIST.800-53.r5 AU-9, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SC-12(2), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::RedshiftServerless::Namespace`

AWS Config rule: redshift-serverless-namespace-cmk-encryption

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`kmsKeyArns`	A list of Amazon Resource Names (ARNs) of AWS KMS keys to include in the evaluation. The control generates a `FAILED` finding if a Redshift Serverless namespace isn't encrypted with a KMS key in the list.	StringList (maximum of 3 items)	1–3 ARNs of existing KMS keys. For example: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab` .	No default value

This control checks whether an Amazon Redshift Serverless namespace is encrypted at rest with a customer managed AWS KMS key. The control fails if the Redshift Serverless namespace isn't encrypted with a customer managed KMS key. You can optionally specify a list of KMS keys for the control to include in the evaluation.

In Amazon Redshift Serverless, a namespace defines a logical container for database objects. This control periodically checks whether the encryption settings for a namespace specify a customer managed AWS KMS key, instead of an AWS managed KMS key, for encryption of data in the namespace. With a customer managed KMS key, you have full control of the key. This includes defining and maintaining the key policy, managing grants, rotating cryptographic material, assigning tags, creating aliases, and enabling and disabling the key.

Remediation

For information about updating the encryption settings for an Amazon Redshift Serverless namespace and specifying a customer managed AWS KMS key, see Changing the AWS KMS key for a namespace in the *Amazon Redshift Management Guide*.

[RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username

Category: Identify > Resource configuration

Severity: Medium

Resource type: `AWS::RedshiftServerless::Namespace`

AWS Config rule: redshift-serverless-default-admin-check

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`validAdminUserNames`	A list of admin usernames that Redshift Serverless namespaces should use. The control generates a `FAILED` finding if a namespace uses an admin username that isn't in the list. The list cannot specify the default value, `admin`.	StringList (maximum of 6 items)	1–6 valid admin usernames for Redshift Serverless namespaces.	No default value

This control checks whether the admin username for an Amazon Redshift Serverless namespace is the default admin username, `admin`. The control fails if the admin username for the Redshift

Serverless namespace is `admin`. You can optionally specify a list of admin usernames for the control to include in the evaluation.

When creating an Amazon Redshift Serverless namespace, you should specify a custom admin username for the namespace. The default admin username is public knowledge. By specifying a custom admin username, you can, for example, help mitigate the risk or effectiveness of brute force attacks against the namespace.

Remediation

You can change the admin username for an Amazon Redshift Serverless namespace by using the Amazon Redshift Serverless console or API. To change it by using the console, choose the namespace configuration, and then choose **Edit admin credentials** on the **Actions** menu. To change it programmatically, use the UpdateNamespace operation or, if you're using the AWS CLI, run the update-namespace command. If you change the admin username, you must also change the admin password at the same time.

[RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::RedshiftServerless::Namespace`

AWS Config rule: redshift-serverless-publish-logs-to-cloudwatch

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon Redshift Serverless namespace is configured to export connection and user logs to Amazon CloudWatch Logs. The control fails if the Redshift Serverless namespace isn't configured to export the logs to CloudWatch Logs.

If you configure Amazon Redshift Serverless to export connection log (`connectionlog`) and user log (`userlog`) data to a log group in Amazon CloudWatch Logs, you can collect and store your log records in durable storage, which can support security, access, and availability reviews and audits. With CloudWatch Logs, you can also perform real-time analysis of log data and use CloudWatch to create alarms and review metrics.

Remediation

To export log data for an Amazon Redshift Serverless namespace to Amazon CloudWatch Logs, the respective logs must be selected for export in the audit logging configuration settings for the namespace. For information about updating these settings, see Editing security and encryption in the *Amazon Redshift Management Guide*.

[RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Identify > Resource configuration

Severity: Medium

Resource type: `AWS::RedshiftServerless::Namespace`

AWS Config rule: redshift-serverless-default-db-name-check

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon Redshift Serverless namespace uses the default database name, `dev`. The control fails if the Redshift Serverless namespace uses the default database name, `dev`.

When creating an Amazon Redshift Serverless namespace, you should specify a unique, custom value for the database name and not use the default database name, which is `dev`. The default database name is public knowledge. By specifying a different database name, you can mitigate risks such as unauthorized users inadvertently gaining access to data in the namespace.

Remediation

You can't change the database name for an Amazon Redshift Serverless namespace after you create the namespace. You can, however, specify a custom database name for a Redshift Serverless namespace when you create the namespace. For information about creating a namespace, see Workgroups and namespaces in the *Amazon Redshift Management Guide*.

Security Hub controls for Route 53

These AWS Security Hub controls evaluate the Amazon Route 53 service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Route53.1] Route 53 health checks should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Route53::HealthCheck`

AWS Config rule:`tagged-route53-healthcheck` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon Route 53 health check has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the health check doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the health check isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which

defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Route 53 health check, see Naming and tagging health checks in the *Amazon Route 53 Developer Guide*.

[Route53.2] Route 53 public hosted zones should log DNS queries

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::Route53::HostedZone`

AWS Config rule: `route53-query-logging-enabled`

Schedule type: Change triggered

Parameters: None

This control checks if DNS query logging is enabled for an Amazon Route 53 public hosted zone. The control fails if DNS query logging isn't enabled for a Route 53 public hosted zone.

Logging DNS queries for a Route 53 hosted zone addresses DNS security and compliance requirements and grants visibility. The logs include information such as the domain or subdomain

that was queried, the date and time of the query, the DNS record type (for example, A or AAAA), and the DNS response code (for example, `NoError` or `ServFail`). When DNS query logging is enabled, Route 53 publishes the log files to Amazon CloudWatch Logs.

Remediation

To log DNS queries for Route 53 public hosted zones, see Configuring logging for DNS queries in the *Amazon Route 53 Developer Guide*.

Security Hub controls for Amazon S3

These AWS Security Hub controls evaluate the Amazon Simple Storage Service (Amazon S3) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[S3.1] S3 general purpose buckets should have block public access settings enabled

Related requirements: CIS AWS Foundations Benchmark v3.0.0/2.1.4, CIS AWS Foundations Benchmark v1.4.0/2.1.5, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.2, PCI DSS v3.2.1/1.3.4, PCI DSS v3.2.1/1.3.6, PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: s3-account-level-public-access-blocks-periodic

Schedule type: Periodic

Parameters:

- `ignorePublicAcls`: `true` (not customizable)
- `blockPublicPolicy`: `true` (not customizable)
- `blockPublicAcls`: `true` (not customizable)

- `restrictPublicBuckets`: `true` (not customizable)

This control checks whether the preceding Amazon S3 block public access settings are configured at the account level for an S3 general purpose bucket. The control fails if one or more of the block public access settings are set to `false`.

The control fails if any of the settings are set to `false`, or if any of the settings are not configured.

Amazon S3 public access block is designed to provide controls across an entire AWS account or at the individual S3 bucket level to ensure that objects never have public access. Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both.

Unless you intend to have your S3 buckets be publicly accessible, you should configure the account level Amazon S3 Block Public Access feature.

To learn more, see Using Amazon S3 Block Public Access in the *Amazon Simple Storage Service User Guide*.

Remediation

To enable Amazon S3 Block Public Access for your AWS account, see Configuring block public access settings for your account in the *Amazon Simple Storage Service User Guide*.

[S3.2] S3 general purpose buckets should block public read access

Related requirements: PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.2, PCI DSS v3.2.1/1.3.6, PCI DSS v3.2.1/7.2.1, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration

Severity: Critical

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-public-read-prohibited

Schedule type: Periodic and change triggered

Parameters: None

This control checks whether an Amazon S3 general purpose bucket permits public read access. It evaluates the block public access settings, the bucket policy, and the bucket access control list (ACL). The control fails if the bucket permits public read access.

> **Note**
>
> If an S3 bucket has a bucket policy, this control doesn't evaluate policy conditions that use wildcard characters or variables. To produce a PASSED finding, conditions in the bucket policy must only use fixed values, which are values that don't contain wildcard characters or policy variables. For information about policy variables, see Variables and tags in the *AWS Identity and Access Management User Guide*.

Some use cases may require that everyone on the internet be able to read from your S3 bucket. However, those situations are rare. To ensure the integrity and security of your data, your S3 bucket should not be publicly readable.

Remediation

To block public read access on your Amazon S3 buckets, see Configuring block public access settings for your S3 buckets in the *Amazon Simple Storage Service User Guide*.

[S3.3] S3 general purpose buckets should block public write access

Related requirements: PCI DSS v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.2, PCI DSS v3.2.1/1.3.4, PCI DSS v3.2.1/1.3.6, PCI DSS v3.2.1/7.2.1, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration

Severity: Critical

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-public-write-prohibited

Schedule type: Periodic and change triggered

Parameters: None

This control checks whether an Amazon S3 general purpose bucket permits public write access. It evaluates the block public access settings, the bucket policy, and the bucket access control list (ACL). The control fails if the bucket permits public write access.

> **Note**
>
> If an S3 bucket has a bucket policy, this control doesn't evaluate policy conditions that use wildcard characters or variables. To produce a PASSED finding, conditions in the bucket policy must only use fixed values, which are values that don't contain wildcard characters or policy variables. For information about policy variables, see Variables and tags in the *AWS Identity and Access Management User Guide*.

Some use cases require that everyone on the internet be able to write to your S3 bucket. However, those situations are rare. To ensure the integrity and security of your data, your S3 bucket should not be publicly writable.

Remediation

To block public write access on your Amazon S3 buckets, see Configuring block public access settings for your S3 buckets in the *Amazon Simple Storage Service User Guide*.

[S3.5] S3 general purpose buckets should require requests to use SSL

Related requirements: CIS AWS Foundations Benchmark v3.0.0/2.1.1, CIS AWS Foundations Benchmark v1.4.0/2.1.2, NIST.800-53.r5 AC-17(2), NIST.800-53.r5 AC-4, NIST.800-53.r5 IA-5(1), NIST.800-53.r5 SC-12(3), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-23, NIST.800-53.r5 SC-23(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8, NIST.800-53.r5 SC-8(1), NIST.800-53.r5 SC-8(2), NIST.800-53.r5 SI-7(6), NIST.800-171.r2 3.13.8, NIST.800-171.r2 3.13.15, PCI DSS v3.2.1/4.1, PCI DSS v4.0.1/4.2.1

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-ssl-requests-only

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon S3 general purpose bucket has a policy that requires requests to use SSL. The control fails if the bucket policy doesn't require requests to use SSL.

S3 buckets should have policies that require all requests (`Action: S3:*`) to only accept transmission of data over HTTPS in the S3 resource policy, indicated by the condition key `aws:SecureTransport`.

Remediation

To update an Amazon S3 bucket policy to deny nonsecure transport, see Adding a bucket policy by using the Amazon S3 console in the *Amazon Simple Storage Service User Guide*.

Add a policy statement similar to the one in the following policy. Replace `amzn-s3-demo-bucket` with the name of the bucket you're modifying.

JSON

```
{
    "Id": "ExamplePolicy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSSLRequestsOnly",
            "Action": "s3:*",
            "Effect": "Deny",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "Bool": {
                     "aws:SecureTransport": "false"
                }
            },
           "Principal": "*"
        }
    ]
}
```

For more information, see What S3 bucket policy should I use to comply with the AWS Config rule s3-bucket-ssl-requests-only? in the *AWS Official Knowledge Center*.

[S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-171.r2 3.13.4

Category: Protect > Secure access management > Sensitive API operations actions restricted

Severity: High

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-blacklisted-actions-prohibited

Schedule type: Change triggered

Parameters:

- `blacklistedactionpatterns`: `s3:DeleteBucketPolicy, s3:PutBucketAcl, s3:PutBucketPolicy, s3:PutEncryptionConfiguration, s3:PutObjectAcl` (not customizable)

This control checks whether an Amazon S3 general purpose bucket policy prevents principals from other AWS accounts from performing denied actions on resources in the S3 bucket. The control fails if the bucket policy allows one or more of the preceding actions for a principal in another AWS account.

Implementing least privilege access is fundamental to reducing security risk and the impact of errors or malicious intent. If an S3 bucket policy allows access from external accounts, it could result in data exfiltration by an insider threat or an attacker.

The `blacklistedactionpatterns` parameter allows for successful evaluation of the rule for S3 buckets. The parameter grants access to external accounts for action patterns that are not included in the `blacklistedactionpatterns` list.

Remediation

To update an Amazon S3 bucket policy to remove permissions, see.Adding a bucket policy by using the Amazon S3 console in the *Amazon Simple Storage Service User Guide*.

On the **Edit bucket policy** page, in the policy editing text box, take one of the following actions:

- Remove the statements that grant other AWS accounts access to denied actions.
- Remove the permitted denied actions from the statements.

[S3.7] S3 general purpose buckets should use cross-Region replication

Related requirements: PCI DSS v3.2.1/2.2, NIST.800-53.r5 AU-9(2), NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-36(2), NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Protect > Secure access management

Severity: Low

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-cross-region-replication-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon S3 general purpose bucket has cross-Region replication enabled. The control fails if the bucket doesn't have cross-Region replication enabled.

Replication is the automatic, asynchronous copying of objects across buckets in the same or different AWS Regions. Replication copies newly created objects and object updates from a source bucket to a destination bucket or buckets. AWS best practices recommend replication for source and destination buckets that are owned by the same AWS account. In addition to availability, you should consider other systems hardening settings.

This control produces a `FAILED` finding for a replication destination bucket if it doesn't have cross-region replication enabled. If there's a legitimate reason that the destination bucket doesn't need cross-region replication to be enabled, you can suppress findings for this bucket.

Remediation

To enable Cross-Region Replication on an S3 bucket, see Configuring replication for source and destination buckets owned by the same account in the *Amazon Simple Storage Service User Guide*. For **Source bucket**, choose **Apply to all objects in the bucket**.

[S3.8] S3 general purpose buckets should block public access

Related requirements: CIS AWS Foundations Benchmark v3.0.0/2.1.4, CIS AWS Foundations Benchmark v1.4.0/2.1.5, NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Secure access management > Access control

Severity: High

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-level-public-access-prohibited

Schedule type: Change triggered

Parameters:

- `excludedPublicBuckets` (not customizable) – A comma-separated list of known allowed public S3 bucket names

This control checks whether an Amazon S3 general purpose bucket blocks public access at the bucket level. The control fails if any of the following settings are set to `false`:

- `ignorePublicAcls`
- `blockPublicPolicy`
- `blockPublicAcls`
- `restrictPublicBuckets`

Block Public Access at the S3 bucket level provides controls to ensure that objects never have public access. Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both.

Unless you intend to have your S3 buckets publicly accessible, you should configure the bucket level Amazon S3 Block Public Access feature.

Remediation

For information on how to remove public access at a bucket level, see Blocking public access to your Amazon S3 storage in the *Amazon S3 User Guide*.

[S3.9] S3 general purpose buckets should have server access logging enabled

Related requirements: NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8), NIST.800-171.r2 3.3.8, PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-logging-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether server access logging is enabled for an Amazon S3 general purpose bucket. The control fails if server access logging isn't enabled. When logging is enabled, Amazon S3 delivers access logs for a source bucket to a chosen target bucket. The target bucket must be in the same AWS Region as the source bucket and must not have a default retention period configured. The target logging bucket does not need to have server access logging enabled, and you should suppress findings for this bucket.

Server access logging provides detailed records of requests made to a bucket. Server access logs can assist in security and access audits. For more information, see Security Best Practices for Amazon S3: Enable Amazon S3 server access logging.

Remediation

To enable Amazon S3 server access logging, see Enabling Amazon S3 server access logging in the *Amazon S3 User Guide*.

[S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-version-lifecycle-policy-check

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon S3 general purpose versioned bucket has a Lifecycle configuration. The control fails if the bucket doesn't have a Lifecycle configuration.

We recommended creating a Lifecycle configuration for your S3 bucket to help you define actions that you want Amazon S3 to take during an object's lifetime.

Remediation

For more information on configuring lifecycle on an Amazon S3 bucket, see Setting lifecycle configuration on a bucket and Managing your storage lifecycle.

[S3.11] S3 general purpose buckets should have event notifications enabled

Related requirements: NIST.800-53.r5 CA-7, NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4, NIST.800-53.r5 SI-4(4), NIST.800-171.r2 3.3.8

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-event-notifications-enabled

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
eventTypes	List of preferred S3 event types	EnumList (maximum of 28 items)	s3:IntelligentTiering, s3:LifecycleExpiration:*, s3:LifecycleExpiration:Delete, s3:LifecycleExpiration:DeleteMarkerCreated, s3:LifecycleTransition, s3:ObjectAcl:Put, s3:ObjectCreated:*, s3:ObjectCreated:CompleteMultipartUpload, s3:ObjectCreated:Copy, s3:Object	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
			Created:Post, s3:ObjectCreated:Put, s3:ObjectRemoved:*, s3:ObjectRemoved:Delete, s3:ObjectRemoved:DeleteMarkerCreated, s3:ObjectRestore:*, s3:ObjectRestore:Completed, s3:ObjectRestore:Delete, s3:ObjectRestore:Post, s3:ObjectTagging:*, s3:Object	

Parameter	Description	Type	Allowed custom values	Security Hub default value
			Tagging:Delete, s3:ObjectTagging:Put, s3:ReducedRedundancyLostObject, s3:Replication:*, s3:Replication:OperationFailedReplication, s3:Replication:OperationMissedThreshold, s3:Replication:OperationNotTracked, s3:Replication:OperationReplicatedAfterThreshold,	

Parameter	Description	Type	Allowed custom values	Security Hub default value
			`s3:TestEvent`	

This control checks whether S3 Event Notifications are enabled on an Amazon S3 general purpose bucket. The control fails if S3 Event Notifications are not enabled on the bucket. If you provide custom values for the `eventTypes` parameter, the control passes only if event notifications are enabled for the specified types of events.

When you enable S3 Event Notifications, you receive alerts when specific events occur that impact your S3 buckets. For example, you can be notified of object creation, object removal, and object restoration. These notifications can alert relevant teams to accidental or intentional modifications that may lead to unauthorized data access.

Remediation

For information about detecting changes to S3 buckets and objects, see Amazon S3 Event Notifications in the *Amazon S3 User Guide*.

[S3.12] ACLs should not be used to manage user access to S3 general purpose buckets

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6

Category: Protect > Secure access management > Access control

Severity: Medium

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-acl-prohibited

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon S3 general purpose bucket provides user permissions with an access control list (ACL). The control fails if an ACL is configured for managing user access on the bucket.

ACLs are legacy access control mechanisms that predate IAM. Instead of ACLs, we recommend using S3 bucket policies or AWS Identity and Access Management (IAM) policies to manage access to your S3 buckets.

Remediation

To pass this control, you should disable ACLs for your S3 buckets. For instructions, see Controlling ownership of objects and disabling ACLs for your bucket in the *Amazon Simple Storage Service User Guide*.

To create an S3 bucket policy, see Adding a bucket policy by using the Amazon S3 console. To create an IAM user policy on an S3 bucket, see Controlling access to a bucket with user policies.

[S3.13] S3 general purpose buckets should have Lifecycle configurations

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-13(5)

Category: Protect > Data protection

Severity: Low

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-lifecycle-policy-check

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`targetTransitionDays`	Number of days after object creation when objects are transitioned to a specified storage class	Integer	1 to 36500	No default value
`targetExpirationDays`	Number of days after object creation when objects are deleted	Integer	1 to 36500	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
`targetTransitionStorageClass`	Destination S3 storage class type	Enum	`STANDARD_IA`, `INTELLIGENT_TIERING`, `ONEZONE_IA`, `GLACIER`, `GLACIER_IR`, `DEEP_ARCHIVE`	No default value

This control checks whether an Amazon S3 general purpose bucket has a Lifecycle configuration. The control fails if the bucket doesn't have a Lifecycle configuration. If you provide custom values for one or more of the preceding parameters, the control passes only if the policy includes the specified storage class, deletion time, or transition time.

Creating a Lifecycle configuration for your S3 bucket defines actions that you want Amazon S3 to take during an object's lifetime. For example, you can transition objects to another storage class, archive them, or delete them after a specified period of time.

Remediation

For information about configuring lifecycle policies on an Amazon S3 bucket, see Setting lifecycle configuration on a bucket and see Managing your storage lifecycle in the *Amazon S3 User Guide*.

[S3.14] S3 general purpose buckets should have versioning enabled

Category: Protect > Data protection > Data deletion protection

Related requirements: NIST.800-53.r5 AU-9(2), NIST.800-53.r5 CP-10, NIST.800-53.r5 CP-6, NIST.800-53.r5 CP-6(1), NIST.800-53.r5 CP-6(2), NIST.800-53.r5 CP-9, NIST.800-53.r5 SC-5(2), NIST.800-53.r5 SI-12, NIST.800-53.r5 SI-13(5), NIST.800-171.r2 3.3.8

Severity: Low

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-versioning-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon S3 general purpose bucket has versioning enabled. The control fails if versioning is suspended for the bucket.

Versioning keeps multiple variants of an object in the same S3 bucket. You can use versioning to preserve, retrieve, and restore earlier versions of an object stored in your S3 bucket. Versioning helps you recover from both unintended user actions and application failures.

> **Tip**
>
> As the number of objects increases in a bucket because of versioning, you can set up a Lifecycle configuration to automatically archive or delete versioned objects based on rules. For more information, see Amazon S3 Lifecycle Management for Versioned Objects.

Remediation

To use versioning on an S3 bucket, see Enabling versioning on buckets in the *Amazon S3 User Guide*.

[S3.15] S3 general purpose buckets should have Object Lock enabled

Category: Protect > Data protection > Data deletion protection

Related requirements: NIST.800-53.r5 CP-6(2), PCI DSS v4.0.1/10.5.1

Severity: Medium

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-default-lock-enabled

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
mode	S3 Object Lock retention mode	Enum	GOVERNANCE , COMPLIANCE	No default value

This control checks whether an Amazon S3 general purpose bucket has Object Lock enabled. The control fails if Object Lock isn't enabled for the bucket. If you provide a custom value for the mode parameter, the control passes only if S3 Object Lock uses the specified retention mode.

You can use S3 Object Lock to store objects using a write-once-read-many (WORM) model. Object Lock can help prevent objects in S3 buckets from being deleted or overwritten for a fixed amount of time or indefinitely. You can use S3 Object Lock to meet regulatory requirements that require WORM storage, or add an extra layer of protection against object changes and deletion.

Remediation

To configure Object Lock for new and existing S3 buckets, see Configuring S3 Object Lock in the *Amazon S3 User Guide*.

[S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys

Category: Protect > Data Protection > Encryption of data-at-rest

Related requirements: NIST.800-53.r5 SC-12(2), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 CA-9(1), NIST.800-53.r5 SI-7(6), NIST.800-53.r5 AU-9, NIST.800-171.r2 3.8.9, NIST.800-171.r2 3.13.11, NIST.800-171.r2 3.13.16, PCI DSS v4.0.1/3.5.1

Severity: Medium

Resource type: AWS::S3::Bucket

AWS Config rule: s3-default-encryption-kms

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon S3 general purpose bucket is encrypted with an AWS KMS key (SSE-KMS or DSSE-KMS). The control fails if the bucket is encrypted with default encryption (SSE-S3).

Server-side encryption (SSE) is the encryption of data at its destination by the application or service that receives it. Unless you specify otherwise, S3 buckets use Amazon S3 managed keys (SSE-S3) by default for server-side encryption. However, for added control, you can choose to configure buckets to use server-side encryption with AWS KMS keys (SSE-KMS or DSSE-KMS) instead. Amazon S3 encrypts your data at the object level as it writes it to disks in AWS data centers and decrypts it for you when you access it.

Remediation

To encrypt an S3 bucket using SSE-KMS, see Specifying server-side encryption with AWS KMS (SSE-KMS) in the *Amazon S3 User Guide*. To encrypt an S3 bucket using DSSE-KMS, see Specifying dual-layer server-side encryption with AWS KMS keys (DSSE-KMS) in the *Amazon S3 User Guide*.

[S3.19] S3 access points should have block public access settings enabled

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS v4.0.1/1.4.4

Category: Protect > Secure access management > Resource not publicly accessible

Severity: Critical

Resource type: `AWS::S3::AccessPoint`

AWS Config rule: s3-access-point-public-access-blocks

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon S3 access point has block public access settings enabled. The control fails if block public access settings aren't enabled for the access point.

The Amazon S3 Block Public Access feature helps you manage access to your S3 resources at three levels: the account, bucket, and access point levels. The settings at each level can be configured independently, allowing you to have different levels of public access restrictions for your data.

The access point settings can't individually override the more restrictive settings at higher levels (account level or bucket assigned to the access point). Instead, the settings at the access point level are additive, meaning they complement and work alongside the settings at the other levels. Unless you intend an S3 access point to be publicly accessible, you should enable block public access settings.

Remediation

Amazon S3 currently doesn't support changing an access point's block public access settings after the access point has been created. All block public access settings are enabled by default when you create a new access point. We recommend that you keep all settings enabled unless you know that you have a specific need to disable any of them. For more information, see Managing public access to access points in the *Amazon Simple Storage Service User Guide*.

[S3.20] S3 general purpose buckets should have MFA delete enabled

Related requirements: CIS AWS Foundations Benchmark v3.0.0/2.1.2, CIS AWS Foundations Benchmark v1.4.0/2.1.3, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-3, NIST.800-53.r5 SC-5(2)

Category: Protect > Data protection > Data deletion protection

Severity: Low

Resource type: `AWS::S3::Bucket`

AWS Config rule: s3-bucket-mfa-delete-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether multi-factor authentication (MFA) delete is enabled for an Amazon S3 general purpose bucket. The control fails if MFA delete is not enabled for the bucket. The control doesn't produce findings for buckets that have a lifecycle configuration.

If you enable versioning for an S3 general purpose bucket, you can optionally add another layer of security by configuring MFA delete for the bucket. If you do this, the bucket owner must include two forms of authentication in any request to delete a version of an object in the bucket or change the versioning state of the bucket. MFA delete provides added security if, for example, the bucket owner’s security credentials are compromised. MFA delete can also help prevent accidental bucket deletions by requiring the user who initiates the delete action to prove physical possession of

an MFA device with an MFA code, which adds an extra layer of friction and security to the delete action.

> **Note**
>
> This control produces a PASSED finding only if MFA delete is enabled for the S3 general purpose bucket. To enable MFA delete for a bucket, versioning must also be enabled for the bucket. Bucket versioning is a method of storing multiple variations of an S3 object in the same bucket. In addition, only the bucket owner who is logged in as a root user can enable MFA delete and perform delete actions on the bucket. You cannot use MFA delete with a bucket that has a lifecycle configuration.

Remediation

For information about enabling versioning and configuring MFA delete for an S3 bucket, see Configuring MFA delete in the *Amazon Simple Storage Service User Guide*.

[S3.22] S3 general purpose buckets should log object-level write events

Related requirements: CIS AWS Foundations Benchmark v3.0.0/3.8, PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: cloudtrail-all-write-s3-data-event-check

Schedule type: Periodic

Parameters: None

This control checks whether an AWS account has at least one AWS CloudTrail multi-Region trail that logs all write data events for Amazon S3 buckets. The control fails if the account doesn't have a multi-Region trail that logs write data events for S3 buckets.

S3 object-level operations, such as `GetObject`, `DeleteObject`, and `PutObject`, are called data events. By default, CloudTrail doesn't log data events, but you can configure trails to log data events for S3 buckets. When you enable object-level logging for write data events, you can log each individual object (file) access within an S3 bucket. Enabling object-level logging can help you

meet data compliance requirements, perform comprehensive security analysis, monitor specific patterns of user behavior in your AWS account, and take action on object-level API activity within your S3 buckets by using Amazon CloudWatch Events. This control produces a PASSED finding if you configure a multi-Region trail that logs write-only or all types of data events for all S3 buckets.

Remediation

To enable object-level logging for S3 buckets, see Enabling CloudTrail event logging for S3 buckets and objects in the *Amazon Simple Storage Service User Guide*.

[S3.23] S3 general purpose buckets should log object-level read events

Related requirements: CIS AWS Foundations Benchmark v3.0.0/3.9, PCI DSS v4.0.1/10.2.1

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: cloudtrail-all-read-s3-data-event-check

Schedule type: Periodic

Parameters: None

This control checks whether an AWS account has at least one AWS CloudTrail multi-Region trail that logs all read data events for Amazon S3 buckets. The control fails if the account doesn't have a multi-Region trail that logs read data events for S3 buckets.

S3 object-level operations, such as `GetObject`, `DeleteObject`, and `PutObject`, are called data events. By default, CloudTrail doesn't log data events, but you can configure trails to log data events for S3 buckets. When you enable object-level logging for read data events, you can log each individual object (file) access within an S3 bucket. Enabling object-level logging can help you meet data compliance requirements, perform comprehensive security analysis, monitor specific patterns of user behavior in your AWS account, and take action on object-level API activity within your S3 buckets by using Amazon CloudWatch Events. This control produces a PASSED finding if you configure a multi-Region trail that logs read-only or all types of data events for all S3 buckets.

Remediation

To enable object-level logging for S3 buckets, see Enabling CloudTrail event logging for S3 buckets and objects in the *Amazon Simple Storage Service User Guide*.

[S3.24] S3 Multi-Region Access Points should have block public access settings enabled

Related requirements: PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::S3::MultiRegionAccessPoint`

AWS Config rule: `s3-mrap-public-access-blocked` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon S3 Multi-Region Access Point has block public access settings enabled. The control fails when the Multi-Region Access Point doesn't have block public access settings enabled.

Publicly accessible resources can be lead to unauthorized access, data breaches, or exploitation of vulnerabilities. Restricting access through authentication and authorization measures helps to safeguard sensitive information and maintain the integrity of your resources.

Remediation

By default, all Block Public Access settings are enabled for an S3 Multi-Region Access Point. For more information , see Blocking public access with Amazon S3 Multi-Region Access Points in the *Amazon Simple Storage Service User Guide*. You can't change the Block Public Access settings for a Multi-Region Access Point after it has been created.

[S3.25] S3 directory buckets should have lifecycle configurations

Category: Protect > Data Protection

Severity: Low

Resource type: `AWS::S3Express::DirectoryBucket`

AWS Config rule: s3express-dir-bucket-lifecycle-rules-check

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`targetExpirationDays`	The number of days, after object creation, when objects should expire.	Integer	1 to 2147483647	No default value

This control checks whether lifecycle rules are configured for an S3 directory bucket. The control fails if lifecycle rules aren't configured for the directory bucket, or a lifecycle rule for the bucket specifies expiration settings that don't match the parameter value that you optionally specify.

In Amazon S3, a lifecycle configuration is a set of rules that define actions for Amazon S3 to apply to a group of objects in a bucket. For an S3 directory bucket, you can create a lifecycle rule that specifies when objects expire based on age (in days). You can also create a lifecycle rule that deletes incomplete multipart uploads. Unlike other types of S3 buckets, such as general purpose buckets, directory buckets do not support other types of actions for lifecycle rules, such as transitioning objects between storage classes.

Remediation

To define a lifecycle configuration for an S3 directory bucket, create a lifecycle rule for the bucket. For more information, see Creating and managing a lifecycle configuration for your directory bucket in the *Amazon Simple Storage Service User Guide*.

Security Hub controls for SageMaker AI

These AWS Security Hub controls evaluate the Amazon SageMaker AI service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9), PCI DSS

v3.2.1/1.2.1, PCI DSS v3.2.1/1.3.1, PCI DSS v3.2.1/1.3.2, PCI DSS v3.2.1/1.3.4, PCI DSS v3.2.1/1.3.6, PCI DSS v4.0.1/1.4.4

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::SageMaker::NotebookInstance`

AWS Config rule: sagemaker-notebook-no-direct-internet-access

Schedule type: Periodic

Parameters: None

This control checks whether direct internet access is disabled for an SageMaker AI notebook instance. The control fails if the `DirectInternetAccess` field is enabled for the notebook instance.

If you configure your SageMaker AI instance without a VPC, then by default direct internet access is enabled on your instance. You should configure your instance with a VPC and change the default setting to **Disable—Access the internet through a VPC**. To train or host models from a notebook, you need internet access. To enable internet access, your VPC must have either an interface endpoint (AWS PrivateLink) or a NAT gateway and a security group that allows outbound connections. To learn more about how to connect a notebook instance to resources in a VPC, see Connect a notebook instance to resources in a VPC in the *Amazon SageMaker AI Developer Guide*. You should also ensure that access to your SageMaker AI configuration is limited to only authorized users. Restrict IAM permissions that permit users to change SageMaker AI settings and resources.

Remediation

You can't change the internet access setting after creating a notebook instance. Instead, you can stop, delete, and recreate the instance with blocked internet access. To delete a notebook instance that permits direct internet access, see Use notebook instances to build models: Clean up in the *Amazon SageMaker AI Developer Guide*. To recreate a notebook instance that denies internet access, see Create a notebook instance. For **Network, Direct internet access**, choose **Disable—Access the internet through a VPC**.

[SageMaker.2] SageMaker notebook instances should be launched in a custom VPC

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7,

NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration > Resources within VPC

Severity: High

Resource type: `AWS::SageMaker::NotebookInstance`

AWS Config rule: sagemaker-notebook-instance-inside-vpc

Schedule type: Change triggered

Parameters: None

This control checks if an Amazon SageMaker AI notebook instance is launched within a custom virtual private cloud (VPC). This control fails if a SageMaker AI notebook instance is not launched within a custom VPC or if it is launched in the SageMaker AI service VPC.

Subnets are a range of IP addresses within a VPC. We recommend keeping your resources inside a custom VPC whenever possible to ensure secure network protection of your infrastructure. An Amazon VPC is a virtual network dedicated to your AWS account. With an Amazon VPC, you can control the network access and internet connectivity of your SageMaker AI Studio and notebook instances.

Remediation

You can't change the VPC setting after creating a notebook instance. Instead, you can stop, delete, and recreate the instance. For instructions, see Use notebook instances to build models: Clean up in the *Amazon SageMaker AI Developer Guide*.

[SageMaker.3] Users should not have root access to SageMaker notebook instances

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3(15), NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-6, NIST.800-53.r5 AC-6(10), NIST.800-53.r5 AC-6(2)

Category: Protect > Secure access management > Root user access restrictions

Severity: High

Resource type: `AWS::SageMaker::NotebookInstance`

AWS Config rule: sagemaker-notebook-instance-root-access-check

Schedule type: Change triggered

Parameters: None

This control checks whether root access is turned on for an Amazon SageMaker AI notebook instance. The control fails if root access is turned on for a SageMaker AI notebook instance.

In adherence to the principal of least privilege, it is a recommended security best practice to restrict root access to instance resources to avoid unintentionally over provisioning permissions.

Remediation

To restrict root access to SageMaker AI notebook instances, see Control root access to a SageMaker AI notebook instance in the *Amazon SageMaker AI Developer Guide*.

[SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1

Related requirements: NIST.800-53.r5 CP-10, NIST.800-53.r5 SC-5, NIST.800-53.r5 SC-36, NIST.800-53.r5 SA-13

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::SageMaker::EndpointConfig`

AWS Config rule: sagemaker-endpoint-config-prod-instance-count

Schedule type: Periodic

Parameters: None

This control checks whether production variants of an Amazon SageMaker AI endpoint have an initial instance count greater than 1. The control fails if the endpoint's production variants have only 1 initial instance.

Production variants running with an instance count greater than 1 permit multi-AZ instance redundancy managed by SageMaker AI. Deploying resources across multiple Availability Zones is an AWS best practice to provide high availability within your architecture. High availability helps you to recover from security incidents.

> **Note**
>
> This control applies only to instance-based endpoint configuration.

Remediation

For more information about the parameters of endpoint configuration, see Create an endpoint configuration in the *Amazon SageMaker AI Developer Guide*.

[SageMaker.5] SageMaker models should have network isolation enabled

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: Medium

Resource type: `AWS::SageMaker::Model`

AWS Config rule: sagemaker-model-isolation-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon SageMaker AI hosted model has network isolation enabled. The control fails if the `EnableNetworkIsolation` parameter for the hosted model is set to `False`.

SageMaker AI training and deployed inference containers are internet-enabled by default. If you don't want SageMaker AI to provide external network access to your training or inference containers, you can enable network isolation. If you enable network isolation, no inbound or outbound network calls can be made to or from the model container, including calls to or from other AWS services. Additionally, no AWS credentials are made available to the container runtime environment. Enabling network isolation helps prevent unintended access to your SageMaker AI resources from the internet.

> **Note**
>
> On August 13, 2025, Security Hub changed the title and description of this control. The new title and description more accurately reflect that the control checks the setting for

the `EnableNetworkIsolation` parameter of Amazon SageMaker AI hosted models. Previously, the title of this control was: *SageMaker models should block inbound traffic*.

Remediation

For more information about network isolation for SageMaker AI models, see Run training and inference containers in internet-free mode in the *Amazon SageMaker AI Developer Guide*. When you create a model, you can enable network isolation by setting the value for the `EnableNetworkIsolation` parameter to `True`.

[SageMaker.6] SageMaker app image configurations should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::SageMaker::AppImageConfig`

AWS Config rule: sagemaker-app-image-config-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon SageMaker AI app image configuration (`AppImageConfig`) has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the app image configuration doesn't have any tag keys, or it doesn't have all the

keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the app image configuration doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

To add tags to an Amazon SageMaker AI app image configuration (`AppImageConfig`), you can use the AddTags operation of the SageMaker AI API or, if you're using the AWS CLI, run the add-tags command.

[SageMaker.7] SageMaker images should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::SageMaker::Image`

AWS Config rule: sagemaker-image-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon SageMaker AI image has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the image doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the image doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

To add tags to an Amazon SageMaker AI image, you can use the AddTags operation of the SageMaker AI API or, if you're using the AWS CLI, run the add-tags command.

[SageMaker.8] SageMaker notebook instances should run on supported platforms

Category: Detect > Vulnerability, patch, and version management

Severity: Medium

Resource type: `AWS::SageMaker::NotebookInstance`

AWS Config rule: sagemaker-notebook-instance-platform-version

Schedule type: Periodic

Parameters:

- `supportedPlatformIdentifierVersions`: `notebook-al2-v3` (not customizable)

This control checks whether an Amazon SageMaker AI notebook instance is configured to run on a supported platform, based on the platform identifier specified for the notebook instance. The control fails if the notebook instance is configured to run on a platform that's no longer supported.

If the platform for an Amazon SageMaker AI notebook instance is no longer supported, it might not receive security patches, bug fixes, or other types of updates. Notebook instances might continue to function, but they won't receive SageMaker AI security updates or critical bug fixes. You assume the risks associated with using an unsupported platform. For more information, see JupyterLab versioning in the *Amazon SageMaker AI Developer Guide*.

Remediation

For information about the platforms that Amazon SageMaker AI currently supports and how to migrate to them, see Amazon Linux 2 notebook instances in the *Amazon SageMaker AI Developer Guide*.

Security Hub controls for Secrets Manager

These AWS Security Hub controls evaluate the AWS Secrets Manager service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[SecretsManager.1] Secrets Manager secrets should have automatic rotation enabled

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3(15), PCI DSS v4.0.1/8.6.3, PCI DSS v4.0.1/8.3.9

Category: Protect > Secure development

Severity: Medium

Resource type: `AWS::SecretsManager::Secret`

AWS Config rule: `secretsmanager-rotation-enabled-check`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`maximumAllowedRotationFrequency`	Maximum number of days allowed for secret rotation frequency	Integer	1 to 365	No default value

This control checks whether a secret stored in AWS Secrets Manager is configured with automatic rotation. The control fails if the secret isn't configured with automatic rotation. If you provide a custom value for the `maximumAllowedRotationFrequency` parameter, the control passes only if the secret is automatically rotated within the specified window of time.

Secrets Manager helps you improve the security posture of your organization. Secrets include database credentials, passwords, and third-party API keys. You can use Secrets Manager to store secrets centrally, encrypt secrets automatically, control access to secrets, and rotate secrets safely and automatically.

Secrets Manager can rotate secrets. You can use rotation to replace long-term secrets with short-term ones. Rotating your secrets limits how long an unauthorized user can use a compromised secret. For this reason, you should rotate your secrets frequently. To learn more about rotation, see Rotating your AWS Secrets Manager secrets in the *AWS Secrets Manager User Guide*.

Remediation

To turn on automatic rotation for Secrets Manager secrets, see Set up automatic rotation for AWS Secrets Manager secrets using the console in the *AWS Secrets Manager User Guide*. You must choose and configure an AWS Lambda function for rotation.

[SecretsManager.2] Secrets Manager secrets configured with automatic rotation should rotate successfully

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3(15), PCI DSS v4.0.1/8.6.3, PCI DSS v4.0.1/8.3.9

Category: Protect > Secure development

Severity: Medium

Resource type: `AWS::SecretsManager::Secret`

AWS Config rule: `secretsmanager-scheduled-rotation-success-check`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS Secrets Manager secret rotated successfully based on the rotation schedule. The control fails if `RotationOccurringAsScheduled` is `false`. The control only evaluates secrets that have rotation turned on.

Secrets Manager helps you improve the security posture of your organization. Secrets include database credentials, passwords, and third-party API keys. You can use Secrets Manager to store secrets centrally, encrypt secrets automatically, control access to secrets, and rotate secrets safely and automatically.

Secrets Manager can rotate secrets. You can use rotation to replace long-term secrets with short-term ones. Rotating your secrets limits how long an unauthorized user can use a compromised secret. For this reason, you should rotate your secrets frequently.

In addition to configuring secrets to rotate automatically, you should ensure that those secrets rotate successfully based on the rotation schedule.

To learn more about rotation, see Rotating your AWS Secrets Manager secrets in the *AWS Secrets Manager User Guide*.

Remediation

If the automatic rotation fails, then Secrets Manager might have encountered errors with the configuration. To rotate secrets in Secrets Manager, you use a Lambda function that defines how to interact with the database or service that owns the secret.

For help diagnosing and fixing common errors related to secrets rotation, see Troubleshooting AWS Secrets Manager rotation of secrets in the *AWS Secrets Manager User Guide*.

[SecretsManager.3] Remove unused Secrets Manager secrets

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3(15)

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::SecretsManager::Secret`

AWS Config rule: `secretsmanager-secret-unused`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`unusedForDays`	Maximum number of days that a secret can remain unused	Integer	1 to 365	90

This control checks whether an AWS Secrets Manager secret has been accessed within the specified time frame. The control fails if a secret is unused beyond the specified time frame. Unless you provide a custom parameter value for the access period, Security Hub uses a default value of 90 days.

Deleting unused secrets is as important as rotating secrets. Unused secrets can be abused by their former users, who no longer need access to these secrets. Also, as more users get access to a secret,

someone might have mishandled and leaked it to an unauthorized entity, which increases the risk of abuse. Deleting unused secrets helps revoke secret access from users who no longer need it. It also helps to reduce the cost of using Secrets Manager. Therefore, it is essential to routinely delete unused secrets.

Remediation

To delete inactive Secrets Manager secrets, see Delete an AWS Secrets Manager secret in the *AWS Secrets Manager User Guide*.

[SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days

Related requirements: NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-3(15), PCI DSS v4.0.1/8.6.3, PCI DSS v4.0.1/8.3.9

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::SecretsManager::Secret`

AWS Config rule: `secretsmanager-secret-periodic-rotation`

Schedule type: Periodic

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`maxDaysSinceRotation`	Maximum number of days that a secret can remain unchanged	Integer	`1` to `180`	`90`

This control checks whether an AWS Secrets Manager secret is rotated at least once within the specified time frame. The control fails if a secret isn't rotated at least this frequently. Unless you

provide a custom parameter value for the rotation period, Security Hub uses a default value of 90 days.

Rotating secrets can help you to reduce the risk of an unauthorized use of your secrets in your AWS account. Examples include database credentials, passwords, third-party API keys, and even arbitrary text. If you do not change your secrets for a long period of time, the secrets are more likely to be compromised.

As more users get access to a secret, it can become more likely that someone mishandled and leaked it to an unauthorized entity. Secrets can be leaked through logs and cache data. They can be shared for debugging purposes and not changed or revoked once the debugging completes. For all these reasons, secrets should be rotated frequently.

You can configure automatic rotation for secrets in AWS Secrets Manager. With automatic rotation, you can replace long-term secrets with short-term ones, significantly reducing the risk of compromise. We recommend that you configure automatic rotation for your Secrets Manager secrets. For more information, see Rotating your AWS Secrets Manager secrets in the *AWS Secrets Manager User Guide*.

Remediation

To turn on automatic rotation for Secrets Manager secrets, see Set up automatic rotation for AWS Secrets Manager secrets using the console in the *AWS Secrets Manager User Guide*. You must choose and configure an AWS Lambda function for rotation.

[SecretsManager.5] Secrets Manager secrets should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::SecretsManager::Secret`

AWS Config rule: `tagged-secretsmanager-secret` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an AWS Secrets Manager secret has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the secret doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the secret isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Secrets Manager secret, see Tag AWS Secrets Manager secrets in the *AWS Secrets Manager User Guide*.

Security Hub controls for AWS Service Catalog

This AWS Security Hub control evaluates the AWS Service Catalog service and resources. The control might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only

Related requirements: NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-6, NIST.800-53.r5 CM-8, NIST.800-53.r5 SC-7

Category: Protect > Secure access management

Severity: High

Resource type: `AWS::ServiceCatalog::Portfolio`

AWS Config rule: service-catalog-shared-within-organization

Schedule type: Change triggered

Parameters: None

This control checks whether AWS Service Catalog shares portfolios within an organization when the integration with AWS Organizations is enabled. The control fails if portfolios aren't shared within an organization.

Portfolio sharing only within Organizations helps ensure that a portfolio isn't shared with incorrect AWS accounts. To share a Service Catalog portfolio with an account in an organization, Security Hub recommends using `ORGANIZATION_MEMBER_ACCOUNT` instead of `ACCOUNT`. This simplifies administration by governing the access granted to the account across the organization. If you have a business need to share Service Catalog portfolios with an external account, you can automatically suppress the findings from this control or disable it.

Remediation

To enable portfolio sharing with AWS Organizations, see Sharing with AWS Organizations in the *AWS Service Catalog Administrator Guide*.

Security Hub controls for Amazon SES

These AWS Security Hub controls evaluate the Amazon Simple Email Service (Amazon SES) service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[SES.1] SES contact lists should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::SES::ContactList`

AWS Configrule: `tagged-ses-contactlist` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon SES contact list has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the contact list doesn't have any tag keys

or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the contact list isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Amazon SES contact list, see TagResource in the *Amazon SES API v2 Reference*.

[SES.2] SES configuration sets should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::SES::ConfigurationSet`

AWS Configrule: `tagged-ses-configurationset` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an Amazon SES configuration set has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the configuration set doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the configuration set isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Amazon SES configuration set, see TagResource in the *Amazon SES API v2 Reference*.

Security Hub controls for Amazon SNS

These AWS Security Hub controls evaluate the Amazon Simple Notification Service (Amazon SNS) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[SNS.1] SNS topics should be encrypted at-rest using AWS KMS

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6), NIST.800-171.r2 3.13.11, NIST.800-171.r2 3.13.16

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::SNS::Topic`

AWS Config rule: `sns-encrypted-kms`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon SNS topic is encrypted at rest using keys managed in AWS Key Management Service (AWS KMS). The controls fails if the SNS topic doesn't use a KMS key for server-side encryption (SSE). By default, SNS stores messages and files using disk encryption. To pass this control, you must choose to use a KMS key for encryption instead. This adds an additional layer of security and provides more access control flexibility.

Encrypting data at rest reduces the risk of data stored on disk being accessed by a user not authenticated to AWS. API permissions are required to decrypt the data before it can be read. We recommend encrypting SNS topics with KMS keys for an added layer of security.

Remediation

To enable SSE for an SNS topic, see Enabling server-side encryption (SSE) for an Amazon SNS topic in the *Amazon Simple Notification Service Developer Guide*. Before you can use SSE, you must also

configure AWS KMS key policies to allow encryption of topics and encryption and decryption of messages. For more information, see Configuring AWS KMS permissions in the *Amazon Simple Notification Service Developer Guide*.

[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic

> **Important**
>
> Security Hub retired this control in April 2024. For more information, see Change log for Security Hub CSPM controls.

Related requirements: NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::SNS::Topic`

AWS Config rule: `sns-topic-message-delivery-notification-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether logging is enabled for the delivery status of notification messages sent to an Amazon SNS topic for the endpoints. This control fails if the delivery status notification for messages is not enabled.

Logging is an important part of maintaining the reliability, availability, and performance of services. Logging message delivery status helps provide operational insights, such as the following:

- Knowing whether a message was delivered to the Amazon SNS endpoint.
- Identifying the response sent from the Amazon SNS endpoint to Amazon SNS.
- Determining the message dwell time (the time between the publish timestamp and the hand off to an Amazon SNS endpoint).

Remediation

To configure delivery status logging for a topic, see Amazon SNS message delivery status in the *Amazon Simple Notification Service Developer Guide*.

[SNS.3] SNS topics should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::SNS::Topic`

AWS Config rule: `tagged-sns-topic` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon SNS topic has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the topic doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the topic isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other

criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an SNS topic, see Configuring Amazon SNS topic tags in the *Amazon Simple Notification Service Developer Guide*.

[SNS.4] SNS topic access policies should not allow public access

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::SNS::Topic`

AWS Config rule: sns-topic-no-public-access

Schedule type: Change triggered

Parameters: None

This control checks if the Amazon SNS topic access policy allows public access. This control fails if the SNS topic access policy allows public access.

You use an Amazon SNS access policy with a particular topic to restrict who can work with that topic (for example, who can publish messages to it or who can subscribe to it). SNS policies can grant access to other AWS accounts, or to users within your own AWS account. Providing a wildcard

(*) in the `Principal` field of the topic policy and a lack of conditions to limit the topic policy can result in data exfiltration, denial of service, or undesired injection of messages into your service by an attacker.

> **Note**
>
> This control doesn't evaluate policy conditions that use wildcard characters or variables. To produce a PASSED finding, conditions in the Amazon SNS access policy for a topic must only use fixed values, which are values that don't contain wildcard characters or policy variables. For information about policy variables, see Variables and tags in the *AWS Identity and Access Management User Guide*.

Remediation

To update access policies for an SNS topic, see Overview of managing access in Amazon SNS in the *Amazon Simple Notification Service Developer Guide*.

Security Hub controls for Amazon SQS

These AWS Security Hub controls evaluate the Amazon Simple Queue Service (Amazon SQS) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[SQS.1] Amazon SQS queues should be encrypted at rest

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-3(6), NIST.800-53.r5 SC-13, NIST.800-53.r5 SC-28, NIST.800-53.r5 SC-28(1), NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SI-7(6)

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::SQS::Queue`

AWS Config rule: `sqs-queue-encrypted` (custom Security Hub rule)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon SQS queue is encrypted at rest. The control fails if the queue isn't encrypted with an SQS-managed key (SSE-SQS) or an AWS Key Management Service (AWS KMS) key (SSE-KMS).

Encrypting data at rest reduces the risk of an unauthorized user accessing data stored on disk. Server-side encryption (SSE) protects the contents of messages in SQS queues using SQS-managed encryption keys (SSE-SQS) or AWS KMS keys (SSE-KMS).

Remediation

To configure SSE for an SQS queue, see Configuring server-side encryption (SSE) for a queue (console) in the *Amazon Simple Queue Service Developer Guide*.

[SQS.2] SQS queues should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::SQS::Queue`

AWS Config rule: `tagged-sqs-queue` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	`No default value`

This control checks whether an Amazon SQS queue has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the queue doesn't have any tag keys or

if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the queue isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an existing queue using the Amazon SQS console, see Configuring cost allocation tags for an Amazon SQS queue (console) in the *Amazon Simple Queue Service Developer Guide*.

[SQS.3] SQS queue access policies should not allow public access

Category: Protect > Secure access management > Resource not publicly accessible

Severity: High

Resource type: `AWS::SQS::Queue`

AWS Config rule: sqs-queue-no-public-access

Schedule type: Change triggered

Parameters: None

This controls checks whether an Amazon SQS access policy allows public access to an SQS queue. The control fails if an SQS access policy allows public access to the queue.

An Amazon SQS access policy can allow public access to an SQS queue, which might allow an anonymous user or any authenticated AWS IAM identity to access the queue. SQS access policies typically provide this access by specifying the wildcard character (*) in the `Principal` element of the policy, not using proper conditions to restrict access to the queue, or both. If an SQS access policy allows public access, third parties might be able to perform tasks such as receive messages from the queue, send messages to the queue, or modify the access policy for the queue. This could result in events such as data exfiltration, a denial of service, or injection of messages into the queue by a threat actor.

> **Note**
>
> This control doesn't evaluate policy conditions that use wildcard characters or variables. To produce a PASSED finding, conditions in the Amazon SQS access policy for a queue must only use fixed values, which are values that don't contain wildcard characters or policy variables. For information about policy variables, see Variables and tags in the *AWS Identity and Access Management User Guide*.

Remediation

For information about configuring the SQS access policy for an SQS queue, see Using custom policies with the Amazon SQS Access Policy Language in the *Amazon Simple Queue Service Developer Guide*.

Security Hub controls for Step Functions

These AWS Security Hub controls evaluate the AWS Step Functions service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[StepFunctions.1] Step Functions state machines should have logging turned on

Related requirements: PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::StepFunctions::StateMachine`

AWS Config rule: `step-functions-state-machine-logging-enabled`

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`logLevel`	Minimum logging level	Enum	`ALL`, `ERROR`, `FATAL`	No default value

This controls checks whether an AWS Step Functions state machine has logging turned on. The control fails if a state machine doesn't have logging turned on. If you provide a custom value for the `logLevel` parameter, the control passes only if the state machine has the specified logging level turned on.

Monitoring helps you maintain the reliability, availability, and performance of Step Functions. You should collect as much monitoring data from the AWS services that you use so you can more easily debug multi-point failures. Having a logging configuration defined for your Step Functions state machines allows for you to track execution history and results in Amazon CloudWatch Logs. Optionally, you can track only errors or fatal events.

Remediation

To turn on logging for a Step Functions state machine, see Configure logging in the *AWS Step Functions Developer Guide*.

[StepFunctions.2] Step Functions activities should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::StepFunctions::Activity`

AWS Config rule:`tagged-stepfunctions-activity` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource must contain. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Step Functions activity has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the activity doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the activity isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

> **Note**
>
> Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to an Step Functions activity, see Tagging in Step Functions in the *AWS Step Functions Developer Guide*.

Security Hub controls for Systems Manager

These AWS Security Hub controls evaluate the AWS Systems Manager (SSM) service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[SSM.1] Amazon EC2 instances should be managed by AWS Systems Manager

Related requirements: PCI DSS v3.2.1/2.4, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-8, NIST.800-53.r5 CM-8(1), NIST.800-53.r5 CM-8(2), NIST.800-53.r5 CM-8(3), NIST.800-53.r5 SA-15(2), NIST.800-53.r5 SA-15(8), NIST.800-53.r5 SA-3, NIST.800-53.r5 SI-2(3)

Category: Identify > Inventory

Severity: Medium

Evaluated resource: `AWS::EC2::Instance`

Required AWS Config recording resources: `AWS::EC2::Instance`, `AWS::SSM::ManagedInstanceInventory`

AWS Config rule: ec2-instance-managed-by-systems-manager

Schedule type: Change triggered

Parameters: None

This control checks whether the stopped and running EC2 instances in your account are managed by AWS Systems Manager. Systems Manager is an AWS service that you can use to view and control your AWS infrastructure.

To help you maintain security and compliance, Systems Manager scans your stopped and running managed instances. A managed instance is a machine that's configured for use with Systems Manager. Systems Manager then reports or takes corrective action on any policy violations that it detects. Systems Manager also helps you configure and maintain your managed instances. To learn more, see the AWS Systems Manager User Guide.

> **Note**
>
> This control generates `FAILED` findings for EC2 instances that are AWS Elastic Disaster Recovery Replication Server instances managed by AWS. A Replication Server instance is an EC2 Instance that's automatically launched by AWS Elastic Disaster Recovery to support continuous data replication from source servers. AWS intentionally removes the Systems Manager (SSM) Agent from these instances to maintain isolation and help prevent potential unintended access paths.

Remediation

For information about managing EC2 instances with AWS Systems Manager, see Amazon EC2 host management in the *AWS Systems Manager User Guide*. In the **Configuration options** section on the AWS Systems Manager console, you can keep the default settings or change them as necessary for your preferred configuration.

[SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation

Related requirements: NIST.800-53.r5 CM-8(3), NIST.800-53.r5 SI-2, NIST.800-53.r5 SI-2(2), NIST.800-53.r5 SI-2(3), NIST.800-53.r5 SI-2(4), NIST.800-53.r5 SI-2(5), NIST.800-171.r2 3.7.1, PCI DSS v3.2.1/6.2, PCI DSS v4.0.1/2.2.1, PCI DSS v4.0.1/6.3.3

Category: Detect > Detection services

Severity: High

Resource type: `AWS::SSM::PatchCompliance`

AWS Config rule: ec2-managedinstance-patch-compliance-status-check

Schedule type: Change triggered

Parameters: None

This control checks whether the compliance status of Systems Manager patch compliance is `COMPLIANT` or `NON_COMPLIANT` after the patch installation on the instance. The control fails if the compliance status is `NON_COMPLIANT`. The control only checks instances that are managed by Systems Manager Patch Manager.

Patching your EC2 instances as required by your organization reduces the attack surface of your AWS accounts.

Remediation

Systems Manager recommends using patch policies to configure patching for your managed instances. You can also use Systems Manager documents, as described in the following procedure, to patch an instance.

To remediate noncompliant patches

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. For **Node Management**, choose **Run Command**, and then choose **Run command**.
3. Choose the option for **AWS-RunPatchBaseline**.
4. Change the **Operation** to **Install**.
5. Choose **Choose instances manually**, and then choose the noncompliant instances.
6. Choose **Run**.
7. After the command is complete, to monitor the new compliance status of your patched instances, choose **Compliance** in the navigation pane.

[SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2, NIST.800-53.r5 CM-2(2), NIST.800-53.r5 CM-8, NIST.800-53.r5 CM-8(1), NIST.800-53.r5 CM-8(3), NIST.800-53.r5 SI-2(3), PCI DSS v3.2.1/2.4, PCI DSS v4.0.1/2.2.1, PCI DSS v4.0.1/6.3.3

Category: Detect > Detection services

Severity: Low

Resource type: `AWS::SSM::AssociationCompliance`

AWS Config rule: ec2-managedinstance-association-compliance-status-check

Schedule type: Change triggered

Parameters: None

This control checks whether the status of the AWS Systems Manager association compliance is `COMPLIANT` or `NON_COMPLIANT` after the association is run on an instance. The control fails if the association compliance status is `NON_COMPLIANT`.

A State Manager association is a configuration that is assigned to your managed instances. The configuration defines the state that you want to maintain on your instances. For example, an association can specify that antivirus software must be installed and running on your instances or that certain ports must be closed.

After you create one or more State Manager associations, compliance status information is immediately available to you. You can view the compliance status in the console or in response to AWS CLI commands or corresponding Systems Manager API actions. For associations, Configuration Compliance shows the compliance status (`Compliant` or `Non-compliant`). It also shows the severity level assigned to the association, such as `Critical` or `Medium`.

To learn more about State Manager association compliance, see About State Manager association compliance in the *AWS Systems Manager User Guide*.

Remediation

A failed association can be related to different things, including targets and Systems Manager document names. To remediate this issue, you must first identify and investigate the association by viewing association history. For instructions on viewing association history, see Viewing association histories in the *AWS Systems Manager User Guide*.

After investigating, you can edit the association to correct the identified issue. You can edit an association to specify a new name, schedule, severity level, or targets. After you edit an association, AWS Systems Manager creates a new version. For instructions on editing an association, see Editing and creating a new version of an association in the *AWS Systems Manager User Guide*.

[SSM.4] SSM documents should not be public

Related requirements: NIST.800-53.r5 AC-21, NIST.800-53.r5 AC-3, NIST.800-53.r5 AC-3(7), NIST.800-53.r5 AC-4, NIST.800-53.r5 AC-4(21), NIST.800-53.r5 AC-6, NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(20), NIST.800-53.r5 SC-7(21), NIST.800-53.r5 SC-7(3), NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-7(9)

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: Critical

Resource type: `AWS::SSM::Document`

AWS Config rule: ssm-document-not-public

Schedule type: Periodic

Parameters: None

This control checks whether AWS Systems Manager documents that are owned by an account are public. The control fails if Systems Manager documents that have `Self` as the owner are public.

Systems Manager documents that are public might allow unintended access to your documents. A public Systems Manager document can expose valuable information about your account, resources, and internal processes.

Unless your use case requires public sharing, we recommend that you block public sharing for Systems Manager documents that have `Self` as the owner.

Remediation

For information about configuring sharing for Systems Manager documents, see Share an SSM document in the *AWS Systems Manager User Guide*.

[SSM.5] SSM documents should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::SSM::Document`

AWS Config rule: ssm-document-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an	StringList (maximum of 6 items)	1–6 tag keys that meet AWS	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
	evaluated resource. Tag keys are case sensitive.		requirements.	

This control checks whether an AWS Systems Manager document has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the document doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the document doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix. The control doesn't evaluate Systems Manager documents that are owned by Amazon.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

To add tags to an AWS Systems Manager document, you can use the AddTagsToResource operation of the AWS Systems Manager API or, if you're using the AWS CLI, run the add-tags-to-resource command. You can also use the AWS Systems Manager console.

[SSM.6] SSM Automation should have CloudWatch logging enabled

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: ssm-automation-logging-enabled

Schedule type: Periodic

Parameters: None

This control checks whether Amazon CloudWatch logging is enabled for AWS Systems Manager (SSM) Automation. The control fails if CloudWatch logging isn't enabled for SSM Automation.

SSM Automation is an AWS Systems Manager tool that helps you build automated solutions to deploy, configure, and manage AWS resources at scale using predefined or custom runbooks. To meet operational or security requirements for your organization, you might need to provide a record of the scripts that it runs. You can configure SSM Automation to send the output from `aws:executeScript` actions in your runbooks to an Amazon CloudWatch Logs log group that you specify. With CloudWatch Logs, you can monitor, store, and access log files from various AWS services.

Remediation

For information about enabling CloudWatch logging for SSM Automation, see Logging Automation action output with CloudWatch Logs in the *AWS Systems Manager User Guide*.

[SSM.7] SSM documents should have the block public sharing setting enabled

Category: Protect > Secure access management > Resource not publicly accessible

Severity: Critical

Resource type: `AWS::::Account`

AWS Config rule: ssm-automation-block-public-sharing

Schedule type: Periodic

Parameters: None

This control checks whether the block public sharing setting is enabled for AWS Systems Manager documents. The control fails if the block public sharing setting is disabled for Systems Manager documents.

The block public sharing setting for AWS Systems Manager (SSM) documents is an account-level setting. Enabling this setting can prevent unwanted access to your SSM documents. If you enable this setting, your change doesn't affect any SSM documents that you're currently sharing with the public. Unless your use case requires you to share SSM documents with the public, we recommend that you enable the block public sharing setting. The setting can differ for each AWS Region.

Remediation

For information about enabling the block public sharing setting for AWS Systems Manager (SSM) documents, see Block public sharing for SSM documents in the *AWS Systems Manager User Guide*.

Security Hub controls for AWS Transfer Family

These AWS Security Hub controls evaluate the AWS Transfer Family service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[Transfer.1] AWS Transfer Family workflows should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Transfer::Workflow`

AWS Config rule: `tagged-transfer-workflow` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredTagKeys`	List of non-system tag keys that the evaluated resource	StringList (maximum of 6 items)	1–6 tag keys that meet AWS	`No default value`

Parameter	Description	Type	Allowed custom values	Security Hub default value
	must contain. Tag keys are case sensitive.		requirements.	

This control checks whether an AWS Transfer Family workflow has tags with the specific keys defined in the parameter `requiredTagKeys`. The control fails if the workflow doesn't have any tag keys or if it doesn't have all the keys specified in the parameter `requiredTagKeys`. If the parameter `requiredTagKeys` isn't provided, the control only checks for the existence of a tag key and fails if the workflow isn't tagged with any key. System tags, which are automatically applied and begin with `aws:`, are ignored.

A tag is a label that you assign to an AWS resource, and it consists of a key and an optional value. You can create tags to categorize resources by purpose, owner, environment, or other criteria. Tags can help you identify, organize, search for, and filter resources. Tagging also helps you track accountable resource owners for actions and notifications. When you use tagging, you can implement attribute-based access control (ABAC) as an authorization strategy, which defines permissions based on tags. You can attach tags to IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or a separate set of policies for your IAM principals. You can design these ABAC policies to allow operations when the principal's tag matches the resource tag. For more information, see What is ABAC for AWS? in the *IAM User Guide*.

Note

Don't add personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible to many AWS services, including AWS Billing. For more tagging best practices, see Tagging your AWS resources in the *AWS General Reference*.

Remediation

To add tags to a Transfer Family workflow (console)

1. Open the AWS Transfer Family console.
2. In the navigation pane, choose **Workflows**. Then, select the workflow that you want to tag.

3. Choose **Manage tags**, and then add the tags.

[Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection

Related requirements: NIST.800-53.r5 CM-7, NIST.800-53.r5 IA-5, NIST.800-53.r5 SC-8, PCI DSS v4.0.1/4.2.1

Category: Protect > Data Protection > Encryption of data-in-transit

Severity: Medium

Resource type: `AWS::Transfer::Server`

AWS Config rule: transfer-family-server-no-ftp

Schedule type: Periodic

Parameters: None

This control checks whether an AWS Transfer Family server uses a protocol other than FTP for endpoint connection. The control fails if the server uses FTP protocol for a client to connect to the server's endpoint.

FTP (File Transfer Protocol) establishes the endpoint connection through unencrypted channels, leaving data sent over these channels vulnerable to interception. Using SFTP (SSH File Transfer Protocol), FTPS (File Transfer Protocol Secure), or AS2 (Applicability Statement 2) offers an extra layer of security by encrypting your data in transit and can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic.

Remediation

To modify the protocol for a Transfer Family server, see Edit the file transfer protocols in the *AWS Transfer Family User Guide*.

[Transfer.3] Transfer Family connectors should have logging enabled

Related requirements: NIST.800-53.r5 AC-2(12), NIST.800-53.r5 AC-2(4), NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AC-6(9), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 AU-9(7),

NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-3(8), NIST.800-53.r5 SI-4, NIST.800-53.r5 SI-4(20), NIST.800-53.r5 SI-7(8)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::Transfer::Connector`

AWS Config rule: transfer-connector-logging-enabled

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon CloudWatch logging is enabled for an AWS Transfer Family connector. The control fails if CloudWatch logging isn't enabled for the connector.

Amazon CloudWatch is a monitoring and observability service that provides visibility into your AWS resources, including AWS Transfer Family resources. For Transfer Family, CloudWatch provides consolidated auditing and logging for workflow progress and results. This includes several metrics that Transfer Family defines for workflows. You can configure Transfer Family to automatically log connector events in CloudWatch. To do this, you specify a logging role for the connector. For the logging role, you create an IAM role and a resource-based IAM policy that defines the permissions for the role.

Remediation

For information about enabling CloudWatch logging for a Transfer Family connector, see Amazon CloudWatch logging for AWS Transfer Family servers in the *AWS Transfer Family User Guide*.

[Transfer.4] Transfer Family agreements should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Transfer::Agreement`

AWS Config rule: transfer-agreement-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Transfer Family agreement has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the agreement doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the agreement doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

Note

Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an AWS Transfer Family agreement, see Resource tagging methods in the *Tagging AWS Resources and Tag Editor User Guide*.

[Transfer.5] Transfer Family certificates should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Transfer::Certificate`

AWS Config rule: transfer-certificate-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Transfer Family certificate has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the certificate doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the certificate doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an AWS Transfer Family certificate, see Resource tagging methods in the *Tagging AWS Resources and Tag Editor User Guide*.

[Transfer.6] Transfer Family connectors should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Transfer::Connector`

AWS Config rule: transfer-connector-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Transfer Family connector has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the connector doesn't have any tag keys, or it

doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the connector doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an AWS Transfer Family connector, see Resource tagging methods in the *Tagging AWS Resources and Tag Editor User Guide*.

[Transfer.7] Transfer Family profiles should be tagged

Category: Identify > Inventory > Tagging

Severity: Low

Resource type: `AWS::Transfer::Profile`

AWS Config rule: transfer-profile-tagged

Schedule type: Change triggered

Parameters:

Parameter	Description	Type	Allowed custom values	Security Hub default value
`requiredKeyTags`	A list of non-system tag keys that must be assigned to an evaluated resource. Tag keys are case sensitive.	StringList (maximum of 6 items)	1–6 tag keys that meet AWS requirements.	No default value

This control checks whether an AWS Transfer Family profile has the tag keys specified by the `requiredKeyTags` parameter. The control fails if the profile doesn't have any tag keys, or it doesn't have all the keys specified by the `requiredKeyTags` parameter. If you don't specify any values for the `requiredKeyTags` parameter, the control checks only for the existence of a tag key and fails if the profile doesn't have any tag keys. The control ignores system tags, which are applied automatically and have the `aws:` prefix. The control evaluates local profiles and partner profiles.

A tag is a label that you create and assign to an AWS resource. Each tag consists of a required tag key and an optional tag value. You can use tags to categorize resources by purpose, owner, environment, or other criteria. They can help you identify, organize, search for, and filter resources. They can also help you track resource owners for actions and notifications. You can also use tags to implement attribute-based access control (ABAC) as an authorization strategy. For more information about ABAC strategies, see Define permissions based on attributes with ABAC authorization in the *IAM User Guide*. For more information about tags, see the Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. Tags are accessible from many AWS services. They aren't intended to be used for private or sensitive data.

Remediation

For information about adding tags to an AWS Transfer Family profile, see Resource tagging methods in the *Tagging AWS Resources and Tag Editor User Guide*.

Security Hub controls for AWS WAF

These AWS Security Hub controls evaluate the AWS WAF service and resources. The controls might not be available in all AWS Regions. For more information, see Availability of controls by Region.

[WAF.1] AWS WAF Classic Global Web ACL logging should be enabled

Related requirements: NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::WAF::WebACL`

AWS Config rule: `waf-classic-logging-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether logging is enabled for an AWS WAF global web ACL. This control fails if logging is not enabled for the web ACL.

Logging is an important part of maintaining the reliability, availability, and performance of AWS WAF globally. It is a business and compliance requirement in many organizations, and allows you to troubleshoot application behavior. It also provides detailed information about the traffic that is analyzed by the web ACL that is attached to AWS WAF.

Remediation

To enable logging for an AWS WAF web ACL, see Logging web ACL traffic information in the *AWS WAF Developer Guide*.

[WAF.2] AWS WAF Classic Regional rules should have at least one condition

Related requirements: NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21)

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAFRegional::Rule`

AWS Config rule: `waf-regional-rule-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF Regional rule has at least one condition. The control fails if no conditions are present within a rule.

A WAF Regional rule can contain multiple conditions. The rule's conditions allow for traffic inspection and take a defined action (allow, block, or count). Without any conditions, the traffic passes without inspection. A WAF Regional rule with no conditions, but with a name or tag suggesting allow, block, or count, could lead to the wrong assumption that one of those actions is occurring.

Remediation

To add a condition to an empty rule, see Adding and removing conditions in a rule in the *AWS WAF Developer Guide*.

[WAF.3] AWS WAF Classic Regional rule groups should have at least one rule

Related requirements: NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21)

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAFRegional::RuleGroup`

AWS Config rule: `waf-regional-rulegroup-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF Regional rule group has at least one rule. The control fails if no rules are present within a rule group.

A WAF Regional rule group can contain multiple rules. The rule's conditions allow for traffic inspection and take a defined action (allow, block, or count). Without any rules, the traffic passes without inspection. A WAF Regional rule group with no rules, but with a name or tag suggesting allow, block, or count, could lead to the wrong assumption that one of those actions is occurring.

Remediation

To add rules and rule conditions to an empty rule group, see Adding and deleting rules from an AWS WAF Classic rule group and Adding and removing conditions in a rule in the *AWS WAF Developer Guide*.

[WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAFRegional::WebACL`

AWS Config rule: `waf-regional-webacl-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF Classic Regional web ACL contains any WAF rules or WAF rule groups. This control fails if a web ACL does not contain any WAF rules or rule groups.

A WAF Regional web ACL can contain a collection of rules and rule groups that inspect and control web requests. If a web ACL is empty, the web traffic can pass without being detected or acted upon by WAF depending on the default action.

Remediation

To add rules or rule groups to an empty AWS WAF Classic Regional web ACL, see Editing a Web ACL in the *AWS WAF Developer Guide*.

[WAF.6] AWS WAF Classic global rules should have at least one condition

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAF::Rule`

AWS Config rule: `waf-global-rule-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF global rule contains any conditions. The control fails if no conditions are present within a rule.

A WAF global rule can contain multiple conditions. A rule's conditions allow for traffic inspection and take a defined action (allow, block, or count). Without any conditions, the traffic passes without inspection. A WAF global rule with no conditions, but with a name or tag suggesting allow, block, or count, could lead to the wrong assumption that one of those actions is occurring.

Remediation

For instructions on creating a rule and adding conditions, see Creating a rule and adding conditions in the *AWS WAF Developer Guide*.

[WAF.7] AWS WAF Classic global rule groups should have at least one rule

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAF::RuleGroup`

AWS Config rule: `waf-global-rulegroup-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF global rule group has at least one rule. The control fails if no rules are present within a rule group.

A WAF global rule group can contain multiple rules. The rule's conditions allow for traffic inspection and take a defined action (allow, block, or count). Without any rules, the traffic passes without inspection. A WAF global rule group with no rules, but with a name or tag suggesting allow, block, or count, could lead to the wrong assumption that one of those actions is occurring.

Remediation

For instructions on adding a rule to a rule group, see Creating an AWS WAF Classic rule group in the *AWS WAF Developer Guide*.

[WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Related requirements: NIST.800-53.r5 AC-4(21), NIST.800-53.r5 SC-7, NIST.800-53.r5 SC-7(11), NIST.800-53.r5 SC-7(16), NIST.800-53.r5 SC-7(21)

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAF::WebACL`

AWS Config rule: `waf-global-webacl-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF global web ACL contains at least one WAF rule or WAF rule group. The control fails if a web ACL does not contain any WAF rules or rule groups.

A WAF global web ACL can contain a collection of rules and rule groups that inspect and control web requests. If a web ACL is empty, the web traffic can pass without being detected or acted upon by WAF depending on the default action.

Remediation

To add rules or rule groups to an empty AWS WAF global web ACL, see Editing a web ACL in the *AWS WAF Developer Guide*. For **Filter**, choose **Global (CloudFront)**.

[WAF.10] AWS WAF web ACLs should have at least one rule or rule group

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAFv2::WebACL`

AWS Config rule: `wafv2-webacl-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAFV2 web access control list (web ACL) contains at least one rule or rule group. The control fails if a web ACL does not contain any rules or rule groups.

A web ACL gives you fine-grained control over all of the HTTP(S) web requests that your protected resource responds to. A web ACL should contain a collection of rules and rule groups that inspect and control web requests. If a web ACL is empty, the web traffic can pass without being detected or acted upon by AWS WAF depending on the default action.

Remediation

To add rules or rule groups to an empty WAFV2 web ACL, see Editing a Web ACL in the *AWS WAF Developer Guide*.

[WAF.11] AWS WAF web ACL logging should be enabled

Related requirements: NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-7(8), PCI DSS v4.0.1/10.4.2

Category: Identify > Logging

Severity: Low

Resource type: `AWS::WAFv2::WebACL`

AWS Config rule: `wafv2-logging-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether logging is activated for an AWS WAFV2 web access control list (web ACL). This control fails if logging is deactivated for the web ACL.

> **Note**
>
> This control doesn't check whether AWS WAF web ACL logging is enabled for an account through Amazon Security Lake.

Logging maintains the reliability, availability, and performance of AWS WAF. In addition, logging is a business and compliance requirement in many organizations. By logging traffic that's analyzed by your web ACL, you can troubleshoot application behavior.

Remediation

To activate logging for an AWS WAF web ACL, see Managing logging for a web ACL in the *AWS WAF Developer Guide*.

[WAF.12] AWS WAF rules should have CloudWatch metrics enabled

Related requirements: NIST.800-53.r5 AC-4(26), NIST.800-53.r5 AU-10, NIST.800-53.r5 AU-12, NIST.800-53.r5 AU-2, NIST.800-53.r5 AU-3, NIST.800-53.r5 AU-6(3), NIST.800-53.r5 AU-6(4), NIST.800-53.r5 CA-7, NIST.800-53.r5 SC-7(10), NIST.800-53.r5 SC-7(9), NIST.800-53.r5 SI-7(8), NIST.800-171.r2 3.14.6, NIST.800-171.r2 3.14.7

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::WAFv2::RuleGroup`

AWS Config rule: `wafv2-rulegroup-logging-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF rule or rule group has Amazon CloudWatch metrics enabled. The control fails if the rule or rule group doesn't have CloudWatch metrics enabled.

Configuring CloudWatch metrics on AWS WAF rules and rule groups provides visibility into traffic flow. You can see which ACL rules are triggered and which requests are accepted and blocked. This visibility can help you identify malicious activity on your associated resources.

Remediation

To enable CloudWatch metrics on an AWS WAF rule group, invoke the UpdateRuleGroup API. To enable CloudWatch metrics on an AWS WAF rule, invoke the UpdateWebACL API. Set the `CloudWatchMetricsEnabled` field to `true`. When you use the AWS WAF console to create rules or rule groups, CloudWatch metrics are automatically enabled.

Security Hub controls for WorkSpaces

These AWS Security Hub controls evaluate the Amazon WorkSpaces service and resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::WorkSpaces::Workspace`

AWS Config rule: `workspaces-user-volume-encryption-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether a user volume in an Amazon WorkSpaces WorkSpace is encrypted at rest. The control fails if the WorkSpace user volume isn't encrypted at rest.

Data at rest refers to data that's stored in persistent, non-volatile storage for any duration. Encrypting data at rest helps you protect its confidentiality, which reduces the risk that an unauthorized user can access it.

Remediation

To encrypt a WorkSpaces user volume, see Encrypt a WorkSpace in the *Amazon WorkSpaces Administration Guide*.

[WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Category: Protect > Data Protection > Encryption of data-at-rest

Severity: Medium

Resource type: `AWS::WorkSpaces::Workspace`

AWS Config rule: `workspaces-root-volume-encryption-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether a root volume in an Amazon WorkSpaces WorkSpace is encrypted at rest. The control fails if the WorkSpace root volume isn't encrypted at rest.

Data at rest refers to data that's stored in persistent, non-volatile storage for any duration. Encrypting data at rest helps you protect its confidentiality, which reduces the risk that an unauthorized user can access it.

Remediation

To encrypt a WorkSpaces root volume, see Encrypt a WorkSpace in the *Amazon WorkSpaces Administration Guide*.

Required permissions to configure controls in Security Hub CSPM

To view information about security controls and enable and disable security controls in standards, the AWS Identity and Access Management (IAM) role that you use to access AWS Security Hub CSPM needs permissions to call the following operations of the Security Hub CSPM API.

To get the necessary permissions, you can use Security Hub CSPM managed policies. Alternatively, you can update custom IAM policies to include permissions for these actions.

- **BatchGetSecurityControls** – Returns information about a batch of security controls for the current account and AWS Region.

- **ListSecurityControlDefinitions** – Returns information about security controls that apply to a specified standard.
- **ListStandardsControlAssociations** – Identifies whether a security control is currently enabled in or disabled from each enabled standard in the account.
- **BatchGetStandardsControlAssociations** – For a batch of security controls, identifies whether each control is currently enabled in or disabled from a specified standard.
- **BatchUpdateStandardsControlAssociations** – Used to enable a security control in standards that include the control, or to disable a control in standards. This is a batch substitute for the existing `UpdateStandardsControl` operation.
- **BatchUpdateStandardsControlAssociations** – Used to enable or disable a batch of security controls in standards that include the controls. This is a batch substitute for the existing `UpdateStandardsControl` operation.
- **UpdateStandardsControl** – Used to enable or disable a single security control in standards that include the control
- **DescribeStandardsControl** – Returns details about specified security controls.

In addition to the preceding APIs, you should add permission to call `BatchGetControlEvaluations` to your IAM role. This permission is necessary to view the enablement and compliance status of a control, the findings count for a control, and the overall security score for controls on the Security Hub CSPM console. Because only the console calls `BatchGetControlEvaluations`, this permission doesn't directly correspond to publicly documented Security Hub CSPM APIs or AWS CLI commands.

Enabling controls in Security Hub CSPM

In AWS Security Hub CSPM, a control is a safeguard within a security standard that helps an organization protect the confidentiality, integrity, and availability of its information. Each Security Hub CSPM control is related to a specific AWS resource. When you enable a control, Security Hub CSPM begins to run security checks for the control and generates findings for it. Security Hub CSPM also considers all enabled controls when calculating security scores.

You can choose to enable a control across all of the security standards that it applies to. Alternatively, you can configure the enablement status differently in different standards. We recommend the former option, in which the enablement status of a control is aligned across all of your enabled standards. For instructions on enabling a control across all standards that it applies it,

see Enabling a control across standards. For instructions on enabling a control in specific standards, see Enabling a control in a specific standard.

If you enable cross-Region aggregation and sign in to an aggregation Region, the Security Hub CSPM console shows controls that are available in at least one linked Region. If a control is available in a linked Region but not in the aggregation Region, you can't enable or disable that control from the aggregation Region.

You can enable and disable controls in each Region by using the Security Hub CSPM console, Security Hub CSPM API, or AWS CLI.

The instructions for enabling and disabling controls vary based on whether or not you use central configuration. This topic describes the differences. Central configuration is available to users who integrate Security Hub CSPM and AWS Organizations. We recommend using central configuration to simplify the process of enabling and disabling controls in multi-account, multi-Region environments. If you use central configuration, you can enable a control across multiple accounts and Regions through the use of configuration policies. If you don't use central configuration, you must enable a control separately in each Region and account.

Enabling a control across standards

We recommend enabling a AWS Security Hub CSPM control across all of the standards that the control applies to. If you turn on consolidated control findings, you receive one finding per control check even if a control belongs to more than one standard.

Cross-standard enablement in multi-account, multi-Region environments

To enable a security control across multiple AWS accounts and AWS Regions, you must be signed in to the delegated Security Hub CSPM administrator account and use central configuration.

Under central configuration, the delegated administrator can create Security Hub CSPM configuration policies that enable specified controls across enabled standards. You can then associate the configuration policy with specific accounts and organizational units (OUs) or the root. A configuration policy takes effect in your home Region (also called an aggregation Region) and all linked Regions.

Configuration policies offer customization. For example, you can choose to enable all controls in one OU, and you can choose to enable only Amazon Elastic Compute Cloud (EC2) controls in another OU. The level of granularity depends on your intended goals for security coverage in your

organization. For instructions on creating a configuration policy that enables specified controls across standards, see Creating and associating configuration policies.

> **Note**
>
> The delegated administrator can create configuration policies to manage controls in all standards except the Service-Managed Standard: AWS Control Tower. Controls for this standard should be configured in the AWS Control Tower service.

If you want some accounts to configure their own controls rather than the delegated administrator, the delegated administrator can designate those accounts as self-managed. Self-managed accounts must configure controls separately in each Region.

Cross-standard enablement in single account and Region

If you don't use central configuration or are a self-managed account, you can't use configuration policies to centrally enable controls in multiple accounts and Regions. However, you can use the following steps to enable a control in a single account and Region.

Security Hub CSPM console

To enable a control across standards in one account and Region

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Choose **Controls** from the navigation pane.
3. Choose the **Disabled** tab.
4. Choose the option next to a control.
5. Choose **Enable Control** (this option doesn't appear for a control that's already enabled).
6. Repeat in each Region in which you want to enable the control.

Security Hub CSPM API

To enable a control across standards in one account and Region

1. Invoke the ListStandardsControlAssociations API. Provide a security control ID.

   Example request:

```
{
    "SecurityControlId": "IAM.1"
}
```

2. Invoke the BatchUpdateStandardsControlAssociations API. Provide the Amazon Resource Name (ARN) of any standards that the control isn't enabled in. To obtain standard ARNs, run DescribeStandards.
3. Set the AssociationStatus parameter equal to ENABLED. If you follow these steps for a control that's already enabled, the API returns an HTTP status code 200 response.

   Example request:

```
{
    "StandardsControlAssociationUpdates": [{"SecurityControlId": "IAM.1",
 "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0", "AssociationStatus": "ENABLED"}, {"SecurityControlId": "IAM.1",
 "StandardsArn": "arn:aws:securityhub:::standards/aws-foundational-security-
best-practices/v/1.0.0", "AssociationStatus": "ENABLED"}]
}
```

4. Repeat in each Region in which you want to enable the control.

AWS CLI

To enable a control across standards in one account and Region

1. Run the list-standards-control-associations command. Provide a security control ID.

```
aws securityhub  --region us-east-1 list-standards-control-associations --
security-control-id CloudTrail.1
```

2. Run the batch-update-standards-control-associations command. Provide the Amazon Resource Name (ARN) of any standards that the control isn't enabled in. To obtain standard ARNs, run the describe-standards command.
3. Set the AssociationStatus parameter equal to ENABLED. If you follow these steps for a control that's already enabled, the command returns an HTTP status code 200 response.

```
aws securityhub  --region us-east-1 batch-update-standards-control-associations
 --standards-control-association-updates '[{"SecurityControlId": "CloudTrail.1",
```

```
 "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0", "AssociationStatus": "ENABLED"}, {"SecurityControlId": "CloudTrail.1",
 "StandardsArn": "arn:aws:securityhub:::standards/cis-aws-foundations-benchmark/
v/1.4.0", "AssociationStatus": "ENABLED"}]'
```

4. Repeat in each Region in which you want to enable the control.

Enabling a control in a specific standard

When you enable a standard in AWS Security Hub CSPM, all of the controls that apply to it are automatically enabled in that standard (the exception to this is service-managed standards). You can then disable and re-enable specific controls in the standard. However, we recommend aligning the enablement status of a control across all of your enabled standards. For instructions on enabling a control across all standards, see Enabling a control across standards.

The details page for a standard contains the list of applicable controls for the standard, and information about which controls are currently enabled in and disabled in that standard.

On the standards details page, you can also enable controls in specific standards. You must enable controls in specific standards separately in each AWS account and AWS Region. When you enable a control in specific standards, it only impacts the current account and Region.

To enable a control in a standard, you must first enable at least one standard to which the control applies. For instructions on enabling a standard, see Enabling a security standard. When you enable a control in one or more standards, Security Hub CSPM starts to generate findings for that control. Security Hub CSPM includes the control status in the calculation of the overall security score and standard security scores. Even if you enable a control in multiple standards, you'll receive a single finding per security check across standards if you turn on consolidated control findings. For more information, see Consolidated control findings.

To enable a control in a standard, the control must be available in your current Region. For more information, see Availability of controls by Region.

Follow these steps to enable a Security Hub CSPM control in a *specific* standard. In lieu of the following steps, you can also use the `UpdateStandardsControl` API action to enable controls in a specific standard. For instructions on enabling a control in *all* standards, see Cross-standard enablement in single account and Region.

Security Hub CSPM console

To enable a control in a specific standard

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Choose **Security standards** from the navigation pane.
3. Choose **View results** for the relevant standard.
4. Select a control.
5. Choose **Enable Control** (this option doesn't appear for a control that's already enabled). Confirm by choosing **Enable**.

Security Hub CSPM API

To enable a control in a specific standard

1. Run `ListSecurityControlDefinitions`, and provide a standard ARN to get a list of available controls for a specific standard. To obtain a standard ARN, run `DescribeStandards`. This API returns standard-agnostic security control IDs, not standard-specific control IDs.

   Example request:

   ```
   {
       "StandardsArn": "arn:aws:securityhub:::standards/aws-foundational-security-best-practices/v/1.0.0"
   }
   ```

2. Run `ListStandardsControlAssociations`, and provide a specific control ID to return the current enablement status of a control in each standard.

   Example request:

   ```
   {
       "SecurityControlId": "IAM.1"
   }
   ```

3. Run `BatchUpdateStandardsControlAssociations`. Provide the ARN of the standard that you want to enable the control in.

4. Set the `AssociationStatus` parameter equal to `ENABLED`.

   Example request:

```
{
    "StandardsControlAssociationUpdates": [{"SecurityControlId": "IAM.1",
 "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0", "AssociationStatus": "ENABLED"}]
}
```

AWS CLI

To enable a control in a specific standard

1. Run the `list-security-control-definitions` command, and provide a standard ARN to get a list of available controls for a specific standard. To obtain a standard ARN, run `describe-standards`. This command returns standard-agnostic security control IDs, not standard-specific control IDs.

```
aws securityhub --region us-east-1 list-security-control-definitions --
standards-arn "arn:aws:securityhub:us-east-1::standards/aws-foundational-
security-best-practices/v/1.0.0"
```

2. Run the `list-standards-control-associations` command, and provide a specific control ID to return the current enablement status of a control in each standard.

```
aws securityhub  --region us-east-1 list-standards-control-associations --
security-control-id CloudTrail.1
```

3. Run the `batch-update-standards-control-associations` command. Provide the ARN of the standard that you want to enable the control in.
4. Set the `AssociationStatus` parameter equal to `ENABLED`.

```
aws securityhub  --region us-east-1 batch-update-standards-control-associations
 --standards-control-association-updates '[{"SecurityControlId": "CloudTrail.1",
 "StandardsArn": "arn:aws:securityhub:us-east-1::standards/aws-foundational-
security-best-practices/v/1.0.0", "AssociationStatus": "ENABLED"}]'
```

Enabling new controls in enabled standards automatically

AWS Security Hub CSPM regularly releases new controls and adds them to one or more standards. You can choose whether to automatically enable new controls in your enabled standards.

We recommend using Security Hub CSPM central configuration to automatically enable new security controls. You can create configuration policies that include a list of controls to be disabled across standards. All other controls, including newly released ones, are enabled by default. Alternatively, you can create policies that include a list of controls to be enabled across standards. All other controls, including newly released ones, are disabled by default. For more information, see Understanding central configuration in Security Hub CSPM.

Security Hub CSPM doesn't enable new controls when they are added to a standard that you haven't enabled.

The following instructions apply only if you don't use central configuration.

Choose your preferred access method, and follow the steps to automatically enable new controls in enabled standards.

> **Note**
>
> When you automatically enable new controls using the following instructions, you can interact with the controls in the console and programmatically immediately after release. However, automatically enabled controls have a temporary default status of **Disabled**. It can take up to several days for Security Hub CSPM to process the control release and designate the control as **Enabled** in your account. During the processing period, you can manually enable or disable a control, and Security Hub CSPM will maintain that designation regardless of whether you have automatic control enablement turned on.

Security Hub CSPM console

To automatically enable new controls

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Settings**, and then choose the **General** tab.
3. Under **Controls**, choose **Edit**.

4. Turn on **Auto-enable new controls in enabled standards**.
5. Choose **Save**.

Security Hub CSPM API

To automatically enable new controls

1. Run `UpdateSecurityHubConfiguration`.
2. To automatically enable new controls for enabled standards, set `AutoEnableControls` to `true`. If you don't want to automatically enable new controls, set `AutoEnableControls` to false.

AWS CLI

To automatically enable new controls

1. Run the `update-security-hub-configuration` command.
2. To automatically enable new controls for enabled standards, specify `--auto-enable-controls`. If you don't want to automatically enable new controls, specify `--no-auto-enable-controls`.

```
aws securityhub update-security-hub-configuration --auto-enable-controls | --no-auto-enable-controls
```

Example command

```
aws securityhub update-security-hub-configuration --auto-enable-controls
```

If you don't automatically enable new controls, then you must enable them manually. For instructions, see Enabling controls in Security Hub CSPM.

Disabling controls in Security Hub CSPM

To reduce finding noise, it can be helpful to disable controls that aren't relevant to your environment. In AWS Security Hub CSPM, you can disable a control across all security standards or for only specific standards.

If you disable a control across all standards, the following occurs:

- Security checks for the control are no longer performed.
- No additional findings are generated for the control.
- Existing findings are no longer updated for the control.
- Existing findings for the control are archived automatically, typically within 3–5 days on a best-effort basis.
- Security Hub CSPM removes any related AWS Config rules that it created for the control.

If you disable a control for only specific standards, Security Hub CSPM stops running security checks for the control for only those standards. This also removes the control from calculations of the security score for each of those standards. If the control is enabled in other standards, Security Hub CSPM retains the associated AWS Config rule, if applicable, and continues running security checks for the control for the other standards. Security Hub CSPM also includes the control when it calculates the security score for each of the other standards, which affects your summary security score.

If you disable a standard, all of the controls that apply to the standard are disabled automatically for that standard. However, the controls might continue to be enabled in other standards. When you disable a standard, Security Hub CSPM doesn't track which controls were disabled for the standard. Consequently, if you later re-enable the same standard, all the controls that apply to it are automatically enabled. For information about disabling a standard, see Disabling a standard.

Disabling a control isn't a permanent action. Suppose you disable a control, and then enable a standard that includes the control. The control is then enabled for that standard. When you enable a standard in Security Hub CSPM, all the controls that apply to the standard are automatically enabled. For information about enabling a standard, see Enabling a standard.

Topics

Disabling a control across standards

We recommend disabling an AWS Security Hub CSPM control across standards to maintain alignment throughout your organization. If you disable a control in only specific standards, you continue to receive findings for the control if it is enabled in other standards.

Cross-standard disablement in multiple accounts and Regions

To disable a security control across multiple AWS accounts and AWS Regions, you must use central configuration.

When you use central configuration, the delegated administrator can create Security Hub CSPM configuration policies that disable specified controls across enabled standards. You can then associate the configuration policy with specific accounts, OUs, or the root. A configuration policy takes effect in your home Region (also called an aggregation Region) and all linked Regions.

Configuration policies offer customization. For example, you can choose to disable all AWS CloudTrail controls in one OU, and you can choose to disable all IAM controls in another OU. The level of granularity depends on your intended goals for security coverage in your organization. For instructions on creating a configuration policy that disables specified controls across standards, see Creating and associating configuration policies.

> **Note**
>
> The delegated administrator can create configuration policies to manage controls in all standards except the Service-Managed Standard: AWS Control Tower. Controls for this standard should be configured in the AWS Control Tower service.

If you want some accounts to configure their own controls rather than the delegated administrator, the delegated administrator can designate those accounts as self-managed. Self-managed accounts must configure controls separately in each Region.

Cross-standard disablement in a single account and Region

If you don't use central configuration or are a self-managed account, you can't use configuration policies to centrally disable controls in multiple accounts and Regions. However, you can disable a control in a single account and Region.

Security Hub CSPM console

To disable a control across standards in one account and Region

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Choose **Controls** from the navigation pane.
3. Choose the option next to a control.
4. Choose **Disable Control**. This option doesn't appear for a control that's already disabled.
5. Select a reason for disabling the control, and confirm by choosing **Disable**.
6. Repeat in each Region in which you want to disable the control.

Security Hub CSPM API

To disable a control across standards in one account and Region

1. Invoke the ListStandardsControlAssociations API. Provide a security control ID.

   Example request:

   ```
   {
       "SecurityControlId": "IAM.1"
   }
   ```

2. Invoke the BatchUpdateStandardsControlAssociations API. Provide the ARN of any standards that the control is enabled in. To obtain standard ARNs, run `DescribeStandards`.
3. Set the `AssociationStatus` parameter equal to `DISABLED`. If you follow these steps for a control that's already disabled, the API returns an HTTP status code 200 response.

   Example request:

   ```
   {
       "StandardsControlAssociationUpdates": [{"SecurityControlId": "IAM.1",
    "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
   benchmark/v/1.2.0", "AssociationStatus": "DISABLED", "UpdatedReason": "Not
    applicable to environment"}, {"SecurityControlId": "IAM.1", "StandardsArn":
    "arn:aws:securityhub:::standards/aws-foundational-security-best-practices/
   ```

```
v/1.0.0", "AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable to
 environment"}}]
}
```

4. Repeat in each Region in which you want to disable the control.

AWS CLI

To disable a control across standards in one account and Region

1. Run the list-standards-control-associations command. Provide a security control ID.

```
aws securityhub  --region us-east-1 list-standards-control-associations --
security-control-id CloudTrail.1
```

2. Run the batch-update-standards-control-associations command. Provide the ARN of any standards that the control is enabled in. To obtain standard ARNs, run the `describe-standards` command.
3. Set the `AssociationStatus` parameter equal to `DISABLED`. If you follow these steps for a control that's already disabled, the command returns an HTTP status code 200 response.

```
aws securityhub  --region us-east-1 batch-update-standards-control-associations
 --standards-control-association-updates '[{"SecurityControlId": "CloudTrail.1",
 "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0", "AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable
 to environment"}, {"SecurityControlId": "CloudTrail.1", "StandardsArn":
 "arn:aws:securityhub:::standards/cis-aws-foundations-benchmark/v/1.4.0",
 "AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable to
 environment"}]'
```

4. Repeat in each Region in which you want to disable the control.

Disabling a control in a specific standard

You can disable a control in only specific security standards, instead of across all standards. If the control applies to other enabled standards, AWS Security Hub CSPM continues to run security checks for the control and you continue to receive findings for the control.

We recommend aligning the enablement status of a control across all of the enabled standards that the control applies to. For information about disabling a control across all of the standards that it applies to, see Disabling a control across standards.

On the standards details page, you can also disable controls in specific standards. You must disable controls in specific standards separately in each AWS account and AWS Region. When you disable a control in specific standards, it affects only the current account and Region.

Choose your preferred method, and follow these steps to disable a control in one or more specific standards.

Security Hub CSPM console

To disable a control in a specific standard

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Choose **Security standards** from the navigation pane. Choose **View results** for the relevant standard.
3. Select a control.
4. Choose **Disable Control**. This option doesn't appear for a control that's already disabled.
5. Provide a reason for disabling the control, and confirm by choosing **Disable**.

Security Hub CSPM API

To disable a control in a specific standard

1. Run `ListSecurityControlDefinitions`, and provide a standard ARN to get a list of available controls for a specific standard. To obtain a standard ARN, run `DescribeStandards`. This API returns standard-agnostic security control IDs, not standard-specific control IDs.

   Example request:

```
{
    "StandardsArn": "arn:aws:securityhub:::standards/aws-foundational-security-best-practices/v/1.0.0"
}
```

2. Run ListStandardsControlAssociations, and provide a specific control ID to return the current enablement status of a control in each standard.

   Example request:

   ```
   {
       "SecurityControlId": "IAM.1"
   }
   ```

3. Run BatchUpdateStandardsControlAssociations. Provide the ARN of the standard in which you want to disable the control.

4. Set the `AssociationStatus` parameter equal to `DISABLED`. If you follow these steps for a control that's already disabled, the API returns an HTTP status code 200 response.

   Example request:

   ```
   {
       "StandardsControlAssociationUpdates": [{"SecurityControlId": "IAM.1",
    "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
   v/1.2.0", "AssociationStatus": "DISABLED",  "UpdatedReason": "Not applicable to
    environment"}]
   }
   ```

AWS CLI

To disable a control in a specific standard

1. Run the list-security-control-definitions command, and provide a standard ARN to get a list of available controls for a specific standard. To obtain a standard ARN, run `describe-standards`. This command returns standard-agnostic security control IDs, not standard-specific control IDs.

   ```
   aws securityhub --region us-east-1 list-security-control-definitions --
   standards-arn "arn:aws:securityhub:us-east-1::standards/aws-foundational-
   security-best-practices/v/1.0.0"
   ```

2. Run the list-standards-control-associations command, and provide a specific control ID to return the current enablement status of a control in each standard.

```
aws securityhub  --region us-east-1 list-standards-control-associations --
security-control-id CloudTrail.1
```

3. Run the batch-update-standards-control-associations command. Provide the ARN of the standard in which you want to disable the control.
4. Set the `AssociationStatus` parameter equal to `DISABLED`. If you follow these steps for a control that's already enabled, the command returns an HTTP status code 200 response.

```
aws securityhub  --region us-east-1 batch-update-standards-control-
associations --standards-control-association-updates '[{"SecurityControlId":
 "CloudTrail.1", "StandardsArn": "arn:aws:securityhub:us-east-1::standards/aws-
foundational-security-best-practices/v/1.0.0", "AssociationStatus": "DISABLED",
 "UpdatedReason": "Not applicable to environment"}]'
```

Suggested controls to disable in Security Hub CSPM

We recommend disabling some AWS Security Hub CSPM controls to reduce finding noise and usage costs.

Controls that use global resources

Some AWS services support global resources, which means that you can access the resource from any AWS Region. To save on the cost of AWS Config, you can disable recording of global resources in all but one Region. After you do this, however, Security Hub CSPM stills run security checks in all Regions where a control is enabled and charges you based on the number of checks per account per Region. Accordingly, to reduce finding noise and save on the cost of Security Hub CSPM, you should also disable controls that involve global resources in all Regions except the Region that records global resources.

If a control involves global resources but is available in only one Region, disabling it in that Region prevents you from getting any findings for the underlying resource. In this case, we recommend keeping the control enabled. When using cross-Region aggregation, the Region in which the control is available should be the aggregation Region or one of the linked Regions. The following controls involve global resources but are available in only a single Region:

- **All CloudFront controls** – Available only in the US East (N. Virginia) Region
- **GlobalAccelerator.1** – Available only in the US West (Oregon) Region

- **Route53.2** – Available only in the US East (N. Virginia) Region
- **WAF.1, WAF.6, WAF.7, WAF.8** – Available only in the US East (N. Virginia) Region

> **Note**
>
> If you use central configuration, Security Hub CSPM automatically disables controls that involve global resources in all Regions except the home Region. Other controls that you choose to enable though a configuration policy are enabled in all Regions where they are available. To limit findings for these controls to just one Region, you can update your AWS Config recorder settings and turn off global resource recording in all Regions except the home Region.
>
> If an enabled control that involves global resources isn't supported in the home Region, Security Hub CSPM tries to enable the control in one linked Region where the control is supported. With central configuration, you lack coverage for a control that isn't available in the home Region or any of the linked Regions.
>
> For more information about central configuration, see Understanding central configuration in Security Hub CSPM.

For controls that have a *periodic* schedule type, disabling them in Security Hub CSPM is required to prevent billing. Setting the AWS Config parameter `includeGlobalResourceTypes` to `false` doesn't affect periodic Security Hub CSPM controls.

The following Security Hub CSPM controls use global resources:

- [Account.1] Security contact information should be provided for an AWS account
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins

- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudFront.16] CloudFront distributions should use origin access control for Lambda function URL origins
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.9] MFA should be enabled for the root user
- [IAM.10] Password policies for IAM users should have strong configurations
- [IAM.11] Ensure IAM password policy requires at least one uppercase letter
- [IAM.12] Ensure IAM password policy requires at least one lowercase letter
- [IAM.13] Ensure IAM password policy requires at least one symbol
- [IAM.14] Ensure IAM password policy requires at least one number
- [IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
- [IAM.16] Ensure IAM password policy prevents password reuse
- [IAM.17] Ensure IAM password policy expires passwords within 90 days or less
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed

- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

CloudTrail logging controls

The CloudTrail.2 control evaluates the use of AWS Key Management Service (AWS KMS) to encrypt AWS CloudTrail trail logs. If you log these trails in a centralized logging account, you need to enable this control only in the account and AWS Region where centralized logging takes place.

If you use central configuration, the enablement status of a control is aligned across the home Region and linked Regions. You can't disable a control in some Regions and enable it in others. In this case, you can suppress findings from the CloudTrail.2 control to reduce finding noise.

CloudWatch alarm controls

If you prefer to use Amazon GuardDuty for anomaly detection instead of Amazon CloudWatch alarms, you can disable the following controls, which focus on CloudWatch alarms:

- [CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user
- [CloudWatch.2] Ensure a log metric filter and alarm exist for unauthorized API calls
- [CloudWatch.3] Ensure a log metric filter and alarm exist for Management Console sign-in without MFA
- [CloudWatch.4] Ensure a log metric filter and alarm exist for IAM policy changes
- [CloudWatch.5] Ensure a log metric filter and alarm exist for CloudTrail configuration changes

- [CloudWatch.6] Ensure a log metric filter and alarm exist for AWS Management Console authentication failures
- [CloudWatch.7] Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys
- [CloudWatch.8] Ensure a log metric filter and alarm exist for S3 bucket policy changes
- [CloudWatch.9] Ensure a log metric filter and alarm exist for AWS Config configuration changes
- [CloudWatch.10] Ensure a log metric filter and alarm exist for security group changes
- [CloudWatch.11] Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)
- [CloudWatch.12] Ensure a log metric filter and alarm exist for changes to network gateways
- [CloudWatch.13] Ensure a log metric filter and alarm exist for route table changes
- [CloudWatch.14] Ensure a log metric filter and alarm exist for VPC changes

Understanding security checks and scores in Security Hub CSPM

For each control that you enable, AWS Security Hub CSPM runs security checks. A security check produces a finding that tells you whether a specific AWS resource is in compliance with the rules that the control includes.

Some checks run on a periodic schedule. Other checks only run when there is a change to the resource state. For more information, see Schedule for running security checks.

Many security checks use AWS Config managed or custom rules to establish the compliance requirements. To run these checks, you must set up AWS Config and turn on resource recording for required resources. For more information on setting up AWS Config, see Enabling and configuring AWS Config for Security Hub CSPM. For a list of AWS Config resources that you must record for each standard, see Required AWS Config resources for control findings. Other controls use custom Lambda functions, which are managed by Security Hub CSPM and don't require any prerequisites.

As Security Hub CSPM runs security checks, it generates findings and assigns them a compliance status. For more information about compliance status, see Evaluating the compliance status of Security Hub CSPM findings.

Security Hub CSPM uses the compliance status of control findings to determine an overall control status. Based on the control status, Security Hub CSPM also calculates a security score across all enabled controls and for specific standards. For more information, see the section called “Compliance status and control status” and the section called “Calculating security scores”.

If you've turned on consolidated control findings, Security Hub CSPM generates a single finding even when a control is associated with more than one standard. For more information, see Consolidated control findings.

Topics

Required AWS Config resources for control findings

In AWS Security Hub CSPM, some controls use service-linked AWS Config rules that detect configuration changes in your AWS resources. For Security Hub CSPM to generate accurate findings for these controls, you must enable AWS Config and turn on resource recording in AWS Config. For information about how Security Hub CSPM uses AWS Config rules and how to enable and configure AWS Config, see Enabling and configuring AWS Config for Security Hub CSPM. For detailed information about resource recording, see Working with the configuration recorder in the *AWS Config Developer Guide*.

To receive accurate control findings, you must turn on AWS Config resource recording for enabled controls with a *change triggered* schedule type. Some controls with a *periodic* schedule type also require resource recording. This page lists the required resources for these Security Hub CSPM controls.

Security Hub CSPM controls can rely on managed AWS Config rules or custom Security Hub CSPM rules. Make sure there aren't any AWS Identity and Access Management (IAM) policies or AWS Organizations managed policies that prevent AWS Config from having permission to record your resources. Security Hub CSPM controls evaluate resource configurations directly and don't take AWS Organizations policies into account.

> **Note**
>
> In AWS Regions where a control isn't available, the corresponding resource isn't available in AWS Config. For a list of these limits, see Regional limits on Security Hub CSPM controls.

Topics

Required resources for all Security Hub CSPM controls

For Security Hub CSPM to generate findings for change triggered controls that are enabled and use an AWS Config rule, you must record the following types of resources in AWS Config. This table also indicates which controls evaluate a particular type of resource. A single control might evaluate more than one type of resource.

AWS service	Resource types	Related controls
AWS Amplify	`AWS::Amplify::App`	Amplify.1
	`AWS::Amplify::Branch`	Amplify.2
Amazon API Gateway	`AWS::ApiGateway::Stage`	APIGateway.1 APIGateway.2 APIGateway.3 APIGateway.4 APIGateway.5

AWS service	Resource types	Related controls
	AWS::ApiGatewayV2::Stage	APIGateway.1 APIGateway.9
AWS AppConfig	AWS::AppConfig::Application	AppConfig.1
	AWS::AppConfig::ConfigurationProfile	AppConfig.2
	AWS::AppConfig::Environment	AppConfig.3
	AWS::AppConfig::ExtensionAssociation	AppConfig.4
Amazon AppFlow	AWS::AppFlow::Flow	AppFlow.1
AWS App Runner	AWS::AppRunner::Service	AppRunner.1
	AWS::AppRunner::VpcConnector	AppRunner.2
AWS AppSync	AWS::AppSync::GraphQLApi	AppSync.2 AppSync.4 AppSync.5

AWS service	Resource types	Related controls
	`AWS::AppSync::ApiCache`	AppSync.1 AppSync.6
AWS Backup	`AWS::Backup::BackupPlan`	Backup.5
	`AWS::Backup::BackupVault`	Backup.3
	`AWS::Backup::RecoveryPoint`	Backup.1 Backup.2
	`AWS::Backup::ReportPlan`	Backup.4
AWS Batch	`AWS::Batch::ComputeEnvironment`	Batch.3 Batch.4
	`AWS::Batch::JobQueue`	Batch.1
	`AWS::Batch::SchedulingPolicy`	Batch.2
AWS Certificate Manager (ACM)	`AWS::ACM::Certificate`	ACM.1 ACM.2 ACM.3
Amazon Athena	`AWS::Athena::DataCatalog`	Athena.2

AWS service	Resource types	Related controls
	`AWS::Athena::WorkGroup`	Athena.3 Athena.4
AWS CloudFormation	`AWS::CloudFormation::Stack`	CloudFormation.2
Amazon CloudFront	`AWS::CloudFront::Distribution`	CloudFront.1 CloudFront.3 CloudFront.4 CloudFront.5 CloudFront.6 CloudFront.7 CloudFront.8 CloudFront.9 CloudFront.10 CloudFront.13 CloudFront.14 CloudFront.15 CloudFront.16
AWS CloudTrail	`AWS::CloudTrail::Trail`	CloudTrail.9
Amazon CloudWatch	`AWS::CloudWatch::Alarm`	CloudWatch.15 CloudWatch.17

AWS service	Resource types	Related controls
AWS CodeArtifact	`AWS::CodeArtifact::Repository`	CodeArtifact.1
AWS CodeBuild	`AWS::CodeBuild::Project`	CodeBuild.1 CodeBuild.2 CodeBuild.3 CodeBuild.4
	`AWS::CodeBuild::ReportGroup`	CodeBuild.7
Amazon CodeGuru Profiler	`AWS::CodeGuruProfiler::ProfilingGroup`	CodeGuruProfiler.1
Amazon CodeGuru Reviewer	`AWS::CodeGuruReviewer::RepositoryAssociation`	CodeGuruReviewer.1
Amazon Cognito	`AWS::Cognito::IdentityPool`	Cognito.2
	`AWS::Cognito::UserPool`	Cognito.1

AWS service	Resource types	Related controls
Amazon Connect	`AWS::CustomerProfiles::ObjectType`	Connect.1
	`AWS::Connect::Instance`	Connect.2
AWS DataSync	`AWS::DataSync::Task`	DataSync.1 DataSync.2
Amazon Detective	`AWS::Detective::Graph`	Detective.1
AWS Database Migration Service (AWS DMS)	`AWS::DMS::Certificate`	DMS.2
	`AWS::DMS::Endpoint`	DMS.9 DMS.10 DMS.11 DMS.12
	`AWS::DMS::EventSubscription`	DMS.3
	`AWS::DMS::ReplicationInstance`	DMS.4 DMS.6
	`AWS::DMS::ReplicationSubnetGroup`	DMS.5

AWS service	Resource types	Related controls
	AWS::DMS::ReplicationTask	DMS.7 DMS.8
Amazon DynamoDB	AWS::DynamoDB::Table	DynamoDB.1 DynamoDB.2 DynamoDB.5 DynamoDB.6
Amazon Elastic Compute Cloud (EC2)	AWS::EC2::ClientVpnEndpoint	EC2.51
	AWS::EC2::CustomerGateway	EC2.36
	AWS::EC2::DHCPOptions	EC2.174
	AWS::EC2::EIP	EC2.12 EC2.37
	AWS::EC2::FlowLog	EC2.48

AWS service	Resource types	Related controls
	AWS::EC2::Instance	EC2.4 EC2.8 EC2.9 EC2.17 EC2.24 EC2.38 EMR.1 SSM.1
	AWS::EC2::InternetGateway	EC2.39
	AWS::EC2::LaunchTemplate	EC2.25 EC2.170 EC2.175
	AWS::EC2::NatGateway	EC2.40
	AWS::EC2::NetworkAcl	EC2.16 EC2.21 EC2.41
	AWS::EC2::NetworkInterface	EC2.22 EC2.35 EC2.180

AWS service	Resource types	Related controls
	AWS::EC2::PrefixList	EC2.176
	AWS::EC2::RouteTable	EC2.42
	AWS::EC2::SecurityGroup	EC2.2 EC2.13 EC2.14 EC2.18 EC2.19 EC2.43
	AWS::EC2::SpotFleet	EC2.173
	AWS::EC2::Subnet	EC2.15 EC2.44 ElastiCache.7
	AWS::EC2::TrafficMirrorFilter	EC2.178
	AWS::EC2::TrafficMirrorSession	EC2.177
	AWS::EC2::TrafficMirrorTarget	EC2.179

AWS service	Resource types	Related controls
	AWS::EC2::TransitGateway	EC2.23 EC2.52
	AWS::EC2::TransitGatewayAttachment	EC2.33
	AWS::EC2::TransitGatewayRouteTable	EC2.34
	AWS::EC2::Volume	EC2.3 EC2.45
	AWS::EC2::VPC	EC2.6 EC2.46
	AWS::EC2::VPCBlockPublicAccessOptions	EC2.172
	AWS::EC2::VPCEndpointService	EC2.47
	AWS::EC2::VPCPeeringConnection	EC2.49

AWS service	Resource types	Related controls
	AWS::EC2::VPNConnection	EC2.20 EC2.171
	AWS::EC2::VPNGateway	EC2.50
Amazon EC2 Auto Scaling	AWS::AutoScaling::AutoScalingGroup	AutoScaling.1 AutoScaling.2 AutoScaling.6 AutoScaling.9 AutoScaling.10
	AWS::AutoScaling::LaunchConfiguration	AutoScaling.3 Autoscaling.5
Amazon EC2 Systems Manager (SSM)	AWS::SSM::AssociationCompliance	SSM.3
	AWS::SSM::ManagedInstanceInventory	SSM.1
	AWS::SSM::PatchCompliance	SSM.2
Amazon Elastic Container Registry (Amazon ECR)	AWS::ECR::PublicRepository	ECR.4

AWS service	Resource types	Related controls
	AWS::ECR::Repository	ECR.2 ECR.3 ECR.5
Amazon Elastic Container Service (Amazon ECS)	AWS::ECS::Cluster	ECS.12 ECS.14
	AWS::ECS::Service	ECS.2 ECS.10 ECS.13
	AWS::ECS::TaskDefinition	ECS.1 ECS.3 ECS.4 ECS.5 ECS.8 ECS.9 ECS.15 ECS.17
	AWS::ECS::TaskSet	ECS.16
Amazon Elastic File System (Amazon EFS)	AWS::EFS::AccessPoint	EFS.3 EFS.4 EFS.5

AWS service	Resource types	Related controls
Amazon Elastic Kubernetes Service (Amazon EKS)	AWS::EFS::FileSystem	EFS.7 EFS.8
	AWS::EKS::Cluster	EKS.2 EKS.6 EKS.8
	AWS::EKS::IdentityProviderConfig	EKS.7
AWS Elastic Beanstalk	AWS::ElasticBeanstalk::Environment	ElasticBeanstalk.1 ElasticBeanstalk.2 ElasticBeanstalk.3
Elastic Load Balancing	AWS::ElasticLoadBalancing::LoadBalancer	ELB.2 ELB.3 ELB.5 ELB.7 ELB.8 ELB.9 ELB.10 ELB.14
	AWS::ElasticLoadBalancingV2::Listener	ELB.17 ELB.18

AWS service	Resource types	Related controls
	`AWS::ElasticLoadBalancingV2::LoadBalancer`	ELB.1 ELB.4 ELB.5 ELB.6 ELB.12 ELB.13 ELB.16
ElasticSearch	`AWS::Elasticsearch::Domain`	ES.3 ES.4 ES.5 ES.6 ES.7 ES.8 ES.9
Amazon EMR	`AWS::EMR::SecurityConfiguration`	EMR.3 EMR.4
Amazon EventBridge	`AWS::Events::EventBus`	EventBridge.2 EventBridge.3
	`AWS::Events::Endpoint`	EventBridge.4

AWS service	Resource types	Related controls
Amazon Fraud Detector	`AWS::FraudDetector::EntityType`	FraudDetector.1
	`AWS::FraudDetector::Label`	FraudDetector.2
	`AWS::FraudDetector::Outcome`	FraudDetector.3
	`AWS::FraudDetector::Variable`	FraudDetector.4
AWS Global Accelerator	`AWS::GlobalAccelerator::Accelerator`	GlobalAccelerator.1
AWS Glue	`AWS::Glue::Job`	Glue.1 Glue.4
	`AWS::Glue::MLTransform`	Glue.3
Amazon GuardDuty	`AWS::GuardDuty::Detector`	GuardDuty.4
	`AWS::GuardDuty::Filter`	GuardDuty.2
	`AWS::GuardDuty::IPSet`	GuardDuty.3

AWS service	Resource types	Related controls
AWS Identity and Access Management (IAM)	`AWS::IAM::Group`	IAM.27 KMS.2
	`AWS::IAM::Policy`	IAM.1 IAM.21 KMS.1
	`AWS::IAM::Role`	IAM.24 IAM.27 KMS.2
	`AWS::IAM::User`	IAM.2 IAM.3 IAM.5 IAM.8 IAM.19 IAM.22 IAM.25 IAM.27 KMS.2
AWS Identity and Access Management Access Analyzer	`AWS::AccessAnalyzer::Analyzer`	IAM.23

AWS service	Resource types	Related controls
Amazon Interactive Video Service (Amazon IVS)	`AWS::IVS::PlaybackKeyPair`	IVS.1
	`AWS::IVS::RecordingConfiguration`	IVS.2
	`AWS::IVS::Channel`	IVS.3
AWS IoT	`AWS::IoT::Authorizer`	IoT.4
	`AWS::IoT::Dimension`	IoT.3
	`AWS::IoT::MitigationAction`	IoT.2
	`AWS::IoT::Policy`	IoT.6
	`AWS::IoT::RoleAlias`	IoT.5
	`AWS::IoT::SecurityProfile`	IoT.1
AWS IoT Events	`AWS::IoTEvents::AlarmModel`	IoTEvents.3

AWS service	Resource types	Related controls
	AWS::IoTEvents::DetectorModel	IoTEvents.2
	AWS::IoTEvents::Input	IoTEvents.1
AWS IoT SiteWise	AWS::IoTSiteWise::AssetModel	IoTSiteWise.1
	AWS::IoTSiteWise::Dashboard	IoTSiteWise.2
	AWS::IoTSiteWise::Gateway	IoTSiteWise.3
	AWS::IoTSiteWise::Portal	IoTSiteWise.4
	AWS::IoTSiteWise::Project	IoTSiteWise.5
AWS IoT TwinMaker	AWS::IoTTwinMaker::Entity	IoTTwinMaker.4
	AWS::IoTTwinMaker::Scene	IoTTwinMaker.3

AWS service	Resource types	Related controls
	`AWS::IoTTwinMaker::SyncJob`	IoTTwinMaker.1
	`AWS::IoTTwinMaker::Workspace`	IoTTwinMaker.2
AWS IoT Wireless	`AWS::IoTWireless::MulticastGroup`	IoTWireless.1
	`AWS::IoTWireless::ServiceProfile`	IoTWireless.2
	`AWS::IoTWireless::FuotaTask`	IoTWireless.3
Amazon Keyspaces (for Apache Cassandra)	`AWS::Cassandra::Keyspace`	Keyspaces.1
Amazon Kinesis	`AWS::Kinesis::Stream`	Kinesis.1 Kinesis.2 Kinesis.3
AWS Key Management Service (AWS KMS)	`AWS::KMS::Alias`	S3.17
	`AWS::KMS::Key`	KMS.3 KMS.5 S3.17

AWS service	Resource types	Related controls
AWS Lambda	`AWS::Lambda::Function`	Lambda.1 Lambda.2 Lambda.3 Lambda.5 Lambda.6 Lambda.7
Amazon MSK	`AWS::MSK::Cluster`	MSK.1 MSK.2 MSK.4 MSK.6
	`AWS::KafkaConnect::Connector`	MSK.3 MSK.5
Amazon MQ	`AWS::AmazonMQ::Broker`	MQ.2 MQ.3 MQ.4 MQ.5 MQ.6

AWS service	Resource types	Related controls
AWS Network Firewall	`AWS::NetworkFirewall::Firewall`	NetworkFirewall.1 NetworkFirewall.7 NetworkFirewall.9 NetworkFirewall.10
	`AWS::NetworkFirewall::FirewallPolicy`	NetworkFirewall.3 NetworkFirewall.4 NetworkFirewall.5 NetworkFirewall.8
	`AWS::NetworkFirewall::RuleGroup`	NetworkFirewall.6

AWS service	Resource types	Related controls
Amazon OpenSearch Service	`AWS::OpenSearch::Domain`	Opensearch.1 Opensearch.2 Opensearch.3 Opensearch.4 Opensearch.5 Opensearch.6 Opensearch.7 Opensearch.8 Opensearch.9 Opensearch.10 Opensearch.11
AWS Private CA	`AWS::ACMPCA::CertificateAuthority`	PCA.2

AWS service	Resource types	Related controls
Amazon Relational Database Service (Amazon RDS)	`AWS::RDS::DBCluster`	DocumentDB.1 DocumentDB.2 DocumentDB.4 DocumentDB.5 Neptune.1 Neptune.2 Neptune.4 Neptune.5 Neptune.7 Neptune.8 Neptune.9 RDS.7 RDS.12 RDS.14 RDS.15 RDS.16 RDS.24 RDS.27 RDS.28 RDS.34 RDS.35

AWS service	Resource types	Related controls
		RDS.37
	AWS::RDS::DBClusterSnapshot	DocumentDB.3 Neptune.3 Neptune.6 RDS.1 RDS.4 RDS.29

AWS service	Resource types	Related controls
	AWS::RDS::DBInstance	RDS.2 RDS.3 RDS.5 RDS.6 RDS.8 RDS.9 RDS.10 RDS.11 RDS.13 RDS.17 RDS.18 RDS.23 RDS.25 RDS.30 RDS.36 RDS.40
	AWS::RDS::DBSecurityGroup	RDS.31

AWS service	Resource types	Related controls
	AWS::RDS::DBSnapshot	RDS.1 RDS.4 RDS.32
	AWS::RDS::DBSubnetGroup	RDS.33
	AWS::RDS::EventSubscription	RDS.19 RDS.20 RDS.21 RDS.22
Amazon Redshift	AWS::Redshift::Cluster	Redshift.1 Redshift.2 Redshift.3 Redshift.4 Redshift.6 Redshift.7 Redshift.8 Redshift.9 Redshift.10 Redshift.11 Redshift.18

AWS service	Resource types	Related controls
	AWS::Redshift::ClusterParameterGroup	Redshift.2 Redshift.17
	AWS::Redshift::ClusterSnapshot	Redshift.13
	AWS::Redshift::ClusterSubnetGroup	Redshift.14 Redshift.16
	AWS::Redshift::EventSubscription	Redshift.12
Amazon Route 53	AWS::Route53::HostedZone	Route53.2
	AWS::Route53::HealthCheck	Route53.1
Amazon Simple Storage Service (Amazon S3)	AWS::S3::AccessPoint	S3.19
	AWS::S3::AccountPublicAccessBlock	S3.2 S3.3

AWS service	Resource types	Related controls
	AWS::S3::Bucket	CloudTrail.6 CloudTrail.7 S3.2 S3.3 S3.5 S3.6 S3.7 S3.8 S3.9 S3.10 S3.11 S3.12 S3.13 S3.14 S3.15 S3.17 S3.20
	AWS::S3::MultiRegionAccessPoint	S3.24

AWS service	Resource types	Related controls
	`AWS::S3Express::DirectoryBucket`	S3.25
Amazon SageMaker AI	`AWS::SageMaker::AppImageConfig`	SageMaker.6
	`AWS::SageMaker::Image`	SageMaker.7
	`AWS::SageMaker::Model`	SageMaker.5
	`AWS::SageMaker::NotebookInstance`	SageMaker.2 SageMaker.3
AWS Secrets Manager	`AWS::SecretsManager::Secret`	SecretsManager.1 SecretsManager.2 SecretsManager.5
AWS Service Catalog	`AWS::ServiceCatalog::Portfolio`	ServiceCatalog.1
Amazon Simple Email Service (Amazon SES)	`AWS::SES::ConfigurationSet`	SES.2
	`AWS::SES::ContactList`	SES.1

AWS service	Resource types	Related controls
Amazon Simple Notification Service (Amazon SNS)	`AWS::SNS::Topic`	SNS.1 SNS.3 SNS.4
Amazon Simple Queue Service (Amazon SQS)	`AWS::SQS::Queue`	SQS.1 SQS.2 SQS.3
AWS Step Functions	`AWS::StepFunctions::StateMachine`	StepFunctions.1
	`AWS::StepFunctions::Activity`	StepFunctions.2
AWS Systems Manager (SSM)	`AWS::SSM::Document`	SSM.5
AWS Transfer Family	`AWS::Transfer::Agreement`	Transfer.4
	`AWS::Transfer::Certificate`	Transfer.5
	`AWS::Transfer::Connector`	Transfer.3 Transfer.6
	`AWS::Transfer::Profile`	Transfer.7

AWS service	Resource types	Related controls
	AWS::Transfer::Workflow	Transfer.1
AWS WAF	AWS::WAF::Rule	WAF.6
	AWS::WAF::RuleGroup	WAF.7
	AWS::WAF::WebACL	WAF.1 WAF.8
	AWS::WAFRegional::Rule	WAF.2
	AWS::WAFRegional::RuleGroup	WAF.3
	AWS::WAFRegional::WebACL	WAF.4
	AWS::WAFv2::RuleGroup	WAF.12
	AWS::WAFv2::WebACL	WAF.10 WAF.11
Amazon WorkSpaces	AWS::WorkSpaces::WorkSpace	WorkSpaces.1 WorkSpaces.2

Required resources for the AWS Foundational Security Best Practices standard

For Security Hub CSPM to accurately report findings for change triggered controls that apply to the AWS Foundational Security Best Practices standard (v.1.0.0), are enabled, and use an AWS Config rule, you must record the following types of resources in AWS Config. For information about this standard, see AWS Foundational Security Best Practices standard in Security Hub CSPM.

AWS service	Resource types
Amazon API Gateway	`AWS::ApiGateway::Stage` , `AWS::ApiGatewayV2::Stage`
AWS AppSync	`AWS::AppSync::ApiCache` , `AWS::AppSync::GraphQLApi`
AWS Backup	`AWS::Backup::RecoveryPoint`
AWS Certificate Manager (ACM)	`AWS::ACM::Certificate`
AWS CloudFormation	`AWS::CloudFormation::Stack`
Amazon CloudFront	`AWS::CloudFront::Distribution`
AWS CodeBuild	`AWS::CodeBuild::Project` , `AWS::CodeBuild::ReportGroup`
Amazon Cognito	`AWS::Cognito::IdentityPool` , `AWS::Cognito::UserPool`
Amazon Connect	`AWS::Connect::Instance`
AWS DataSync	`AWS::DataSync::Task`
AWS Database Migration Service (AWS DMS)	`AWS::DMS::Endpoint` , `AWS::DMS::ReplicationInstance` , `AWS::DMS::ReplicationTask`
Amazon DynamoDB	`AWS::DynamoDB::Table`
Amazon EC2 Systems Manager (SSM)	`AWS::SSM::AssociationCompliance` , `AWS::SSM::ManagedInstanceIn`

AWS service	Resource types
	ventory ,AWS::SSM::PatchCompliance
Amazon Elastic Compute Cloud (Amazon EC2)	AWS::EC2::ClientVpnEndpoint ,AWS::EC2::Instance ,AWS::EC2::LaunchTemplate ,AWS::EC2::NetworkAcl ,AWS::EC2::NetworkInterface ,AWS::EC2::SecurityGroup ,AWS::EC2::SpotFleet ,AWS::EC2::Subnet ,AWS::EC2::TransitGateway ,AWS::EC2::VPCBlockPublicAccessOptions ,AWS::EC2::VPNConnection ,AWS::EC2::Volume
Amazon EC2 Auto Scaling	AWS::AutoScaling::AutoScalingGroup ,AWS::AutoScaling::LaunchConfiguration
Amazon Elastic Container Registry (Amazon ECR)	AWS::ECR::Repository
Amazon Elastic Container Service (Amazon ECS)	AWS::ECS::Cluster ,AWS::ECS::Service ,AWS::ECS::TaskDefinition ,AWS::ECS::TaskSet
Amazon Elastic File System (Amazon EFS)	AWS::EFS::AccessPoint ,AWS::EFS::FileSystem
Amazon Elastic Kubernetes Service (Amazon EKS)	AWS::EKS::Cluster
AWS Elastic Beanstalk	AWS::ElasticBeanstalk::Environment

AWS service	Resource types
Elastic Load Balancing	AWS::ElasticLoadBalancing::LoadBalancer , AWS::ElasticLoadBalancingV2::Listener , AWS::ElasticLoadBalancingV2::LoadBalancer
ElasticSearch	AWS::Elasticsearch::Domain
Amazon EMR	AWS::EMR::SecurityConfiguration
AWS Glue	AWS::Glue::Job , AWS::Glue::MLTransform
AWS Identity and Access Management (IAM)	AWS::IAM::Group , AWS::IAM::Policy , AWS::IAM::Role , AWS::IAM::User
Amazon Kinesis	AWS::Kinesis::Stream
AWS Key Management Service (AWS KMS)	AWS::KMS::Key
AWS Lambda	AWS::Lambda::Function
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	AWS::MSK::Cluster , AWS::KafkaConnect::Connector
AWS Network Firewall	AWS::NetworkFirewall::Firewall , AWS::NetworkFirewall::FirewallPolicy , AWS::NetworkFirewall::RuleGroup
Amazon OpenSearch Service	AWS::OpenSearch::Domain
Amazon Relational Database Service (Amazon RDS)	AWS::RDS::DBCluster , AWS::RDS::DBClusterSnapshot , AWS::RDS::DBInstance , AWS::RDS::DBProxy , AWS::RDS::DBSnapshot , AWS::RDS::EventSubscription

AWS service	Resource types
Amazon Redshift	AWS::Redshift::Cluster , AWS::Redshift::ClusterSubnetGroup
Amazon Redshift Serverless	AWS::RedshiftServerless::Workgroup
Amazon Route 53	AWS::Route53::HostedZone
Amazon Simple Storage Service (Amazon S3)	AWS::S3::AccessPoint , AWS::S3::AccountPublicAccessBlock , AWS::S3::Bucket , AWS::S3::MultiRegionAccessPoint , AWS::S3Express::DirectoryBucket
Amazon SageMaker AI	AWS::SageMaker::Model , AWS::SageMaker::NotebookInstance
Amazon Simple Notification Service (Amazon SNS)	AWS::SNS::Topic
Amazon Simple Queue Service (Amazon SQS)	AWS::SQS::Queue
AWS Secrets Manager	AWS::SecretsManager::Secret
AWS Step Functions	AWS::StepFunctions::StateMachine
AWS Transfer Family	AWS::Transfer::Connector
AWS WAF	AWS::WAF::Rule , AWS::WAF::RuleGroup , AWS::WAF::WebACL , AWS::WAFRegional::Rule , AWS::WAFRegional::RuleGroup , AWS::WAFRegional::WebACL , AWS::WAFv2::RuleGroup , AWS::WAFv2::WebACL
Amazon WorkSpaces	AWS::WorkSpaces::WorkSpace

Required resources for the CIS AWS Foundations Benchmark

To run security checks for enabled controls that apply to the Center for Internet Security (CIS) AWS Foundations Benchmark, Security Hub CSPM either runs through the exact audit steps prescribed for the checks or uses specific AWS Config managed rules. For information about this standard in Security Hub CSPM, see CIS AWS Foundations Benchmark in Security Hub CSPM.

Required resources for CIS v3.0.0

For Security Hub CSPM to accurately report findings for enabled CIS v3.0.0 change triggered controls that use an AWS Config rule, you must record the following types of resources in AWS Config.

AWS service	Resource types
Amazon Elastic Compute Cloud (Amazon EC2)	`AWS::EC2::Instance` , `AWS::EC2::NetworkAcl` , `AWS::EC2::SecurityGroup`
AWS Identity and Access Management (IAM)	`AWS::IAM::Group` , `AWS::IAM::User` , `AWS::IAM::Role`
Amazon Relational Database Service (Amazon RDS)	`AWS::RDS::DBInstance`
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::Bucket`

Required resources for CIS v1.4.0

For Security Hub CSPM to accurately report findings for enabled CIS v1.4.0 change triggered controls that use an AWS Config rule, you must record the following types of resources in AWS Config.

AWS service	Resource types
Amazon Elastic Compute Cloud (Amazon EC2)	`AWS::EC2::NetworkAcl` , `AWS::EC2::SecurityGroup`
AWS Identity and Access Management (IAM)	`AWS::IAM::Policy` , `AWS::IAM::User`

AWS service	Resource types
Amazon Relational Database Service (Amazon RDS)	`AWS::RDS::DBInstance`
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::Bucket`

Required resources for CIS v1.2.0

For Security Hub CSPM to accurately report findings for enabled CIS v1.2.0 change triggered controls that use an AWS Config rule, you must record the following types of resources in AWS Config.

AWS service	Resource types
Amazon Elastic Compute Cloud (Amazon EC2)	`AWS::EC2::SecurityGroup`
AWS Identity and Access Management (IAM)	`AWS::IAM::Policy` , `AWS::IAM::User`

Required resources for the NIST SP 800-53 Revision 5 standard

For Security Hub CSPM to accurately report findings for change triggered controls that apply to the NIST SP 800-53 Revision 5 standard, are enabled, and use an AWS Config rule, you must record the following types of resources in AWS Config. For information about this standard, see NIST SP 800-53 Revision 5 in Security Hub CSPM.

AWS service	Resource types
Amazon API Gateway	`AWS::ApiGateway::Stage` , `AWS::ApiGatewayV2::Stage`
AWS AppSync	`AWS::AppSync::GraphQLApi`
AWS Backup	`AWS::Backup::RecoveryPoint`
AWS Certificate Manager (ACM)	`AWS::ACM::Certificate`
AWS CloudFormation	`AWS::CloudFormation::Stack`

AWS service	Resource types
Amazon CloudFront	AWS::CloudFront::Distribution
Amazon CloudWatch	AWS::CloudWatch::Alarm
AWS CodeBuild	AWS::CodeBuild::Project
AWS Database Migration Service (AWS DMS)	AWS::DMS::Endpoint , AWS::DMS::ReplicationInstance , AWS::DMS::ReplicationTask
Amazon DynamoDB	AWS::DynamoDB::Table
Amazon Elastic Compute Cloud (Amazon EC2)	AWS::EC2::ClientVpnEndpoint , AWS::EC2::EIP , AWS::EC2::Instance , AWS::EC2::LaunchTemplate , AWS::EC2::NetworkAcl , AWS::EC2::NetworkInterface , AWS::EC2::SecurityGroup , AWS::EC2::Subnet , AWS::EC2::TransitGateway , AWS::EC2::VPNConnection , AWS::EC2::Volume
Amazon EC2 Auto Scaling	AWS::AutoScaling::AutoScalingGroup , AWS::AutoScaling::LaunchConfiguration
Amazon Elastic Container Registry (Amazon ECR)	AWS::ECR::Repository
Amazon Elastic Container Service (Amazon ECS)	AWS::ECS::Cluster , AWS::ECS::Service , AWS::ECS::TaskDefinition
Amazon Elastic File System (Amazon EFS)	AWS::EFS::AccessPoint
Amazon Elastic Kubernetes Service (Amazon EKS)	AWS::EKS::Cluster

AWS service	Resource types
AWS Elastic Beanstalk	`AWS::ElasticBeanstalk::Environment`
Elastic Load Balancing	`AWS::ElasticLoadBalancing::LoadBalancer` , `AWS::ElasticLoadBalancingV2::Listener` , `AWS::ElasticLoadBalancingV2::LoadBalancer`
Amazon ElasticSearch	`AWS::Elasticsearch::Domain`
Amazon EMR	`AWS::EMR::SecurityConfiguration`
Amazon EventBridge	`AWS::Events::Endpoint` , `AWS::Events::EventBus`
AWS Glue	`AWS::Glue::Job`
AWS Identity and Access Management (IAM)	`AWS::IAM::Group` , `AWS::IAM::Policy` , `AWS::IAM::Role` , `AWS::IAM::User`
AWS Key Management Service (AWS KMS)	`AWS::KMS::Alias` , `AWS::KMS::Key`
Amazon Kinesis	`AWS::Kinesis::Stream`
AWS Lambda	`AWS::Lambda::Function`
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	`AWS::MSK::Cluster`
Amazon MQ	`AWS::AmazonMQ::Broker`
AWS Network Firewall	`AWS::NetworkFirewall::Firewall` , `AWS::NetworkFirewall::FirewallPolicy` , `AWS::NetworkFirewall::RuleGroup`
Amazon OpenSearch Service	`AWS::OpenSearch::Domain`

AWS service	Resource types
Amazon Relational Database Service (Amazon RDS)	AWS::RDS::DBCluster , AWS::RDS::DBClusterSnapshot , AWS::RDS::DBInstance , AWS::RDS::DBSnapshot , AWS::RDS::EventSubscription
Amazon Redshift	AWS::Redshift::Cluster , AWS::Redshift::ClusterSubnetGroup
Amazon Route 53	AWS::Route53::HostedZone
Amazon Simple Storage Service (Amazon S3)	AWS::S3::AccessPoint , AWS::S3::AccountPublicAccessBlock , AWS::S3::Bucket
AWS Service Catalog	AWS::ServiceCatalog::Portfolio
Amazon Simple Notification Service (Amazon SNS)	AWS::SNS::Topic
Amazon Simple Queue Service (Amazon SQS)	AWS::SQS::Queue
Amazon EC2 Systems Manager (SSM)	AWS::SSM::AssociationCompliance , AWS::SSM::ManagedInstanceInventory , AWS::SSM::PatchCompliance
Amazon SageMaker AI	AWS::SageMaker::NotebookInstance
AWS Secrets Manager	AWS::SecretsManager::Secret
AWS Transfer Family	AWS::Transfer::Connector

AWS service	Resource types
AWS WAF	AWS::WAF::Rule , AWS::WAF::RuleGroup , AWS::WAF::WebACL , AWS::WAFRegional::Rule , AWS::WAFRegional::RuleGroup , AWS::WAFRegional::WebACL , AWS::WAFv2::RuleGroup , AWS::WAFv2::WebACL

Required resources for the NIST SP 800-171 Revision 2 standard

For Security Hub CSPM to accurately report findings for change triggered controls that apply to the NIST SP 800-171 Revision 2 standard, are enabled, and use an AWS Config rule, you must record the following types of resources in AWS Config. For information about this standard, see NIST SP 800-171 Revision 2 in Security Hub CSPM.

AWS service	Resource types
AWS Certificate Manager (ACM)	AWS::ACM::Certificate
Amazon API Gateway	AWS::ApiGateway::Stage
Amazon CloudFront	AWS::CloudFront::Distribution
Amazon CloudWatch	AWS::CloudWatch::Alarm
Amazon Elastic Compute Cloud (Amazon EC2)	AWS::EC2::ClientVpnEndpoint , AWS::EC2::NetworkAcl , AWS::EC2::SecurityGroup , AWS::EC2::VPC , AWS::EC2::VPNConnection
Elastic Load Balancing	AWS::ElasticLoadBalancing::LoadBalancer
AWS Identity and Access Management (IAM)	AWS::IAM::Policy , AWS::IAM::User
AWS Key Management Service (AWS KMS)	AWS::KMS::Alias , AWS::KMS::Key

AWS service	Resource types
AWS Network Firewall	`AWS::NetworkFirewall::FirewallPolicy` , `AWS::NetworkFirewall::RuleGroup`
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::Bucket`
Amazon Simple Notification Service (Amazon SNS)	`AWS::SNS::Topic`
AWS Systems Manager (SSM)	`AWS::SSM::PatchCompliance`
AWS WAF	`AWS::WAFv2::RuleGroup`

Required resources for PCI DSS v3.2.1

For Security Hub CSPM to accurately report findings for controls that apply to v3.2.1 of the Payment Card Industry Data Security Standard (PCI DSS), are enabled, and use an AWS Config rule, you must record the following types of resources in AWS Config. For information about this standard, see PCI DSS in Security Hub CSPM.

AWS service	Resource types
AWS CodeBuild	`AWS::CodeBuild::Project`
Amazon Elastic Compute Cloud (Amazon EC2)	`AWS::EC2::EIP` , `AWS::EC2::Instance` , `AWS::EC2::SecurityGroup`
Amazon EC2 Auto Scaling	`AWS::AutoScaling::AutoScalingGroup`
AWS Identity and Access Management (IAM)	`AWS::IAM::Policy` , `AWS::IAM::User`
AWS Lambda	`AWS::Lambda::Function`
Amazon OpenSearch Service	`AWS::OpenSearch::Domain`

AWS service	Resource types
Amazon Relational Database Service (Amazon RDS)	`AWS::RDS::DBClusterSnapshot` , `AWS::RDS::DBInstance` , `AWS::RDS::DBSnapshot`
Amazon Redshift	`AWS::Redshift::Cluster`
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::AccountPublicAccessBlock` , `AWS::S3::Bucket`
Amazon EC2 Systems Manager (SSM)	`AWS::SSM::AssociationCompliance` , `AWS::SSM::ManagedInstanceInventory` , `AWS::SSM::PatchCompliance`

Required resources for the AWS Resource Tagging standard

All the controls that apply to the AWS Resource Tagging standard are change triggered and use an AWS Config rule. For Security Hub CSPM to accurately report findings for these controls, you must record the following types of resources in AWS Config. For information about this standard, see AWS Resource Tagging standard in Security Hub CSPM.

AWS service	Resource types
AWS Amplify	`AWS::Amplify::App` , `AWS::Amplify::Branch`
Amazon AppFlow	`AWS::AppFlow::Flow`
AWS App Runner	`AWS::AppRunner::Service` , `AWS::AppRunner::VpcConnector`
AWS AppConfig	`AWS::AppConfig::Application` , `AWS::AppConfig::ConfigurationProfile` , `AWS::AppConfig::Environment` , `AWS::AppConfig::ExtensionAssociation`

AWS service	Resource types
AWS AppSync	AWS::AppSync::GraphQLApi
Amazon Athena	AWS::Athena::DataCatalog , AWS::Athena::WorkGroup
AWS Backup	AWS::Backup::BackupPlan ,AWS::Backup::BackupVault ,AWS::Backup::RecoveryPlan ,AWS::Backup::ReportPlan
AWS Batch	AWS::Batch::ComputeEnvironment , AWS::Batch::JobQueue ,AWS::Batch::SchedulingPolicy
AWS Certificate Manager (ACM)	AWS::ACM::Certificate
AWS CloudFormation	AWS::CloudFormation::Stack
Amazon CloudFront	AWS::CloudFront::Distribution
AWS CloudTrail	AWS::CloudTrail::Trail
AWS CodeArtifact	AWS::CodeArtifact::Repository
Amazon CodeGuru	AWS::CodeGuruProfiler::ProfilingGroup ,AWS::CodeGuruReviewer::RepositoryAssociation
Amazon Connect	AWS::CustomerProfiles::ObjectType
AWS Database Migration Service (AWS DMS)	AWS::DMS::Certificate ,AWS::DMS::EventSubscription AWS::DMS::ReplicationInstance , AWS::DMS::ReplicationSubnetGroup
AWS DataSync	AWS::DataSync::Task

AWS service	Resource types
Amazon Detective	AWS::Detective::Graph
Amazon DynamoDB	AWS::DynamoDB::Trail
Amazon Elastic Compute Cloud (EC2)	AWS::EC2::CustomerGateway , AWS::EC2::DHCPOptions ,AWS::EC2::EIP ,AWS::EC2::FlowLog ,AWS::EC2::Instance ,AWS::EC2::InternetGateway ,AWS::EC2::LaunchTemplate , AWS::EC2::NatGateway ,AWS::EC2::NetworkAcl ,AWS::EC2::NetworkInterface ,AWS::EC2::PrefixList , AWS::EC2::RouteTable ,AWS::EC2::SecurityGroup ,AWS::EC2::Subnet , AWS::EC2::TrafficMirrorFilter ,AWS::EC2::TrafficMirrorSession ,AWS::EC2::TrafficMirrorTarget ,AWS::EC2::TransitGateway , AWS::EC2::TransitGatewayAttachment ,AWS::EC2::TransitGatewayRouteTable ,AWS::EC2::Volume ,AWS::EC2::VPC ,AWS::EC2::VPCEndpointService ,AWS::EC2::VPCPeeringConnection ,AWS::EC2::VPNGateway
Amazon EC2 Auto Scaling	AWS::AutoScaling::AutoScalingGroup
Amazon Elastic Container Registry (Amazon ECR)	AWS::ECR::PublicRepository
Amazon Elastic Container Service (Amazon ECS)	AWS::ECS::Cluster ,AWS::ECS::Service ,AWS::ECS::TaskDefinition

AWS service	Resource types
Amazon Elastic File System (Amazon EFS)	AWS::EFS::AccessPoint
Amazon Elastic Kubernetes Service (Amazon EKS)	AWS::EKS::Cluster ,AWS::EKS::IdentityProviderConfig
AWS Elastic Beanstalk	AWS::ElasticBeanstalk::Environment
ElasticSearch	AWS::Elasticsearch::Domain
Amazon EventBridge	AWS::Events::EventBus
Amazon Fraud Detector	AWS::FraudDetector::EntityType , AWS::FraudDetector::Label AWS::FraudDetector::Outcome , AWS::FraudDetector::Variable
AWS Global Accelerator	AWS::GlobalAccelerator::Accelerator
AWS Glue	AWS::Glue::Job
Amazon GuardDuty	AWS::GuardDuty::Detector , AWS::GuardDuty::Filter ,AWS::GuardDuty::IPSet
AWS Identity and Access Management (IAM)	AWS::IAM::Role ,AWS::IAM::User
AWS Identity and Access Management Access Analyzer (IAM Access Analyzer)	AWS::AccessAnalyzer::Analyzer
AWS IoT	AWS::IoT::Authorizer ,AWS::IoT::Dimension ,AWS::IoT::MitigationAction ,AWS::IoT::Policy , AWS::IoT::RoleAlias ,AWS::IoT::SecurityProfile

AWS service	Resource types
AWS IoT Events	AWS::IoTEvents::AlarmModel , AWS::IoTEvents::DetectorModel , AWS::IoTEvents::Input
AWS IoT SiteWise	AWS::IoTSiteWise::Dashboard ,AWS::IoTSiteWise::Gateway , AWS::IoTSiteWise::Portal , AWS::IoTSiteWise::Project
AWS IoT TwinMaker	AWS::IoTTwinMaker::Entity , AWS::IoTTwinMaker::Scene , AWS::IoTTwinMaker::SyncJob , AWS::IoTTwinMaker::Workspace
AWS IoT Wireless	AWS::IoTWireless::FuotaTask ,AWS::IoTWireless::Multicast Group ,AWS::IoTWireless:: ServiceProfile
Amazon Interactive Video Service (Amazon IVS)	AWS::IVS::Channel ,AWS::IVS: :PlaybackKeyPair ,AWS::IVS: :RecordingConfiguration
Amazon Keyspaces (for Apache Cassandra)	AWS::Cassandra::Keyspace
Amazon Kinesis	AWS::Kinesis::Stream
AWS Lambda	AWS::Lambda::Function
Amazon MQ	AWS::AmazonMQ::Broker
AWS Network Firewall	AWS::NetworkFirewall::Firew all ,AWS::NetworkFirewall::Firew allPolicy
Amazon OpenSearch Service	AWS::OpenSearch::Domain

AWS service	Resource types
AWS Private Certificate Authority	`AWS::ACMPCA::CertificateAuthority`
Amazon Relational Database Service	`AWS::RDS::DBCluster` , `AWS::RDS::DBClusterSnapshot` , `AWS::RDS::DBInstance` , `AWS::RDS::DBSecurityGroup` , `AWS::RDS::DBSnapshot` , `AWS::RDS::DBSubnetGroup`
Amazon Redshift	`AWS::Redshift::Cluster` , `AWS::Redshift::ClusterParameterGroup` , `AWS::Redshift::ClusterSnapshot` , `AWS::Redshift::ClusterSubnetGroup` , `AWS::Redshift::EventSubscription`
Amazon Route 53	`AWS::Route53::HealthCheck`
Amazon SageMaker AI	`AWS::SageMaker::AppImageConfig` , `AWS::SageMaker::Image`
AWS Secrets Manager	`AWS::SecretsManager::Secret`
Amazon Simple Email Service (Amazon SES)	`AWS::SES::ConfigurationSet` , `AWS::SES::ContactList`
Amazon Simple Notification Service (Amazon SNS)	`AWS::SNS::Topic`
Amazon Simple Queue Service (Amazon SQS)	`AWS::SQS::Queue`
AWS Step Functions	`AWS::StepFunctions::Activity`
AWS Systems Manager (SSM)	`AWS::SSM::Document`

AWS service	Resource types
AWS Transfer Family	`AWS::Transfer::Agreement` , `AWS::Transfer::Certificate` , `AWS::Transfer::Connector` , `AWS::Transfer::Profile` , `AWS::Transfer::Workflow`

Required resources for the AWS Control Tower service-managed standard

For Security Hub CSPM to accurately report findings for change triggered controls that apply to the AWS Control Tower service-managed standard, are enabled, and use an AWS Config rule, you must record the following types of resources in AWS Config. For information about this standard, see Service-Managed Standard: AWS Control Tower.

AWS service	Resource types
Amazon API Gateway	`AWS::ApiGateway::Stage` `AWS::ApiGatewayV2::Stage`
AWS Certificate Manager (ACM)	`AWS::ACM::Certificate`
AWS CodeBuild	`AWS::CodeBuild::Project`
Amazon DynamoDB	`AWS::DynamoDB::Table`

Amazon Elastic Compute Cloud (EC2)	`AWS::EC2::Instance`
	`AWS::EC2::NetworkAcl`
	`AWS::EC2::NetworkInterface`
	`AWS::EC2::SecurityGroup`
	`AWS::EC2::Subnet`
	`AWS::EC2::VPNConnection`

AWS service	Resource types
	AWS::EC2::Volume
Amazon EC2 Auto Scaling	AWS::AutoScaling::AutoScalingGroup AWS::AutoScaling::LaunchConfiguration
Amazon Elastic Container Registry (Amazon ECR)	AWS::ECR::Repository
Amazon Elastic Container Service (Amazon ECS)	AWS::ECS::Cluster AWS::ECS::Service AWS::ECS::TaskDefinition
Amazon Elastic File System (Amazon EFS)	AWS::EFS::AccessPoint
Amazon EKS	AWS::EKS::Cluster
ElasticBeanstalk	AWS::ElasticBeanstalk::Environment
Elastic Load Balancing	AWS::ElasticLoadBalancing::LoadBalancer AWS::ElasticLoadBalancingV2::LoadBalancer
ElasticSearch	AWS::Elasticsearch::Domain
AWS Identity and Access Management (IAM)	AWS::IAM::Group AWS::IAM::Policy AWS::IAM::Role AWS::IAM::User

AWS service	Resource types
AWS Key Management Service (AWS KMS)	`AWS::KMS::Alias` `AWS::KMS::Key`
Amazon Kinesis	`AWS::Kinesis::Stream`
AWS Lambda	`AWS::Lambda::Function`
AWS Network Firewall	`AWS::NetworkFirewall::FirewallPolicy` `AWS::NetworkFirewall::RuleGroup`
Amazon OpenSearch Service	`AWS::OpenSearch::Domain`

Amazon Relational Database Service (Amazon RDS)	`AWS::RDS::DBCluster`
	`AWS::RDS::DBClusterSnapshot`
	`AWS::RDS::DBInstance`
	`AWS::RDS::DBSnapshot`
	`AWS::RDS::EventSubscription`

Amazon Redshift	`AWS::Redshift::Cluster`
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::AccountPublicAccessBlock` `AWS::S3::Bucket`
Amazon Simple Notification Service (Amazon SNS)	`AWS::SNS::Topic`
Amazon Simple Queue Service (Amazon SQS)	`AWS::SQS::Queue`
AWS Secrets Manager	`AWS::SecretsManager::Secret`

AWS service	Resource types

Amazon EC2 Systems Manager (SSM)	`AWS::SSM::AssociationCompliance`
	`AWS::SSM::ManagedInstanceInventory`
	`AWS::SSM::PatchCompliance`

AWS WAF	`AWS::WAFRegional::Rule`
	`AWS::WAFRegional::RuleGroup`
	`AWS::WAFRegional::WebACL`
	`AWS::WAFv2::WebACL`

Schedule for running security checks

After you enable a security standard, AWS Security Hub CSPM begins to run all checks within two hours. Most checks begin to run within 25 minutes. Security Hub CSPM runs checks by evaluating the rule underlying a control. Until a control completes its first run of checks, its status is **No data**.

When you enable a new standard, it might take up to 24 hours for Security Hub CSPM to generate findings for controls that use the same underlying AWS Config service-linked rule as enabled controls from other enabled standards. For example, if you enable the Lambda.1 control in the AWS Foundational Security Best Practices (FSBP) standard, Security Hub CSPM creates the service-linked rule and typically generates findings within minutes. After this, if you enable the Lambda.1 control in the Payment Card Industry Data Security Standard (PCI DSS), it might take up to 24 hours for Security Hub CSPM to generate findings for the control because it uses the same service-linked rule.

After the initial check, the schedule for each control can be either periodic or change triggered. For a control that is based on a managed AWS Config rule, the control description includes a link to the rule description in the *AWS Config Developer Guide*. That description specifies whether the rule is change triggered or periodic.

Periodic security checks

Periodic security checks run automatically within 12 or 24 hours after the most recent run. Security Hub CSPM determines the periodicity, and you can't change it. Periodic controls reflect an evaluation at the moment the check runs.

If you update the workflow status of a periodic control finding, and then in the next check the compliance status of the finding stays the same, the workflow status remains in its modified state. For example, if you have a failed finding for the KMS.4 control (*AWS KMS key rotation should be enabled*), and then remediate the finding, Security Hub CSPM changes the workflow status from `NEW` to `RESOLVED`. If you disable KMS key rotation before the next periodic check, the workflow status of the finding remains `RESOLVED`.

Checks that use Security Hub CSPM custom Lambda functions are periodic.

Change-triggered security checks

Change-triggered security checks run when the associated resource changes state. AWS Config lets you choose between *continuous recording* of changes in resource state and *daily recording*. If you choose daily recording, AWS Config delivers resource configuration data at the end of each 24 hour period if there are changes in resource state. If there are no changes, no data is delivered. This may delay the generation of Security Hub CSPM findings until a 24-hour period is complete. Regardless of your chosen recording period, Security Hub CSPM checks every 18 hours to ensure no resource updates from AWS Config were missed.

In general, Security Hub CSPM uses change-triggered rules whenever possible. For a resource to use a change-triggered rule, it must support AWS Config configuration items.

Generating and updating control findings

AWS Security Hub CSPM generates and updates control findings when it runs checks against security controls. Control findings use the AWS Security Finding Format (ASFF).

Security Hub CSPM normally charges for each security check for a control. However, if multiple controls use the same AWS Config rule, Security Hub CSPM charges only once for each check against the rule. For example, the AWS Config `iam-password-policy` rule is used by multiple controls in the CIS AWS Foundations Benchmark standard and the AWS Foundational Security Best Practices standard. Each time Security Hub CSPM runs a check against that rule, it generates a separate control finding for each related control, but charges only once for the check.

If the size of a control finding exceeds the maximum of 240 KB, Security Hub CSPM removes the `Resource.Details` object from the finding. For controls that are backed by AWS Config resources, you can review resource details by using the AWS Config console.

Topics

Consolidated control findings

If consolidated control findings is enabled for your account, Security Hub CSPM generates a single finding or finding update for each security check of a control, even if a control applies to multiple enabled standards. For a list of controls and the standards that they apply to, see the Control reference for Security Hub CSPM. We recommend enabling consolidated control findings to reduce finding noise.

If you enabled Security Hub CSPM for an AWS account before February 23, 2023, you can enable consolidated control findings by following the instructions later in this section. If you enable Security Hub CSPM on or after February 23, 2023, consolidated control findings is enabled automatically for your account.

If you use the Security Hub CSPM integration with AWS Organizations or invited member accounts through a manual invitation process, consolidated control findings is enabled for member accounts only if it's enabled for the administrator account. If the feature is disabled for the administrator account, it's disabled for member accounts. This behavior applies to new and existing member accounts. In addition, if the administrator uses central configuration to manage Security Hub CSPM for multiple accounts, they cannot use central configuration policies to enable or disable consolidated control findings for the accounts.

If you disable consolidated control findings for your account, Security Hub CSPM generates or updates a separate control finding for each enabled standard that includes a control. For example, if you enable four standards that share a control, you receive four separate findings after a security check for the control. If you enable consolidated control findings, you receive only one finding.

When you enable consolidated control findings, Security Hub CSPM creates new standard-agnostic findings and archives the original standard-based findings. Some control finding fields and values will change, which might impact your existing workflows. For information about these changes, see Consolidated control findings – ASFF changes. Enabling consolidated control findings might also affect findings that integrated third-party products receive from Security Hub CSPM. If you use the Automated Security Response on AWS v2.0.0 solution, note that it supports consolidated control findings.

To enable or disable consolidated control findings, you must be signed in to an administrator account or a standalone account.

> **Note**
>
> After you enable consolidated control findings, it can take up to 24 hours for Security Hub CSPM to generate new consolidated findings and archive the existing standard-based findings. Similarly, after disabling consolidated control findings, it can take up to 24 hours for Security Hub CSPM to generate new standard-based findings and archive the existing consolidated findings. During these times, you might see a mix of standard-agnostic and standard-based findings in your account.

Security Hub CSPM console

To enable or disable consolidated control findings

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, under **Settings**, choose **General**.
3. In the **Controls** section, choose **Edit**.
4. Use the **Consolidated control findings** switch to enable or disable consolidated control findings.
5. Choose **Save**.

Security Hub CSPM API

To enable or disable consolidated control findings programmatically, use the UpdateSecurityHubConfiguration operation of the Security Hub CSPM API. Or, if you're using the AWS CLI, run the update-security-hub-configuration command.

For the `control-finding-generator` parameter, specify `SECURITY_CONTROL` to enable consolidated control findings. To disable consolidated control findings, specify `STANDARD_CONTROL`.

For example, the following AWS CLI command enables consolidated control findings.

```
$ aws securityhub  --region us-east-1 update-security-hub-configuration --control-finding-generator SECURITY_CONTROL
```

The following AWS CLI command disables consolidated control findings.

```
$ aws securityhub  --region us-east-1 update-security-hub-configuration --control-finding-generator STANDARD_CONTROL
```

Generating, updating, and archiving control findings

Security Hub CSPM runs security checks on a schedule. The first time Security Hub CSPM runs a security check for a control, it generates a new finding for each AWS resource that the control checks. Each time Security Hub CSPM subsequently runs a security check for the control, it updates existing findings to report the results of the check. This means that you can use the data provided by individual findings to track compliance changes for particular resources against particular controls.

For example, if the compliance status of a resource changes from `FAILED` to `PASSED` for a particular control, Security Hub CSPM doesn't generate a new finding. Instead, Security Hub CSPM updates the existing finding for the control and resource. In the finding, Security Hub CSPM changes the value for the compliance status (`Compliance.Status`) field to `PASSED`. Security Hub CSPM also updates the values for additional fields to reflect the results of the check—for example, the severity label, workflow status, and timestamps that indicate when Security Hub CSPM most recently ran the check and updated the finding.

When reporting changes to compliance status, Security Hub CSPM might update any of the following fields in a control finding:

- `Compliance.Status` – The new compliance status of the resource for the specified control.
- `FindingProviderFields.Severity.Label` – The new qualitative representation of the severity of the finding, such as `LOW`, `MEDIUM`, or `HIGH`.

- `FindingProviderFields.Severity.Original` – The new quantitative representation of the severity of the finding, such as `0` for a compliant resource.
- `FirstObservedAt` – When the compliance status of the resource most recently changed.
- `LastObservedAt` – When Security Hub CSPM most recently ran the security check for the specified control and resource.
- `ProcessedAt` – When Security Hub CSPM most recently began processing the finding.
- `ProductFields.PreviousComplianceStatus` – The previous compliance status (`Compliance.Status`) of the resource for the specified control.
- `UpdatedAt` – When Security Hub CSPM most recently updated the finding.
- `Workflow.Status` – The status of the investigation into the finding, based on the new compliance status of the resource for the specified control.

Whether Security Hub CSPM updates a field depends primarily on the results of the latest security check for the applicable control and resource. For example, if the compliance status of a resource changes from `PASSED` to `FAILED` for a particular control, Security Hub CSPM changes the workflow status of the finding to `NEW`. To track updates to individual findings, you can refer to the history of a finding. For details about individual fields in findings, see AWS Security Finding Format (ASFF).

In certain cases, Security Hub CSPM generates new findings for subsequent checks by a control, instead of updating existing findings. This can occur if there's an issue with the AWS Config rule that backs a control. If this happens, Security Hub CSPM archives the existing finding and generates a new finding for each check. In the new findings, the compliance status is `NOT_AVAILABLE` and the record state is `ARCHIVED`. After you address the issue with the AWS Config rule, Security Hub CSPM generates new findings and begins updating them to track subsequent changes to the compliance status of individual resources.

In addition to generating and updating control findings, Security Hub CSPM automatically archives control findings that meet certain criteria. Security Hub CSPM archives a finding if the control is disabled, the specified resource is deleted, or the specified resource no longer exists. A resource might not exist anymore because the associated service is no longer used. More specifically, Security Hub CSPM automatically archives a control finding if the finding meets one of the following criterion:

- The finding hasn't been updated for 3-5 days. Note that archival based on this time frame is on a best-effort basis and is not guaranteed.

- The associated AWS Config evaluation returned `NOT_APPLICABLE` for the compliance status of the specified resource.

To determine whether a finding is archived, you can refer to the record state (`RecordState`) field of the finding. If a finding is archived, the value for this field is `ARCHIVED`.

Security Hub CSPM stores archived control findings for 30 days. After 30 days, the findings expire and Security Hub CSPM permanently deletes them. To determine whether an archived control finding has expired, Security Hub CSPM bases its calculation on the value for the `UpdatedAt` field of the finding.

To store archived control findings for more than 30 days, you can export the findings to an S3 bucket. You can do this by using a custom action with an Amazon EventBridge rule. For more information, see Using EventBridge for automated response and remediation.

> **Note**
>
> Prior to July 3, 2025, Security Hub CSPM generated and updated control findings differently when the compliance status of a resource changed for a control. Previously, Security Hub CSPM created a new control finding and archived the existing finding for a resource. Therefore, you might have multiple archived findings for a particular control and resource until those findings expire (after 30 days).

Automation and suppression of control findings

You can use Security Hub CSPM automation rules to update or suppress specific control findings. If you suppress a finding, you can continue to access it. However, suppression indicates your belief that no action is needed to address the finding.

By suppressing findings, you can reduce finding noise. For example, you might suppress control findings that are generated in test accounts. Or, you might suppress findings related to specific resources. To learn more about updating or suppressing findings automatically, see Understanding automation rules in Security Hub CSPM.

Automation rules are appropriate when you want to update or suppress specific control findings. However, if a control isn't relevant to your organization or use case, we recommend disabling the control. If you disable a control, Security Hub CSPM doesn't run security checks for it and you aren't charged for it.

Compliance details for control findings

In findings generated by security checks for controls, the Compliance object and fields in the AWS Security Finding Format (ASFF) provide compliance details for individual resources that a control checked. This includes the following information:

- `AssociatedStandards` – The enabled standards that the control is enabled in.
- `RelatedRequirements` – The related requirements for the control in all enabled standards. These requirements derive from third-party security frameworks for the control, such as the Payment Card Industry Data Security Standard (PCI DSS) or the NIST SP 800-171 Revision 2 standard.
- `SecurityControlId` – The identifier for the control across the standards that Security Hub CSPM supports.
- `Status` – The result of the most recent check that Security Hub CSPM ran for the control. The results of previous checks are retained in the history of the finding.
- `StatusReasons` – An array that lists reasons for the value specified by the `Status` field. For each reason, this includes a reason code and a description.

The following table lists reason codes and descriptions that a finding might include in the `StatusReasons` array. The remediation steps vary based on which control generated a finding with a specified reason code. To review the remediation guidance for a control, refer to the Control reference for Security Hub CSPM.

Reason code	Compliance status	Description
`CLOUDTRAIL_METRIC_FILTER_NOT_VALID`	`FAILED`	The multi-Region CloudTrail trail does not have a valid metric filter.
`CLOUDTRAIL_METRIC_FILTERS_NOT_PRESENT`	`FAILED`	Metric filters are not present for the multi-Region CloudTrail trail.
`CLOUDTRAIL_MULTI_REGION_NOT_PRESENT`	`FAILED`	The account does not have a multi-Region CloudTrail trail with the required configuration.

Reason code	Compliance status	Description
CLOUDTRAIL_REGION_INVAILD	WARNING	Multi-Region CloudTrail trails are not in the current Region.
CLOUDWATCH_ALARM_ACTIONS_NOT_VALID	FAILED	No valid alarm actions are present.
CLOUDWATCH_ALARMS_NOT_PRESENT	FAILED	CloudWatch alarms do not exist in the account.
CONFIG_ACCESS_DENIED	NOT_AVAILABLE AWS Config status is ConfigError	AWS Config access denied. Verify that AWS Config is enabled and has been granted sufficient permissions.
CONFIG_EVALUATIONS_EMPTY	PASSED	AWS Config evaluated your resources based on the rule. The rule did not apply to the AWS resources in its scope, the specified resources were deleted, or the evaluation results were deleted.
CONFIG_RECORDER_CUSTOM_ROLE	FAILED (for Config.1)	The AWS Config recorder uses a custom IAM role instead of the AWS Config service-linked role, and the includeConfigServiceLinkedRoleCheck custom parameter for Config.1 isn't set to false.
CONFIG_RECORDER_DISABLED	FAILED (for Config.1)	AWS Config isn't enabled with the configuration recorder turned on.

Reason code	Compliance status	Description
CONFIG_RECORDER_MISSING_REQUIRED_RESOURCE_TYPES	FAILED (for Config.1)	AWS Config isn't recording all resource types that correspond to enabled Security Hub CSPM controls. Turn on recording for the following resources: *Resources that aren't being recorded*.
CONFIG_RETURNS_NOT_APPLICABLE	NOT_AVAILABLE	The compliance status is NOT_AVAILABLE because AWS Config returned a status of **Not Applicable**. AWS Config does not provide the reason for the status. Here are some possible reasons for the **Not Applicable** status: • The resource was removed from the scope of the AWS Config rule. • The AWS Config rule was deleted. • The resource was deleted. • The AWS Config rule logic can produce a **Not Applicable** status.

Reason code	Compliance status	Description
CONFIG_RULE_EVALUATION_ERROR	NOT_AVAILABLE AWS Config status is ConfigError	This reason code is used for several different types of evaluation errors. The description provides the specific reason information. The type of error can be one of the following: • An inability to perform the evaluation because of a lack of permissions. The description provides the specific permission that is missing. • A missing or invalid value for a parameter. The description provides the parameter and the requirements for the parameter value. • An error reading from an S3 bucket. The description identifies the bucket and provides the specific error. • A missing AWS subscription. • A general timeout on the evaluation. • A suspended account.
CONFIG_RULE_NOT_FOUND	NOT_AVAILABLE AWS Config status is ConfigError	The AWS Config rule is in the process of being created.

Reason code	Compliance status	Description
INTERNAL_SERVICE_ERROR	NOT_AVAILABLE	An unknown error occurred.
LAMBDA_CUSTOM_RUNTIME_DETAILS_NOT_AVAILABLE	FAILED	Security Hub CSPM is unable to perform a check against a custom Lambda runtime.
S3_BUCKET_CROSS_ACCOUNT_CROSS_REGION	WARNING	The finding is in a WARNING state because the S3 bucket that is associated with this rule is in a different Region or account. This rule does not support cross-Region or cross-account checks. It is recommended that you disable this control in this Region or account. Only run it in the Region or account where the resource is located.
SNS_SUBSCRIPTION_NOT_PRESENT	FAILED	The CloudWatch Logs metric filters do not have a valid Amazon SNS subscription.

Reason code	Compliance status	Description
SNS_TOPIC_CROSS_ACCOUNT	WARNING	The finding is in a WARNING state. The SNS topic associated with this rule is owned by a different account. The current account cannot obtain the subscription information. The account that owns the SNS topic must grant to the current account the sns:ListSubscriptionsByTopic permission for the SNS topic.
SNS_TOPIC_CROSS_ACCOUNT_CROSS_REGION	WARNING	The finding is in a WARNING state because the SNS topic that is associated with this rule is in a different Region or account. This rule does not support cross-Region or cross-account checks. It is recommended that you disable this control in this Region or account. Only run it in the Region or account where the resource is located.
SNS_TOPIC_INVALID	FAILED	The SNS topic associated with this rule is invalid.
THROTTLING_ERROR	NOT_AVAILABLE	The relevant API operation exceeded the allowed rate.

ProductFields details for control findings

In findings generated by security checks for controls, the ProductFields attribute in the AWS Security Finding Format (ASFF) can include the following fields.

`ArchivalReasons:0/Description`

Describes why Security Hub CSPM archived a finding.

For example, Security Hub CSPM archives existing findings when you disable a control or standard, or you enable or disable consolidated control findings.

`ArchivalReasons:0/ReasonCode`

Specifies why Security Hub CSPM archived a finding.

For example, Security Hub CSPM archives existing findings when you disable a control or standard, or you enable or disable consolidated control findings.

`PreviousComplianceStatus`

The previous compliance status (`Compliance.Status`) of the resource for the specified control, as of the most recent update to the finding. If the compliance status of the resource didn't change during the most recent update, this value is the same as the value for the `Compliance.Status` field of the finding. For a list of possible values, see Evaluating compliance status and control status.

`StandardsGuideArn` or `StandardsArn`

The ARN of the standard associated with the control.

For the CIS AWS Foundations Benchmark standard, the field is `StandardsGuideArn`. For the PCI DSS and AWS Foundational Security Best Practices standards, the field is `StandardsArn`.

These fields are removed in favor of `Compliance.AssociatedStandards` if you enable consolidated control findings.

`StandardsGuideSubscriptionArn` or `StandardsSubscriptionArn`

The ARN of the account's subscription to the standard.

For the CIS AWS Foundations Benchmark standard, the field is `StandardsGuideSubscriptionArn`. For the PCI DSS and AWS Foundational Security Best Practices standards, the field is `StandardsSubscriptionArn`.

These fields are removed if you enable consolidated control findings.

`RuleId` or `ControlId`

The identifier for the control.

For version 1.2.0 of the CIS AWS Foundations Benchmark standard, the field is `RuleId`. For other standards, including subsequent versions of the CIS AWS Foundations Benchmark standard, the field is `ControlId`.

These fields are removed in favor of `Compliance.SecurityControlId` if you enable consolidated control findings.

`RecommendationUrl`

The URL for remediation information for the control. This field is removed in favor of `Remediation.Recommendation.Url` if you enable consolidated control findings.

`RelatedAWSResources:0/name`

The name of the resource associated with the finding.

`RelatedAWSResource:0/type`

The type of resource associated with the control.

`StandardsControlArn`

The ARN of the control. This field is removed if you enable consolidated control findings.

`aws/securityhub/ProductName`

For control findings, the product name is `Security Hub`.

`aws/securityhub/CompanyName`

For control findings, the company name is `AWS`.

`aws/securityhub/annotation`

A description of the issue uncovered by the control.

`aws/securityhub/FindingId`

The identifier for the finding.

This field doesn't reference a standard if you enable consolidated control findings.

Severity levels for control findings

The severity assigned to a Security Hub CSPM control indicates the importance of the control. The severity of a control determines the severity label assigned to the control findings.

Severity criteria

The severity of a control is determined based on an assessment of the following criteria:

- **How difficult is it for a threat actor to take advantage of the configuration weakness associated with the control?** The difficulty is determined by the amount of sophistication or complexity that is required to use the weakness to carry out a threat scenario.
- **How likely is it that the weakness will lead to a compromise of your AWS accounts or resources?** A compromise of your AWS accounts or resources means that confidentiality, integrity, or availability of your data or AWS infrastructure is damaged in some way. The likelihood of compromise indicates how likely it is that the threat scenario will result in a disruption or breach of your AWS services or resources.

As an example, consider the following configuration weaknesses:

- User access keys are not rotated every 90 days.
- IAM root user key exists.

Both weaknesses are equally difficult for an adversary to take advantage of. In both cases, the adversary can use credential theft or some other method to acquire a user key. They can then use it to access your resources in an unauthorized way.

However, the likelihood of a compromise is much higher if the threat actor acquires the root user access key because this gives them greater access. As a result, the root user key weakness has a higher severity.

The severity does not take into account the criticality of the underlying resource. Criticality is the level of importance of the resources that are associated with the finding. For example, a resource that is associated with a mission critical application is more critical than one that is associated with non-production testing. To capture resource criticality information, use the `Criticality` field of the AWS Security Finding Format (ASFF).

The following table maps the difficulty to exploit and the likelihood of compromise to the security labels.

	Compromise highly likely	Compromise likely	Compromise unlikely	Compromise highly unlikely
Very easy to exploit	Critical	Critical	High	Medium
Somewhat easy to exploit	Critical	High	Medium	Medium
Somewhat difficult to exploit	High	Medium	Medium	Low
Very difficult to exploit	Medium	Medium	Low	Low

Severity definitions

The severity labels are defined as follows.

Critical – The issue should be remediated immediately to avoid it escalating.

For example, an open S3 bucket is considered a critical severity finding. Because so many threat actors scan for open S3 buckets, data in exposed S3 buckets is likely to be discovered and accessed by others.

In general, resources that are publicly accessible are considered critical security issues. You should treat critical findings with the utmost urgency. You also should consider the criticality of the resource.

High – The issue must be addressed as a near-term priority.

For example, if a default VPC security group is open to inbound and outbound traffic, it is considered high severity. It is somewhat easy for a threat actor to compromise a VPC using this method. It is also likely that the threat actor will be able to disrupt or exfiltrate resources once they are in the VPC.

Security Hub CSPM recommends that you treat a high severity finding as a near-term priority. You should take immediate remediation steps. You also should consider the criticality of the resource.

Medium – The issue should be addressed as a mid-term priority.

For example, lack of encryption for data in transit is considered a medium severity finding. It requires a sophisticated man-in-the-middle attack to take advantage of this weakness. In other words, it is somewhat difficult. It is likely that some data will be compromised if the threat scenario is successful.

Security Hub CSPM recommends that you investigate the implicated resource at your earliest convenience. You also should consider the criticality of the resource.

Low – The issue does not require action on its own.

For example, failure to collect forensics information is considered low severity. This control can help to prevent future compromises, but the absence of forensics does not lead directly to a compromise.

You do not need to take immediate action on low severity findings, but they can provide context when you correlate them with other issues.

Informational – No configuration weakness was found.

In other words, the status is `PASSED`, `WARNING`, or `NOT AVAILABLE`.

There is no recommended action. Informational findings help customers to demonstrate that they are in a compliant state.

Evaluating compliance status and control status

The `Compliance.Status` field of the AWS Security Finding Format describes the result of a control finding. AWS Security Hub CSPM uses the compliance status of control findings to determine an overall control status. The control status is displayed on the details page of a control on the Security Hub CSPM console.

Evaluating the compliance status of Security Hub CSPM findings

The compliance status for each finding is assigned one of the following values:

- `PASSED` – Indicates that the control passed the security check for the finding. This automatically sets the Security Hub CSPM `Workflow.Status` to `RESOLVED`.
- `FAILED` – Indicates that the control didn't pass the security check for the finding.

- WARNING – Indicates that Security Hub CSPM can't determine whether the resource is in a PASSED or FAILED state. For example, AWS Config resource recording isn't turned on for the corresponding resource type.
- NOT_AVAILABLE – Indicates that the check can't be completed because a server failed, the resource was deleted, or the result of the AWS Config evaluation was NOT_APPLICABLE. If the AWS Config evaluation result was NOT_APPLICABLE, Security Hub CSPM automatically archives the finding.

If the compliance status for a finding changes from PASSED to FAILED, WARNING, or NOT_AVAILABLE, and Workflow.Status was either NOTIFIED or RESOLVED, Security Hub CSPM automatically changes Workflow.Status to NEW.

If you don't have resources corresponding to a control, Security Hub CSPM produces a PASSED finding at the account level. If you have a resource corresponding to a control but then delete the resource, Security Hub CSPM creates a NOT_AVAILABLE finding and archives it immediately. After 18 hours, you receive a PASSED finding because you no longer have resources corresponding to the control.

Deriving control status from compliance status

Security Hub CSPM derives an overall control status from the compliance status of the control findings. When determining control status, Security Hub CSPM ignores findings that have a RecordState of ARCHIVED and findings that have a Workflow.Status of SUPPRESSED.

Control status is assigned one of the following values:

- **Passed** – Indicates that all findings have a compliance status of PASSED.
- **Failed** – Indicates that at least one finding has a compliance status of FAILED.
- **Unknown** – Indicates that at least one finding has a compliance status of WARNING or NOT_AVAILABLE. No findings have a compliance status of FAILED.
- **No data** – Indicates that there are no findings for the control. For example, a newly enabled control has this status until Security Hub CSPM starts to generate findings for it. A control also has this status if all of its findings are SUPPRESSED or it's unavailable in the current AWS Region.
- **Disabled** – Indicates that the control is disabled in the current account and Region. No security checks are currently being performed for this control in the current account and Region. However, the findings of a disabled control may have a value for compliance status for up to 24 hours after disablement.

For an administrator account, control status reflects the control status for the administrator account and the member accounts. Specifically, the overall status of a control appears as **Failed** if the control has one or more failed findings in the administrator account or any of the member accounts. If you have set an aggregation Region, the control status in the aggregation Region reflects the control status in the aggregation Region and the linked Regions. Specifically, the overall status of a control appears as **Failed** if the control has one or more failed findings in the aggregation Region or any of the linked Regions.

Security Hub CSPM typically generates the initial control status within 30 minutes after your first visit to the **Summary** page or the **Security standards** page on the Security Hub CSPM console. You must have AWS Config resource recording configured for the control status to appear. After control statuses are generated for the first time, Security Hub CSPM updates control statuses every 24 hours based on the findings from the previous 24 hours. A timestamp on the control details page indicates when control status was last updated.

> **Note**
>
> After enabling a control for first time, it can take up to 24 hours for control statuses to be generated in the China Regions and the AWS GovCloud (US) Region.

Calculating security scores

On the AWS Security Hub CSPM console, the **Summary** page and the **Controls** page display a summary security score across all of your enabled standards. On the **Security standards** page, Security Hub CSPM also displays a security score from 0–100 percent for each enabled standard.

When you first enable Security Hub CSPM, Security Hub CSPM calculates the summary security score and standard security scores within 30 minutes of your first visit to the **Summary** or **Security standards** page on the console. Scores are generated only for standards that are enabled when you visit those pages on the console. In addition, AWS Config resource recording must be configured for the scores to appear. The summary security score is the average of the standard security scores. To review a list of standards that are currently enabled, you can use the GetEnabledStandards operation of the Security Hub CSPM API.

After first-time score generation, Security Hub CSPM updates security scores every 24 hours. Security Hub CSPM displays a timestamp to indicate when a security score was last updated. Note that it can take up to 24 hours for first-time security scores to be generated in the China Regions and AWS GovCloud (US) Regions.

If you turn on consolidated control findings, it can take up to 24 hours for your security scores to update. In addition, enabling a new aggregation Region or updating linked Regions resets existing security scores. It can take up to 24 hours for Security Hub CSPM to generate new security scores that include data from the updated Regions.

Method of calculating security scores

Security scores represent the proportion of **Passed** controls to enabled controls. The score is displayed as a percentage rounded up or down to the nearest whole number.

Security Hub CSPM calculates a summary security score across all of your enabled standards. Security Hub CSPM also calculates a security score for each enabled standard. For purposes of score calculation, enabled controls include controls with a status of **Passed**, **Failed**, and **Unknown**. Controls with a status of **No data** are excluded from the score calculation.

Security Hub CSPM ignores archived and suppressed findings when calculating control status. This can impact security scores. For example, if you suppress all failed findings for a control, its status becomes **Passed**, which can in turn improve your security scores. For more information about control status, see Evaluating compliance status and control status.

Scoring example:

Standard	Passed controls	Failed controls	Unknown controls	Standard score
AWS Foundational Security Best Practices v1.0.0	168	22	0	88%
CIS AWS Foundations Benchmark v1.4.0	8	29	0	22%
CIS AWS Foundations Benchmark v1.2.0	6	35	0	15%

Standard	Passed controls	Failed controls	Unknown controls	Standard score
NIST Special Publication 800-53 Revision 5	159	56	0	74%
PCI DSS v3.2.1	28	17	0	62%

When calculating the summary security score, Security Hub CSPM counts each control only once across standards. For example, if you have enabled a control that applies to three enabled standards, it only counts as one enabled control for scoring purposes.

In this example, although the total number of enabled controls across enabled standards is 528, Security Hub CSPM counts each unique control only once for scoring purposes. The number of unique enabled controls is likely lower than 528. If we assume the number of unique enabled controls is 515, and the number of unique passed controls is 357, the summary score is 69%. This score is calculated by dividing the number of unique passed controls by the number of unique enabled controls.

You might have a summary score that differs from the standard security score, even if you've enabled only one standard in your account in the current Region. This can occur if you're signed in to an administrator account and member accounts have additional standards or different standards enabled. This can also occur if you're viewing the score from the aggregation Region and additional standards or different standards are enabled in linked Regions.

Security scores for administrator accounts

If you're signed in to an administrator account, the summary security score and standard scores account for control statuses in the administrator account and all of the member accounts.

If the status of a control is **Failed** in even one member account, its status is **Failed** in the administrator account and impacts the administrator account scores.

If you're signed in to an administrator account and are viewing scores in an aggregation Region, security scores account for control statuses in all member accounts *and* all linked Regions.

Security scores if you have set an aggregation Region

If you have set an aggregation AWS Region, the summary security score and standard scores account for control statuses in all
linked Regions.

If the status of a control is **Failed** in even one linked Region, its status is **Failed** in the aggregation Region and impacts the aggregation Region scores.

If you're signed in to an administrator account and are viewing scores in an aggregation Region, security scores account for control statuses in all member accounts *and* all linked Regions.

Control categories in Security Hub CSPM

Each control is assigned a category. The category for a control reflects the security function that the control applies to.

The category value contains the category, the subcategory within the category, and, optionally, a classifier within the subcategory. For example:

- Identify > Inventory
- Protect > Data protection > Encryption of data in transit

Here are the descriptions of the available categories, subcategories, and classifiers.

Identify

Develop the organizational understanding to manage cybersecurity risk to systems, assets, data, and capabilities.

Inventory

Has the service implemented the correct resource tagging strategies? Do the tagging strategies include the resource owner?

What resources does the service use? Are they approved resources for this service?

Do you have visibility into the approved inventory? For example, do you use services such as Amazon EC2 Systems Manager and Service Catalog?

Logging

Have you securely enabled all relevant logging for the service? Examples of log files include the following:

- Amazon VPC Flow Logs
- Elastic Load Balancing access logs
- Amazon CloudFront logs
- Amazon CloudWatch Logs
- Amazon Relational Database Service logging
- Amazon OpenSearch Service slow index logs
- X-Ray tracing
- AWS Directory Service logs
- AWS Config items
- Snapshots

Protect

Develop and implement the appropriate safeguards to ensure delivery of critical infrastructure services and secure coding practices.

Secure access management

Does the service use least privilege practices in its IAM or resource policies?

Are passwords and secrets sufficiently complex? Are they rotated appropriately?

Does the service use multi-factor authentication (MFA)?

Does the service avoid the root user?

Do resource-based policies allow public access?

Secure network configuration

Does the service avoid public and insecure remote network access?

Does the service use VPCs properly? For example, are jobs required to run in VPCs?

Does the service properly segment and isolate sensitive resources?

Data protection

Encryption of data at rest – Does the service encrypt data at rest?

Encryption of data in transit – Does the service encrypt data in transit?

Data integrity – Does the service validate data for integrity?

Data deletion protection – Does the service protect data from accidental deletion?

Data management / usage – Do you use services such as Amazon Macie to track the location of your sensitive data?

API protection

Does the service use AWS PrivateLink to protect the service API operations?

Protective services

Are the correct protective services in place? Do they provide the correct amount of coverage?

Protective services help you deflect attacks and compromises that are directed at the service. Examples of protective services in AWS include AWS Control Tower, AWS WAF, AWS Shield Advanced, Vanta, Secrets Manager, IAM Access Analyzer, and AWS Resource Access Manager.

Secure development

Do you use secure coding practices?

Do you avoid vulnerabilities such as the Open Web Application Security Project (OWASP) Top Ten?

Detect

Develop and implement the appropriate activities to identify the occurrence of a cybersecurity event.

Detection services

Are the correct detection services in place?

Do they provide the correct amount of coverage?

Examples of AWS detection services include Amazon GuardDuty, AWS Security Hub CSPM, Amazon Inspector, Amazon Detective, Amazon CloudWatch Alarms, AWS IoT Device Defender, and AWS Trusted Advisor.

Respond

Develop and implement the appropriate activities to take action regarding a detected cybersecurity event.

Response actions

Do you respond to security events swiftly?

Do you have any active critical or high severity findings?

Forensics

Can you securely acquire forensic data for the service? For example, do you acquire Amazon EBS snapshots associated with true positive findings?

Have you set up a forensic account?

Recover

Develop and implement the appropriate activities to maintain plans for resilience and to restore any capabilities or services that were impaired due to a cybersecurity event.

Resilience

Does the service configuration support graceful failovers, elastic scaling, and high availability?

Have you established backups?

Reviewing the details of controls in Security Hub CSPM

Selecting a control on the **Controls** page or standard details page of the Security Hub CSPM console takes you to a page of control details.

The top of the control details page indicates the control status. The control status summarizes the performance of a control based on the compliance status of the control findings. Security Hub

CSPM typically generates the initial control status within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub CSPM console. Statuses are only available for controls that are enabled when you visit those pages.

The control details page also provides a breakdown of the compliance status of the control findings for the past 24 hours. For more information about control status and compliance status, see Evaluating compliance status and control status.

AWS Config resource recording must be configured for the control status to appear. After control statuses are generated for the first time, Security Hub CSPM updates the control status every 24 hours based on findings from the previous 24 hours.

Administrator accounts see an aggregated control status across the administrator account and member accounts. If you have set an aggregation Region, the control status includes findings across all linked Regions. For more information about control status, see the section called "Compliance status and control status".

You can also enable or disable the control from the control details page.

> **Note**
>
> It can take up to 24 hours after enabling a control for first-time control statuses to be generated in the China Regions and AWS GovCloud (US) Regions.

The **Standards and Requirements** tab lists the standards that a control can be enabled for and the requirements related to the control from different compliance frameworks.

The **Checks** tab lists active findings for the control for the past 24 hours. Control findings are generated and updated when Security Hub CSPM runs security checks for the control. The list on this tab doesn't include archived findings.

For each finding, the list provides access to finding details such as the compliance status and related resource. You can also set the workflow status of each finding and send findings to custom actions. For more information, see Reviewing and managing control findings.

Viewing details for a control

Choose your preferred access method, and follow these steps to review details for a control. Details apply to the current account and Region and include the following:

- The title and description of the control.
- A link to remediation guidance for failed control findings.
- The severity of the control.
- The status of the control.

On the console, you can also review a list of recent findings for the control. To do this programmatically, you can use the GetFindings operation of the Security Hub CSPM API.

Security Hub CSPM console

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Choose **Controls** in the navigation pane.
3. Select a control.

Security Hub CSPM API

1. Run `ListSecurityControlDefinitions`, and provide one or more standard ARNs to get a list of control IDs for that standard. To obtain standard ARNs, run `DescribeStandards`. If you don't provide a standard ARN, this API returns all Security Hub CSPM control IDs. This API returns standard-agnostic security control IDs, not the standard-based control IDs that existed prior to these feature releases.

   Example request:

   ```
   {
       "StandardsArn": "arn:aws:securityhub:::standards/aws-foundational-security-best-practices/v/1.0.0"
   }
   ```

2. Run `BatchGetSecurityControls` to get details about one or more controls in the current AWS account and AWS Region.

   Example request:

   ```
   {
       "SecurityControlIds": ["Config.1", "IAM.1"]
   ```

```
}
```

AWS CLI

1. Run the `list-security-control-definitions` command, and provide one or more standard ARNs to get a list of control IDs. To obtain standard ARNs, run the `describe-standards` command. If you don't provide a standard ARN, this command returns all Security Hub CSPM control IDs. This command returns standard-agnostic security control IDs, not the standard-based control IDs that existed prior to these feature releases.

```
aws securityhub --region us-east-1 list-security-control-definitions --
standards-arn "arn:aws:securityhub:us-east-1::standards/aws-foundational-
security-best-practices/v/1.0.0"
```

2. Run the `batch-get-security-controls` command to get details about one or more controls in the current AWS account and AWS Region.

```
aws securityhub --region us-east-1 batch-get-security-controls --security-
control-ids '["Config.1", "IAM.1"]'
```

Filtering and sorting controls in Security Hub CSPM

On the AWS Security Hub CSPM console, you can use the **Controls** page to review a table of the controls that are available in the current AWS Region. The exception is an aggregation Region. If you configured an aggregation Region and sign in to that Region, the console shows controls that are available in the aggregation Region or one or more linked Regions.

To focus on a specific subset of controls, you can sort and filter the table of controls. The **Filter by** options next to the table can help you quickly focus on these specific subsets:

- All enabled controls, which are controls that are enabled in at least one enabled standard.
- All disabled controls, which are controls that are disabled in all standards.
- All enabled controls that have a specific control status, such as **Failed**. The **No data** option displays only those controls that don't currently have findings. For information about control status, see Evaluating compliance status and control status.

In addition to the **Filter by** options, you can filter the table by entering filter criteria in the **Filter controls** box above the table. For example, you can filter by control ID or severity.

By default, controls with a **Failed** status are listed first, in descending order by severity. You can change the sort order by choosing a different column heading.

> **Tip**
>
> If you have automated workflows based on control findings, we recommend using the `SecurityControlId` or `SecurityControlArn` ASFF fields as filters, rather than the `Title` or `Description` fields. The latter fields can change occasionally, whereas control ID and ARN are static identifiers.

If you're signed in to a Security Hub CSPM administrator account, **Enabled** controls include controls that are enabled in at least one member account. If you configured an aggregation Region, **Enabled** controls include controls that are enabled in at least one linked Region.

If you select the option next to an enabled a control, a panel appears and displays the standards in which the control is currently enabled. You can also see the standards in which the control is currently disabled. From this panel, you can disable a control in all standards. For more information, see Disabling controls in Security Hub CSPM. For administrator accounts, the information in the panel reflects settings for all of your member accounts.

To retrieve a list of controls programmatically, you can use the ListSecurityControlDefinitions operation of the Security Hub CSPM API. To retrieve the details of individual controls, use the BatchGetSecurityControls operation.

Understanding control parameters in Security Hub CSPM

Some controls in AWS Security Hub CSPM use parameters that affect how the control is evaluated. Typically, such controls are evaluated against the default parameter values that Security Hub CSPM defines. However, for a subset of these controls, you can modify the parameter values. When you modify a control parameter value, Security Hub CSPM starts evaluating the control against the value that you specify. If the resource underlying the control satisfies the custom value, Security Hub CSPM generates a `PASSED` finding. If the resource doesn't satisfy the custom value, Security Hub CSPM generates a `FAILED` finding.

By customizing control parameters, you can refine the security best practices recommended and monitored by Security Hub CSPM to align with your business requirements and security expectations. Instead of suppressing findings for a control, you can customize one or more of its parameters to get findings that suit your security needs.

Here are some sample use cases for modifying control parameters and setting custom values:

- **[CloudWatch.16] – CloudWatch log groups should be retained for a specified time period**

  You can specify the retention time period.
- **[IAM.7] – Password policies for IAM users should have strong configurations**

  You can specify parameters related to password strength.
- **[EC2.18] – Security groups should only allow unrestricted incoming traffic for authorized ports**

  You can specify which ports are authorized to permit unrestricted incoming traffic.
- **[Lambda.5] – VPC Lambda functions should operate in multiple Availability Zones**

  You can specify the minimum number of Availability Zones that produces a passed finding.

This section covers things to consider when you modify control parameters.

Effect of modifying control parameter values

When you change a parameter value, you also trigger a new security check that evaluates the control based on the new value. Security Hub CSPM then generates new control findings based on the new value. During periodic updates to control findings, Security Hub CSPM also uses the new parameter value. If you change parameter values for a control, but haven't enabled any standards that include the control, Security Hub CSPM doesn't conduct any security checks using the new values. You have to enable at least one relevant standard for Security Hub CSPM to evaluate the control based on the new parameter value.

A control can have one or more customizable parameters. Possible data types for each control parameter include the following:

- Boolean
- Double
- Enum

- EnumList
- Integer
- IntegerList
- String
- StringList

Custom parameter values apply across your enabled standards. You can't customize the parameters for a control that's not supported in your current Region. For a list of Regional limits for individual controls, see Regional limits on Security Hub CSPM controls.

For some controls, acceptable parameter values must fall into a specified range to be valid. In these cases, Security Hub CSPM provides the acceptable range.

Security Hub CSPM chooses default parameter values and might occasionally update them. After you customize a control parameter, its value continues to be the value that you specified for the parameter unless your change it. That is to say, the parameter stops tracking updates to the default Security Hub CSPM value, even if the custom value of the parameter matches the current, default value defined by Security Hub CSPM. Here's an example for the control **[ACM.1] – Imported and ACM-issued certificates should be renewed after a specified time period**:

```
{
    "SecurityControlId": "ACM.1",
    "Parameters": {
        "daysToExpiration": {
            "ValueType": "CUSTOM",
            "Value": {
                "Integer": 30
            }
        }
    }
}
```

In the preceding example, the `daysToExpiration` parameter has a custom value of `30`. The current default value for this parameter is also `30`. If Security Hub CSPM changes the default value to `14`, the parameter in this example won't track that change. It will retain a value of `30`.

If you want to track updates to the default Security Hub CSPM value for a parameter, set the `ValueType` field to `DEFAULT` instead of `CUSTOM`. For more information, see Reverting to default control parameters in a single account and Region.

Controls that support custom parameters

For a list of security controls that support custom parameters, see the **Controls** page of the Security Hub CSPM console or the Control reference for Security Hub CSPM. To retrieve this list programmatically, you can use the ListSecurityControlDefinitions operation. In the response, the `CustomizableProperties` object indicates which controls support customizable parameters.

Reviewing current control parameter values

It can be helpful to know the current value of a control parameter before you modify it.

You can review the current values for individual control parameters in your account. If you use central configuration, the delegated AWS Security Hub CSPM administrator can also review parameter values that are specified in a configuration policy.

Choose your preferred method, and follow the steps to review current control parameter values.

Security Hub CSPM console

To review current control parameter values (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Controls**. Choose a control.
3. Choose the **Parameters** tab. This tab shows the current parameter values for the control.

Security Hub CSPM API

To review current control parameter values (API)

Invoke the BatchGetSecurityControls API, and provide one or more security control IDs or ARNs. The `Parameters` object in the response shows the current parameter values for the specified controls.

For example, the following AWS CLI command shows the current parameter values for `APIGatway.1`, `CloudWatch.15`, and `IAM.7`. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub batch-get-security-controls \
```

```
--region us-east-1 \
--security-control-ids '["APIGateway.1", "CloudWatch.15", "IAM.7"]'
```

Choose your preferred method to view the current parameter values in a central configuration policy.

Security Hub CSPM console

To review current control parameter values in a configuration policy (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.
2. In the navigation pane, choose **Settings** and **Configuration**.
3. On the **Policies** tab, select the configuration policy, and then choose **View details**. The policy details then appear, including current parameter values.

Security Hub CSPM API

To review current control parameter values in a configuration policy (API)

1. Invoke the GetConfigurationPolicy API from the delegated administrator account in the home Region.
2. Provide the ARN or ID of the configuration policy whose details you want to see. The response includes current parameter values.

For example, the following AWS CLI command retrieves the current control parameter values in the specified configuration policy. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub get-configuration-policy \
--region us-east-1 \
--identifier "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Control findings also include the current values of control parameters. In the AWS Security Finding Format (ASFF), these values appear in the `Parameters` field of the `Compliance` object. To review findings on the Security Hub CSPM console, choose **Findings** in the navigation pane. To review findings programmatically, use the GetFindings operation of the Security Hub CSPM API.

Customizing control parameter values

The instructions for customizing control parameters vary based on whether you use central configuration in AWS Security Hub CSPM. Central configuration is a feature that the delegated Security Hub CSPM administrator can use to configure Security Hub CSPM capabilities across AWS Regions, accounts, and organizational units (OUs).

If your organization uses central configuration, the delegated administrator can create configuration policies that include custom control parameters. These policies can be associated with centrally managed member accounts and OUs, and they take effect in your home Region and all linked Regions. The delegated administrator can also designate one or more accounts as self-managed, which allows the account owner to configure its own parameters separately in each Region. If your organization doesn't use central configuration, you must customize control parameters separately in each account and Region.

We recommend using central configuration because it allows you to align control parameter values across different parts of your organization. For example, all of your test accounts might use certain parameter values, and all production accounts might use different values.

Customizing control parameters in multiple accounts and Regions

If you're the delegated Security Hub CSPM administrator for an organization that uses central configuration, choose your preferred method, and follow the steps to customize control parameters across multiple accounts and Regions.

Security Hub CSPM console

To customize control parameter values in multiple accounts and Regions (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Ensure that you're signed in to the home Region.

2. In the navigation pane, choose **Settings** and **Configuration**.
3. Choose the **Policies** tab.
4. To create a new configuration policy that includes custom parameters, choose **Create policy**. To specify custom parameters in an existing configuration policy, select the policy, and then choose **Edit**.

   To create a new configuration policy with custom control parameter values

   1. In the **Custom policy** section, choose the security standards and controls that you want to enable.
   2. Select **Customize control parameters**.
   3. Select a control, and then specify custom values for one or more parameters.
   4. To customize parameters for more controls, choose **Customize additional control**.
   5. In the **Accounts** section, select the accounts or OUs that you want to apply the policy to.
   6. Choose **Next**.
   7. Choose **Create policy and apply**. In your home Region and all linked Regions, this action overrides the existing configuration settings of accounts and OUs that are associated with this configuration policy. Accounts and OUs can be associated with a configuration policy through direct application or inheritance from a parent.

   To customize control parameter values in an existing configuration policy

   1. In the **Controls** section, under **Custom policy**, specify the new custom parameter values that you want.
   2. If this is your first time customizing control parameters in this policy, select **Customize control parameters**, and then select a control to customize. To customize parameters for more controls, choose **Customize additional control**.
   3. In the **Accounts** section, verify the accounts or OUs that you want to apply the policy to.
   4. Choose **Next**.
   5. Review your changes, and verify that they're correct. When you finish, choose **Save policy and apply**. In your home Region and all linked Regions, this action overrides the existing configuration settings of accounts and OUs that are associated with this configuration policy. Accounts and OUs can be associated with a configuration policy through direct application or inheritance from a parent.

To customize control parameter values in multiple accounts and Regions (API)

To create a new configuration policy with custom control parameter values

1. Invoke the CreateConfigurationPolicy API from the delegated administrator account in the home Region.
2. For the `SecurityControlCustomParameters` object, provide the identifier of each control that you want to customize.
3. For the `Parameters` object, provide the name of each parameter that you want to customize. For each parameter that you customize, provide `CUSTOM` for `ValueType`. For `Value`, provide the data type of the parameter and the custom value. The `Value` field can't be empty when `ValueType` is `CUSTOM`. If your request omits a parameter that the control supports, that parameter retains its current value. You can find supported parameters, data types, and valid values for a control by invoking the GetSecurityControlDefinition API.

To customize control parameter values in an existing configuration policy

1. Invoke the UpdateConfigurationPolicy API from the delegated administrator account in the home Region.
2. For the `Identifier` field, provide the Amazon Resource Name (ARN) or ID of the configuration policy that you want to update.
3. For the `SecurityControlCustomParameters` object, provide the identifier of each control that you want to customize.
4. For the `Parameters` object, provide the name of each parameter that you want to customize. For each parameter that you customize, provide `CUSTOM` for `ValueType`. For `Value`, provide the data type of the parameter and the custom value. If your request omits a parameter that the control supports, that parameter retains its current value. You can find supported parameters, data types, and valid values for a control by invoking the GetSecurityControlDefinition API.

For example, the following AWS CLI command creates a new configuration policy with a custom value for the `daysToExpiration` parameter of `ACM.1`. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub create-configuration-policy \
--region us-east-1 \
--name "SampleConfigurationPolicy" \
--description "Configuration policy for production accounts" \
--configuration-policy '{"SecurityHub": {"ServiceEnabled": true,
 "EnabledStandardIdentifiers": ["arn:aws:securityhub:us-east-1::standards/aws-
foundational-security-best-practices/v/1.0.0","arn:aws:securityhub:::ruleset/
cis-aws-foundations-benchmark/v/1.2.0"],"SecurityControlsConfiguration":
{"DisabledSecurityControlIdentifiers": ["CloudTrail.2"],
 "SecurityControlCustomParameters": [{"SecurityControlId": "ACM.1", "Parameters":
 {"daysToExpiration": {"ValueType": "CUSTOM", "Value": "Integer": 15}}}]}}}'
```

Customizing control parameters in a single account and Region

If you don't use central configuration or have a self-managed account, you can customize control parameters for your account in one Region at a time only.

Choose your preferred method, and follow the steps to customize control parameters. Your changes apply only to your account in the current Region. To customize the control parameters in additional Regions, repeat the following steps in each additional account and Region in which you want to customize parameters. The same control can use different parameter values in different Regions.

Security Hub CSPM console

To customize control parameter values in one account and Region (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Controls**. In the table, choose a control that supports custom parameters and you want to change the parameters for. The **Custom parameters** column indicates which controls support custom parameters.
3. On the details page for the control, choose the **Parameters** tab, and then choose **Edit**.
4. Specify the parameter values that you want.
5. Optionally, in the **Reason for change** section, select a reason for customizing the parameters.
6. Choose **Save**.

Security Hub CSPM API

To customize control parameter values in one account and Region (API)

1. Invoke the UpdateSecurityControl API.
2. For `SecurityControlId`, provide the ID of the control that you want to customize.
3. For the `Parameters` object, provide the name of each parameter that you want to customize. For each parameter that you customize, provide `CUSTOM` for `ValueType`. For `Value`, provide the data type of the parameter and the custom value. If your request omits a parameter that the control supports, that parameter retains its current value. You can find supported parameters, data types, and valid values for a control by invoking the GetSecurityControlDefinition API.
4. Optionally, for `LastUpdateReason`, provide a reason for customizing the control parameters.

For example, the following AWS CLI command defines a custom value for the `daysToExpiration` parameter of `ACM.1`. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub update-security-control \
--region us-east-1 \
--security-control-id ACM.1 \
--parameters '{"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer":
 15}}}' \
--last-update-reason "Internal compliance requirement"
```

Reverting to default control parameter values

A control parameter can have a default value that AWS Security Hub CSPM defines. Occasionally, Security Hub CSPM updates the default value for a parameter to reflect evolving security best practices. If you haven't specified a custom value for a control parameter, the control automatically tracks those updates and uses the new default value.

You can revert to using default parameter values for a control. The instructions for reversion depend on whether you use central configuration in Security Hub CSPM. Central configuration is

a feature that the delegated Security Hub CSPM administrator can use to configure Security Hub CSPM capabilities across AWS Regions, accounts, and organizational units (OUs).

> **Note**
>
> Not all control parameters have a default Security Hub CSPM value. In such cases, when `ValueType` is set to `DEFAULT`, there isn't a specific default value that Security Hub CSPM uses. Rather, Security Hub CSPM ignores the parameter in the absence of a custom value.

Reverting to default control parameters in multiple accounts and Regions

If you use central configuration, you can revert control parameters for multiple, centrally managed accounts and OUs in the home Region and linked Regions.

Choose your preferred method, and follow the steps to revert to default parameter values across multiple accounts and Regions using central configuration.

Security Hub CSPM console

To revert to default control parameter values in multiple accounts and Regions (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub CSPM administrator account in the home Region.
2. In the navigation pane, choose **Settings** and **Configuration**.
3. Choose the **Policies** tab.
4. Select a policy, and then choose **Edit**.
5. Under **Custom policy**, the **Controls** section shows a list of controls that you specified custom parameters for.
6. Find the control that has one or more parameter values to revert. Then, choose **Remove** to revert to the default values.
7. In the **Accounts** section, verify the accounts or OUs that you want to apply the policy to.
8. Choose **Next**.
9. Review your changes, and verify that they're correct. When you finish, choose **Save policy and apply**. In your home Region and all linked Regions, this action overrides the existing

configuration settings of accounts and OUs that are associated with this configuration policy. Accounts and OUs can be associated with a configuration policy through direct application or inheritance from a parent.

Security Hub CSPM API

To revert to default control parameter values in multiple accounts and Regions (API)

1. Invoke the UpdateConfigurationPolicy API from the delegated administrator account in the home Region.
2. For the `Identifier` field, provide the Amazon Resource Name (ARN) or ID of the policy that you want to update.
3. For the `SecurityControlCustomParameters` object, provide the identifier of each control for which you want to revert one or more parameters.
4. In the `Parameters` object, for each parameter that you want to revert, provide `DEFAULT` for the `ValueType` field. When `ValueType` is set to `DEFAULT`, you don't need to provide a value for the `Value` field. If a value is included in your request, Security Hub CSPM ignores it. If your request omits a parameter that the control supports, that parameter retains its current value.

> **Warning**
>
> If you omit a control object from the `SecurityControlCustomParameters` field, Security Hub CSPM reverts all custom parameters for the control to their default values. A completely empty list for `SecurityControlCustomParameters` reverts custom parameters for all controls to their default values.

For example, the following AWS CLI command reverts the `daysToExpiration` control parameter for `ACM.1` to its default value in the specified configuration policy. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub create-configuration-policy \
--region us-east-1 \
--identifier "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
```

```
--name "TestConfigurationPolicy" \
--description "Updated configuration policy" \
--updated-reason "Revert ACM.1 parameter to default value"
--configuration-policy '{"SecurityHub": {"ServiceEnabled": true,
 "EnabledStandardIdentifiers": ["arn:aws:securityhub:us-east-1::standards/aws-
foundational-security-best-practices/v/1.0.0","arn:aws:securityhub:::ruleset/
cis-aws-foundations-benchmark/v/1.2.0"],"SecurityControlsConfiguration":
{"DisabledSecurityControlIdentifiers": ["CloudTrail.2"],
 "SecurityControlCustomParameters": [{"SecurityControlId": "ACM.1", "Parameters":
 {"daysToExpiration": {"ValueType": "DEFAULT"}}}]}}}'
```

Reverting to default control parameters in a single account and Region

If you don't use central configuration or have a self-managed account, you can revert to using default parameter values for your account in one Region at a time.

Choose your preferred method, and follow the steps to revert to default parameter values for your account in a single Region. To revert to default parameter values in additional Regions, repeat these steps in each additional Region.

> **Note**
>
> If you disable Security Hub CSPM, your custom control parameters are reset. If you enable Security Hub CSPM again in the future, all controls will use default parameter values to start.

Security Hub CSPM console

To revert to default control parameter values in one account and Region (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Controls**. Choose the control that you want to revert to default parameter values.
3. On the `Parameters` tab, choose **Customized** next to a control parameter. Then, choose **Remove customization**. This parameter now uses the default Security Hub CSPM value and tracks future updates to the default value.

4. Repeat the preceding step for each parameter value that you want to revert.

Security Hub CSPM API

To revert to default control parameter values in one account and Region (API)

1. Invoke the UpdateSecurityControl API.
2. For `SecurityControlId`, provide the ARN or ID of the control whose parameters you want to revert.
3. In the `Parameters` object, for each parameter that you want to revert, provide `DEFAULT` for the `ValueType` field. When `ValueType` is set to `DEFAULT`, you don't need to provide a value for the `Value` field. If a value is included in your request, Security Hub CSPM ignores it.
4. Optionally, for `LastUpdateReason`, provide a reason for reverting to default parameter values.

For example, the following AWS CLI command reverts the `daysToExpiration` control parameter for `ACM.1` to its default value. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub update-security-control \
--region us-east-1 \
--security-control-id ACM.1 \
--parameters '{"daysToExpiration": {"ValueType": "DEFAULT"}}' \
--last-update-reason "New internal requirement"
```

Checking the status of control parameter changes

When you attempt to customize a control parameter or revert to the default value, you can validate whether the desired changes were effective. This helps ensure that a control works as you expect and provides the intended security value. If a parameter update is unsuccessful, Security Hub CSPM retains the current value for the parameter.

To verify that a parameter update was successful, you can review the details of the control on the Security Hub CSPM console. On the console, choose **Controls** on the navigation pane. Then, choose a control to display its details. The **Parameters** tab shows the status of the parameter change.

Programmatically, if your request to update a parameter is valid, the value of the `UpdateStatus` field is `UPDATING` in a response to the BatchGetSecurityControls operation. This means that the update was valid, but all findings might not yet include the updated parameter values. When the value of `UpdateState` changes to `READY`, Security Hub CSPM uses the updated control parameter values when running security checks of the control. Findings include the updated parameter values.

The `UpdateSecurityControl` operation returns an `InvalidInputException` response for invalid parameter values. The response provides additional details about the reason for failure. For example, you might have specified a value that's outside the valid range for a parameter. Or, you might have specified a value that doesn't use the correct data type. Submit your request again with valid input.

If an internal failure occurs when you try to update a parameter value, Security Hub CSPM automatically retries if you have AWS Config enabled. For more information, see Considerations before enabling and configuring AWS Config.

Reviewing and managing control findings in Security Hub CSPM

The control details page displays a list of active findings for a control. The list does not include archived findings.

The control details page supports cross-Region aggregation. If you have set an aggregation Region, the control status and list of security checks on the control details page include checks from all linked AWS Regions.

The list provides tools to filter and sort the findings, so that you can focus on more urgent findings first. A finding may include links to resource details in the related service console. For controls that are based on AWS Config rules, you can view details about the rule.

You can also use the AWS Security Hub CSPM API to retrieve a list of findings and finding details.

For more information, see Reviewing finding details and history.

To reflect the current status of your investigation of a control finding, you set the workflow status. For more information, see the section called “Setting the workflow status of findings”.

You can also send selected Security Hub CSPM findings to a custom action in Amazon EventBridge. For more information, see the section called “Sending findings to a custom action”.

Topics

- Filtering and sorting control findings

- Samples of control findings

Filtering and sorting control findings

Selecting a control from the **Controls** page of the AWS Security Hub CSPM console or from the details page of a standard takes you to the control details page.

The control details page shows the title and description of the control, the overall control status, and a breakdown of security checks for the control in the last 24 hours.

Use the **Filter by** options next to the control checks list to quickly focus on findings with a specific workflow status or compliance status.

In addition to the **Filter by** options, you can use the **Add filter** box to filter the checks list by other fields, such as AWS account ID or resource ID.

By default, findings with a compliance status of **PASSED** are listed first. You can change the default sorting by choosing a different option in the column headers.

From the control details page, you can choose **Download** to download the current page of control findings to a .csv file.

If you filter the finding list, then the download only includes the controls that match the filter. If you select specific findings from the list, then the download only includes the selected findings.

For more information about filtering findings, see Filtering findings in Security Hub CSPM.

Samples of control findings

The following samples provide examples of AWS Security Hub CSPM control findings in the AWS Security Finding Format (ASFF). The contents of control findings vary depending on whether you enabled consolidated control findings.

If you enable consolidated control findings, Security Hub CSPM generates a single finding for a control, even if the control applies to multiple enabled standards. If you don't enable this feature, Security Hub CSPM generates a separate control finding for each enabled standard that a control applies to. For example, if you enable two standards and a control applies to both of them, you receive two separate findings for the control, one for each standard. If you enable consolidated control findings, you receive only one finding for the control. For more information, see Consolidated control findings.

The samples on this page provide examples for both scenarios. The samples include: control findings for individual Security Hub CSPM standards when consolidated control findings is disabled, and a control finding for multiple Security Hub CSPM standards when consolidated control findings is enabled.

Samples of control findings

- Sample finding for the AWS Foundational Security Best Practices standard
- Sample finding for CIS AWS Foundations Benchmark v3.0.0
- Sample finding for CIS AWS Foundations Benchmark v1.4.0
- Sample finding for CIS AWS Foundations Benchmark v1.2.0
- Sample finding for the NIST SP 800-53 Revision 5 standard
- Sample finding for the NIST SP 800-171 Revision 2 standard
- Sample finding for Payment Card Industry Data Security Standard v3.2.1
- Sample finding for the AWS Resource Tagging standard
- Sample finding for the AWS Control Tower service-managed standard
- Sample consolidated finding for multiple standards

> **Note**
>
> Control findings reference different fields and values in the China Regions and the AWS GovCloud (US) Regions. For more information, see Impact of consolidation on ASFF fields and values.

Sample finding for the AWS Foundational Security Best Practices standard

The following sample provides an example of a finding for a control that applies to the AWS Foundational Security Best Practices (FSBP) standard. In this sample, consolidated control findings is disabled.

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-2:123456789012:subscription/aws-foundational-
security-best-practices/v/1.0.0/CloudTrail.2/finding/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:us-east-2::product/aws/securityhub",
```

```
  "ProductName": "Security Hub CSPM",
  "CompanyName": "AWS",
  "Region": "us-east-2",
  "GeneratorId": "aws-foundational-security-best-practices/v/1.0.0/CloudTrail.2",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards/AWS-
Foundational-Security-Best-Practices"
  ],
  "FirstObservedAt": "2020-08-06T02:18:23.076Z",
  "LastObservedAt": "2021-09-28T16:10:06.956Z",
  "CreatedAt": "2020-08-06T02:18:23.076Z",
  "UpdatedAt": "2021-09-28T16:10:00.093Z",
  "Severity": {
    "Product": 40,
    "Label": "MEDIUM",
    "Normalized": 40,
    "Original": "MEDIUM"
  },
  "Title": "CloudTrail.2 CloudTrail should have encryption at-rest enabled",
  "Description": "This AWS control checks whether AWS CloudTrail is configured to use
 the server side encryption (SSE) AWS Key Management Service (AWS KMS) customer master
 key (CMK) encryption. The check will pass if the KmsKeyId is defined.",
  "Remediation": {
    "Recommendation": {
      "Text": "For directions on how to correct this issue, consult the AWS Security
 Hub CSPM controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation"
    }
  },
  "ProductFields": {
    "StandardsArn": "arn:aws:securityhub:::standards/aws-foundational-security-best-
practices/v/1.0.0",
    "StandardsSubscriptionArn": "arn:aws:securityhub:us-
east-2:123456789012:subscription/aws-foundational-security-best-practices/v/1.0.0",
    "ControlId": "CloudTrail.2",
    "RecommendationUrl": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/
remediation",
    "RelatedAWSResources:0/name": "securityhub-cloud-trail-encryption-enabled-
fe95bf3f",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "StandardsControlArn": "arn:aws:securityhub:us-east-2:123456789012:control/aws-
foundational-security-best-practices/v/1.0.0/CloudTrail.2",
    "aws/securityhub/ProductName": "Security Hub CSPM",
```

```
    "aws/securityhub/CompanyName": "AWS",
    "Resources:0/Id": "arn:aws:cloudtrail:us-east-2:123456789012:trail/AWSMacieTrail-
DO-NOT-EDIT",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-2::product/aws/
securityhub/arn:aws:securityhub:us-east-2:123456789012:subscription/aws-foundational-
security-best-practices/v/1.0.0/CloudTrail.2/finding/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
  },
  "Resources": [
    {
      "Type": "AwsCloudTrailTrail",
      "Id": "arn:aws:cloudtrail:us-east-2:123456789012:trail/AWSMacieTrail-DO-NOT-
EDIT",
      "Partition": "aws",
      "Region": "us-east-2"
    }
  ],
  "Compliance": {
    "Status": "FAILED",
    "SecurityControlId": "CloudTrail.2",
    "AssociatedStandards": [{
      "StandardsId": "standards/aws-foundation-best-practices/v/1.0.0"
    }]
  },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "MEDIUM",
      "Original": "MEDIUM"
    },
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards/AWS-
Foundational-Security-Best-Practices"
    ]
  }
}
```

Sample finding for CIS AWS Foundations Benchmark v3.0.0

The following sample provides an example of a finding for a control that applies to CIS AWS Foundations Benchmark v3.0.0. In this sample, consolidated control findings is disabled.


```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-1:123456789012:subscription/cis-aws-foundations-
benchmark/v/3.0.0/2.2.1/finding/38a89798-6819-4fae-861f-9cca8034602c",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/securityhub",
  "ProductName": "Security Hub CSPM",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "cis-aws-foundations-benchmark/v/3.0.0/2.2.1",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards/CIS AWS
 Foundations Benchmark"
  ],
  "FirstObservedAt": "2024-04-18T07:46:18.193Z",
  "LastObservedAt": "2024-04-23T07:47:01.137Z",
  "CreatedAt": "2024-04-18T07:46:18.193Z",
  "UpdatedAt": "2024-04-23T07:46:46.165Z",
  "Severity": {
    "Product": 40,
    "Label": "MEDIUM",
    "Normalized": 40,
    "Original": "MEDIUM"
  },
  "Title": "2.2.1 EBS default encryption should be enabled",
  "Description": "Elastic Compute Cloud (EC2) supports encryption at rest when using
 the Elastic Block Store (EBS) service. While disabled by default, forcing encryption
 at EBS volume creation is supported.",
  "Remediation": {
    "Recommendation": {
      "Text": "For information on how to correct this issue, consult the AWS Security
 Hub CSPM controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/EC2.7/remediation"
    }
  },
  "ProductFields": {
    "StandardsArn": "arn:aws:securityhub:::standards/cis-aws-foundations-benchmark/
v/3.0.0",
```

```
    "StandardsSubscriptionArn": "arn:aws:securityhub:us-
east-1:123456789012:subscription/cis-aws-foundations-benchmark/v/3.0.0",
    "ControlId": "2.2.1",
    "RecommendationUrl": "https://docs.aws.amazon.com/console/securityhub/EC2.7/
remediation",
    "RelatedAWSResources:0/name": "securityhub-ec2-ebs-encryption-by-default-2843ed9e",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "StandardsControlArn": "arn:aws:securityhub:us-east-1:123456789012:control/cis-aws-
foundations-benchmark/v/3.0.0/2.2.1",
    "aws/securityhub/ProductName": "Security Hub CSPM",
    "aws/securityhub/CompanyName": "AWS",
    "aws/securityhub/annotation": "EBS Encryption by default is not enabled.",
    "Resources:0/Id": "arn:aws:iam::123456789012:root",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/
securityhub/arn:aws:securityhub:us-east-1:123456789012:subscription/cis-aws-
foundations-benchmark/v/3.0.0/2.2.1/finding/38a89798-6819-4fae-861f-9cca8034602c"
  },
  "Resources": [
    {
      "Type": "AwsAccount",
      "Id": "AWS::::Account:123456789012",
      "Partition": "aws",
      "Region": "us-east-1"
    }
  ],
  "Compliance": {
    "Status": "FAILED",
    "RelatedRequirements": [
      "CIS AWS Foundations Benchmark v3.0.0/2.2.1"
    ],
    "SecurityControlId": "EC2.7",
    "AssociatedStandards": [
      {
        "StandardsId": "standards/cis-aws-foundations-benchmark/v/3.0.0"
      }
    ]
  },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
```

```
      "Label": "MEDIUM",
      "Original": "MEDIUM"
    },
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards/CIS AWS
 Foundations Benchmark"
    ]
  },
  "ProcessedAt": "2024-04-23T07:47:07.088Z"
}
```

Sample finding for CIS AWS Foundations Benchmark v1.4.0

The following sample provides an example of a finding for a control that applies to CIS AWS Foundations Benchmark v1.4.0. In this sample, consolidated control findings is disabled.

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-1:123456789012:subscription/cis-aws-foundations-
benchmark/v/1.4.0/3.7/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/securityhub",
  "ProductName": "Security Hub CSPM",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "cis-aws-foundations-benchmark/v/1.4.0/3.7",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards/CIS AWS
 Foundations Benchmark"
  ],
  "FirstObservedAt": "2022-10-21T22:14:48.913Z",
  "LastObservedAt": "2022-12-22T22:24:56.980Z",
  "CreatedAt": "2022-10-21T22:14:48.913Z",
  "UpdatedAt": "2022-12-22T22:24:52.409Z",
  "Severity": {
    "Product": 40,
    "Label": "MEDIUM",
    "Normalized": 40,
    "Original": "MEDIUM"
  },
  "Title": "3.7 Ensure CloudTrail logs are encrypted at rest using KMS CMKs",
  "Description": "AWS CloudTrail is a web service that records AWS API calls for an
 account and makes those logs available to users and resources in accordance with IAM
```

```
 policies. AWS Key Management Service (KMS) is a managed service that helps create
 and control the encryption keys used to encrypt account data, and uses Hardware
 Security Modules (HSMs) to protect the security of encryption keys. CloudTrail logs
 can be configured to leverage server side encryption (SSE) and AWS KMS customer
 created master keys (CMK) to further protect CloudTrail logs. It is recommended that
 CloudTrail be configured to use SSE-KMS.",
  "Remediation": {
    "Recommendation": {
      "Text": "For directions on how to correct this issue, consult the AWS Security
 Hub CSPM controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation"
    }
  },
  "ProductFields": {
    "StandardsArn": "arn:aws:securityhub:::standards/cis-aws-foundations-benchmark/
v/1.4.0",
    "StandardsSubscriptionArn": "arn:aws:securityhub:us-
east-1:123456789012:subscription/cis-aws-foundations-benchmark/v/1.4.0",
    "ControlId": "3.7",
    "RecommendationUrl": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/
remediation",
    "RelatedAWSResources:0/name": "securityhub-cloud-trail-encryption-
enabled-855f82d1",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "StandardsControlArn": "arn:aws:securityhub:us-east-1:123456789012:control/cis-aws-
foundations-benchmark/v/1.4.0/3.7",
    "aws/securityhub/ProductName": "Security Hub CSPM",
    "aws/securityhub/CompanyName": "AWS",
    "Resources:0/Id": "arn:aws:cloudtrail:us-west-2:123456789012:trail/AWSMacieTrail-
DO-NOT-EDIT",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/
securityhub/arn:aws:securityhub:us-east-1:123456789012:subscription/cis-aws-
foundations-benchmark/v/1.4.0/3.7/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "Resources": [
    {
      "Type": "AwsCloudTrailTrail",
      "Id": "arn:aws:cloudtrail:us-west-2:123456789012:trail/AWSMacieTrail-DO-NOT-
EDIT",
      "Partition": "aws",
      "Region": "us-east-1"
    }
  ],
  "Compliance": {
```

```
    "Status": "FAILED",
    "RelatedRequirements": [
      "CIS AWS Foundations Benchmark v1.4.0/3.7"
    ],
    "SecurityControlId": "CloudTrail.2",
    "AssociatedStandards": [{
      "StandardsId": "standards/cis-aws-foundations-benchmark/v/1.4.0"
    }]
  },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "MEDIUM",
      "Original": "MEDIUM"
    },
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards/CIS AWS
 Foundations Benchmark"
    ]
  }
}
```

Sample finding for CIS AWS Foundations Benchmark v1.2.0

The following sample provides an example of a finding for a control that applies to CIS AWS Foundations Benchmark v1.2.0. In this sample, consolidated control findings is disabled.

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-2:123456789012:subscription/cis-aws-foundations-
benchmark/v/1.2.0/2.7/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:us-east-2::product/aws/securityhub",
  "ProductName": "Security Hub CSPM",
  "CompanyName": "AWS",
  "Region": "us-east-2",
  "GeneratorId": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/
rule/2.7",
  "AwsAccountId": "123456789012",
  "Types": [
```

```
    "Software and Configuration Checks/Industry and Regulatory Standards/CIS AWS
 Foundations Benchmark"
  ],
  "FirstObservedAt": "2020-08-29T04:10:06.337Z",
  "LastObservedAt": "2021-09-28T16:10:05.350Z",
  "CreatedAt": "2020-08-29T04:10:06.337Z",
  "UpdatedAt": "2021-09-28T16:10:00.087Z",
  "Severity": {
    "Product": 40,
    "Label": "MEDIUM",
    "Normalized": 40,
    "Original": "MEDIUM"
  },
  "Title": "2.7 Ensure CloudTrail logs are encrypted at rest using KMS CMKs",
  "Description": "AWS Key Management Service (KMS) is a managed service that helps
 create and control the encryption keys used to encrypt account data, and uses Hardware
 Security Modules (HSMs) to protect the security of encryption keys. CloudTrail
 logs can be configured to leverage server side encryption (SSE) and KMS customer
 created master keys (CMK) to further protect CloudTrail logs. It is recommended that
 CloudTrail be configured to use SSE-KMS.",
  "Remediation": {
    "Recommendation": {
      "Text": "For directions on how to correct this issue, consult the AWS Security
 Hub CSPM controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation"
    }
  },
  "ProductFields": {
    "StandardsGuideArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0",
    "StandardsGuideSubscriptionArn": "arn:aws:securityhub:us-
east-2:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0",
    "RuleId": "2.7",
    "RecommendationUrl": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/
remediation",
    "RelatedAWSResources:0/name": "securityhub-cloud-trail-encryption-enabled-
fe95bf3f",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "StandardsControlArn": "arn:aws:securityhub:us-east-2:123456789012:control/cis-aws-
foundations-benchmark/v/1.2.0/2.7",
    "aws/securityhub/ProductName": "Security Hub CSPM",
    "aws/securityhub/CompanyName": "AWS",
    "Resources:0/Id": "arn:aws:cloudtrail:us-east-2:123456789012:trail/AWSMacieTrail-
DO-NOT-EDIT",
```

```
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-2::product/aws/
securityhub/arn:aws:securityhub:us-east-2:123456789012:subscription/cis-aws-
foundations-benchmark/v/1.2.0/2.7/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "Resources": [
    {
      "Type": "AwsCloudTrailTrail",
      "Id": "arn:aws:cloudtrail:us-east-2:123456789012:trail/AWSMacieTrail-DO-NOT-
EDIT",
      "Partition": "aws",
      "Region": "us-east-2"
    }
  ],
  "Compliance": {
    "Status": "FAILED",
    "SecurityControlId": "CloudTrail.2",
    "AssociatedStandards": [{
      "StandardsId": "ruleset/cis-aws-foundations-benchmark/v/1.2.0"
    }]
  },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "MEDIUM",
      "Original": "MEDIUM"
    },
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards/CIS AWS
 Foundations Benchmark"
    ]
  }
}
```

Sample finding for the NIST SP 800-53 Revision 5 standard

The following sample provides an example of a finding for a control that applies to the NIST SP 800-53 Revision 5 standard. In this sample, consolidated control findings is disabled.

```
{
```

```
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-1:123456789012:subscription/nist-800-53/v/5.0.0/
CloudTrail.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/securityhub",
  "ProductName": "Security Hub CSPM",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "nist-800-53/v/5.0.0/CloudTrail.2",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards"
  ],
  "FirstObservedAt": "2023-02-17T14:22:46.726Z",
  "LastObservedAt": "2023-02-17T14:22:50.846Z",
  "CreatedAt": "2023-02-17T14:22:46.726Z",
  "UpdatedAt": "2023-02-17T14:22:46.726Z",
  "Severity": {
    "Product": 40,
    "Label": "MEDIUM",
    "Normalized": 40,
    "Original": "MEDIUM"
  },
  "Title": "CloudTrail.2 CloudTrail should have encryption at-rest enabled",
  "Description": "This AWS control checks whether AWS CloudTrail is configured to use
 the server side encryption (SSE) AWS Key Management Service (AWS KMS) customer master
 key (CMK) encryption. The check will pass if the KmsKeyId is defined.",
  "Remediation": {
    "Recommendation": {
      "Text": "For directions on how to fix this issue, consult the AWS Security Hub
 CSPM NIST 800-53 R5 documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation"
    }
  },
  "ProductFields": {
    "StandardsArn": "arn:aws:securityhub:::standards/nist-800-53/v/5.0.0",
    "StandardsSubscriptionArn": "arn:aws:securityhub:us-
east-1:123456789012:subscription/nist-800-53/v/5.0.0",
    "ControlId": "CloudTrail.2",
    "RecommendationUrl": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.9/
remediation",
    "RelatedAWSResources:0/name": "securityhub-cloud-trail-encryption-enabled-
fe95bf3f",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
```

```
    "StandardsControlArn": "arn:aws:securityhub:us-east-2:123456789012:control/aws-
foundational-security-best-practices/v/1.0.0/CloudTrail.2",
    "aws/securityhub/ProductName": "Security Hub CSPM",
    "aws/securityhub/CompanyName": "AWS",
    "Resources:0/Id": "arn:aws:cloudtrail:us-west-2:123456789012:trail/AWSMacieTrail-
DO-NOT-EDIT",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/
securityhub/arn:aws:securityhub:us-east-1:123456789012:subscription/nist-800-53/
v/5.0.0/CloudTrail.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "Resources": [
    {
      "Type": "AwsCloudTrailTrail",

      "Id": "arn:aws:cloudtrail:us-east-1:123456789012:trail/AWSMacieTrail-DO-NOT-
EDIT",

      "Partition": "aws",

      "Region": "us-east-1"
    }
  ],
  "Compliance": {
    "Status": "FAILED",
    "RelatedRequirements": [
        "NIST.800-53.r5 AU-9",
        "NIST.800-53.r5 CA-9(1)",
        "NIST.800-53.r5 CM-3(6)",
        "NIST.800-53.r5 SC-13",
        "NIST.800-53.r5 SC-28",
        "NIST.800-53.r5 SC-28(1)",
        "NIST.800-53.r5 SC-7(10)",
        "NIST.800-53.r5 SI-7(6)"
    ],
    "SecurityControlId": "CloudTrail.2",
    "AssociatedStandards": [
      {
        "StandardsId": "standards/nist-800-53/v/5.0.0"
      }
    ]
  },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
```

```
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "MEDIUM",
      "Original": "MEDIUM"
    },
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards"
    ]
  },
  "ProcessedAt": "2023-02-17T14:22:53.572Z"
}
```

Sample finding for the NIST SP 800-171 Revision 2 standard

The following sample provides an example of a finding for a control that applies to the NIST SP 800-171 Revision 2 standard. In this sample, consolidated control findings is disabled.

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-1:123456789012:subscription/nist-800-171/v/2.0.0/
CloudTrail.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/securityhub",
  "ProductName": "Security Hub",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "nist-800-171/v/2.0.0/CloudTrail.2",
  "AwsAccountId": "123456789012",
  "AwsAccountName": "TestAcct",
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards"
  ],
  "FirstObservedAt": "2025-05-29T05:23:58.690Z",
  "LastObservedAt": "2025-05-30T05:50:11.898Z",
  "CreatedAt": "2025-05-29T05:24:24.772Z",
  "UpdatedAt": "2025-05-30T05:50:34.292Z",
  "Severity": {
    "Product": 40,
    "Label": "MEDIUM",
    "Normalized": 40,
    "Original": "MEDIUM"
  },
```

```
  "Title": "CloudTrail.2 CloudTrail should have encryption at-rest enabled",
  "Description": "This AWS control checks whether AWS CloudTrail is configured to use
 the server side encryption (SSE) AWS Key Management Service (AWS KMS) customer master
 key (CMK) encryption. The check will pass if the KmsKeyId is defined.",
  "Remediation": {
    "Recommendation": {
      "Text": "For information on how to correct this issue, consult the AWS Security
 Hub CSPM controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation"
    }
  },
  "ProductFields": {
    "StandardsArn": "arn:aws:securityhub:::standards/nist-800-171/v/2.0.0",
    "StandardsSubscriptionArn": "arn:aws:securityhub:us-
east-1:123456789012:subscription/nist-800-171/v/2.0.0",
    "ControlId": "CloudTrail.2",
    "RecommendationUrl": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/
remediation",
    "RelatedAWSResources:0/name": "securityhub-cloud-trail-encryption-
enabled-0ab1c2d4",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "StandardsControlArn": "arn:aws:securityhub:us-east-1:123456789012:control/
nist-800-171/v/2.0.0/CloudTrail.2",
    "aws/securityhub/ProductName": "Security Hub",
    "aws/securityhub/CompanyName": "AWS",
    "Resources:0/Id": "arn:aws:cloudtrail:ca-central-1:123456789012:trail/aws-
BaselineCloudTrail",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/
securityhub/arn:aws:securityhub:us-east-1:123456789012:subscription/nist-800-171/
v/2.0.0/CloudTrail.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "Resources": [
    {
      "Id": "arn:aws:cloudtrail:ca-central-1:123456789012:trail/aws-
BaselineCloudTrail",
      "Partition": "aws",
      "Region": "us-east-1",
      "Type": "AwsCloudTrailTrail"
    }
  ],
  "Compliance": {
    "Status": "FAILED",
    "SecurityControlId": "CloudTrail.2",
    "RelatedRequirements": [
```

```
      "NIST.800-171.r2/3.3.8"
    ],
    "AssociatedStandards": [
      {
        "StandardsId": "standards/nist-800-171/v/2.0.0"
      }
    ]
  },
  "Workflow": {
    "Status": "NEW"
  },
  "WorkflowState": "NEW",
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards"
    ],
    "Severity": {
      "Product": 40,
      "Label": "MEDIUM",
      "Normalized": 40,
      "Original": "MEDIUM"
    }
  },
  "ProcessedAt": "2025-05-30T05:50:40.297Z"
}
```

Sample finding for Payment Card Industry Data Security Standard v3.2.1

The following sample provides an example of a finding for a control that applies to Payment Card Industry Data Security Standard (PCI DSS) v3.2.1. In this sample, consolidated control findings is disabled.

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-2:123456789012:subscription/pci-dss/v/3.2.1/
PCI.CloudTrail.1/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:us-east-2::product/aws/securityhub",
  "ProductName": "Security Hub CSPM",
  "CompanyName": "AWS",
  "Region": "us-east-2",
  "GeneratorId": "pci-dss/v/3.2.1/PCI.CloudTrail.1",
  "AwsAccountId": "123456789012",
```

```
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards/PCI-DSS"
  ],
  "FirstObservedAt": "2020-08-06T02:18:23.089Z",
  "LastObservedAt": "2021-09-28T16:10:06.942Z",
  "CreatedAt": "2020-08-06T02:18:23.089Z",
  "UpdatedAt": "2021-09-28T16:10:00.090Z",
  "Severity": {
    "Product": 40,
    "Label": "MEDIUM",
    "Normalized": 40,
    "Original": "MEDIUM"
  },
  "Title": "PCI.CloudTrail.1 CloudTrail logs should be encrypted at rest using AWS KMS
 CMKs",
  "Description": "This AWS control checks whether AWS CloudTrail is configured to use
 the server side encryption (SSE) AWS Key Management Service (AWS KMS) customer master
 key (CMK) encryption by checking if the KmsKeyId is defined.",
  "Remediation": {
    "Recommendation": {
      "Text": "For directions on how to correct this issue, consult the AWS Security
 Hub CSPM controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation"
    }
  },
  "ProductFields": {
    "StandardsArn": "arn:aws:securityhub:::standards/pci-dss/v/3.2.1",
    "StandardsSubscriptionArn": "arn:aws:securityhub:us-
east-2:123456789012:subscription/pci-dss/v/3.2.1",
    "ControlId": "PCI.CloudTrail.1",
    "RecommendationUrl": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/
remediation",
    "RelatedAWSResources:0/name": "securityhub-cloud-trail-encryption-enabled-
fe95bf3f",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "StandardsControlArn": "arn:aws:securityhub:us-east-2:123456789012:control/pci-dss/
v/3.2.1/PCI.CloudTrail.1",
    "aws/securityhub/ProductName": "Security Hub CSPM",
    "aws/securityhub/CompanyName": "AWS",
    "Resources:0/Id": "arn:aws:cloudtrail:us-east-2:123456789012:trail/AWSMacieTrail-
DO-NOT-EDIT",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-2::product/aws/
securityhub/arn:aws:securityhub:us-east-2:123456789012:subscription/pci-dss/v/3.2.1/
PCI.CloudTrail.1/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

```
  },
  "Resources": [
    {
      "Type": "AwsCloudTrailTrail",
      "Id": "arn:aws:cloudtrail:us-east-2:123456789012:trail/AWSMacieTrail-DO-NOT-
EDIT",
      "Partition": "aws",
      "Region": "us-east-2"
    }
  ],
  "Compliance": {
    "Status": "FAILED",
    "RelatedRequirements": [
      "PCI DSS 3.4"
    ],
    "SecurityControlId": "CloudTrail.2",
    "AssociatedStandards": [{
      "StandardsId": "standards/pci-dss/v/3.2.1"
    }]
  },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "MEDIUM",
      "Original": "MEDIUM"
    },
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards/PCI-DSS"
    ]
  }
}
```

Sample finding for the AWS Resource Tagging standard

The following sample provides an example of a finding for a control that applies to the AWS Resource Tagging standard. In this sample, consolidated control findings is disabled.

```
{
  "SchemaVersion": "2018-10-08",
```

```
  "Id": "arn:aws:securityhub:eu-central-1:123456789012:security-control/EC2.44/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:eu-central-1::product/aws/securityhub",
  "ProductName": "Security Hub CSPM",
  "CompanyName": "AWS",
  "Region": "eu-central-1",
  "GeneratorId": "security-control/EC2.44",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards"
  ],
  "FirstObservedAt": "2024-02-19T21:00:32.206Z",
  "LastObservedAt": "2024-04-29T13:01:57.861Z",
  "CreatedAt": "2024-02-19T21:00:32.206Z",
  "UpdatedAt": "2024-04-29T13:01:41.242Z",
  "Severity": {
    "Label": "LOW",
    "Normalized": 1,
    "Original": "LOW"
  },
  "Title": "EC2 subnets should be tagged",
  "Description": "This control checks whether an Amazon EC2 subnet has tags with the
 specific keys defined in the parameter requiredTagKeys. The control fails if the
 subnet doesn't have any tag keys or if it doesn't have all the keys specified in
 the parameter requiredTagKeys. If the parameter requiredTagKeys isn't provided, the
 control only checks for the existence of a tag key and fails if the subnet isn't
 tagged with any key. System tags, which are automatically applied and begin with aws:,
 are ignored.",
  "Remediation": {
    "Recommendation": {
      "Text": "For information on how to correct this issue, consult the AWS Security
 Hub CSPM controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/EC2.44/remediation"
    }
  },
  "ProductFields": {
    "RelatedAWSResources:0/name": "securityhub-tagged-ec2-subnet-6ceafede",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "aws/securityhub/ProductName": "Security Hub CSPM",
    "aws/securityhub/CompanyName": "AWS",
    "aws/securityhub/annotation": "No tags are present.",
    "Resources:0/Id": "arn:aws:ec2:eu-central-1:123456789012:subnet/
subnet-1234567890abcdef0",
```

```
    "aws/securityhub/FindingId": "arn:aws:securityhub:eu-central-1::product/aws/
securityhub/arn:aws:securityhub:eu-central-1:123456789012:security-control/EC2.44/
finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "Resources": [
    {
      "Type": "AwsEc2Subnet",
      "Id": "arn:aws:ec2:eu-central-1:123456789012:subnet/subnet-1234567890abcdef0",
      "Partition": "aws",
      "Region": "eu-central-1",
      "Details": {
        "AwsEc2Subnet": {
          "AssignIpv6AddressOnCreation": false,
          "AvailabilityZone": "eu-central-1b",
          "AvailabilityZoneId": "euc1-az3",
          "AvailableIpAddressCount": 4091,
          "CidrBlock": "10.24.34.0/23",
          "DefaultForAz": true,
          "MapPublicIpOnLaunch": true,
          "OwnerId": "123456789012",
          "State": "available",
          "SubnetArn": "arn:aws:ec2:eu-central-1:123456789012:subnet/
subnet-1234567890abcdef0",
          "SubnetId": "subnet-1234567890abcdef0",
          "VpcId": "vpc-021345abcdef6789"
        }
      }
    }
  ],
  "Compliance": {
    "Status": "FAILED",
    "SecurityControlId": "EC2.44",
    "AssociatedStandards": [
      {
        "StandardsId": "standards/aws-resource-tagging-standard/v/1.0.0"
      }
    ],
    "SecurityControlParameters": [
      {
        "Name": "requiredTagKeys",
        "Value": [
          "peepoo"
        ]
      }
```

```
    ],
            },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "LOW",
      "Original": "LOW"
    },
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards"
    ]
  },
  "ProcessedAt": "2024-04-29T13:02:03.259Z"
}
```

Sample finding for the AWS Control Tower service-managed standard

The following sample provides an example of a finding for a control that applies to the AWS Control Tower service-managed standard. In this sample, consolidated control findings is disabled.

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-1:123456789012:subscription/service-managed-aws-
control-tower/v/1.0.0/CloudTrail.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/securityhub",
  "ProductName": "Security Hub CSPM",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "service-managed-aws-control-tower/v/1.0.0/CloudTrail.2",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards"
  ],
  "FirstObservedAt": "2022-11-17T01:25:30.296Z",
  "LastObservedAt": "2022-11-17T01:25:45.805Z",
  "CreatedAt": "2022-11-17T01:25:30.296Z",
  "UpdatedAt": "2022-11-17T01:25:30.296Z",
  "Severity": {
    "Product": 40,
```

```
    "Label": "MEDIUM",
    "Normalized": 40,
    "Original": "MEDIUM"
  },
  "Title": "CT.CloudTrail.2 CloudTrail should have encryption at-rest enabled",
  "Description": "This AWS control checks whether AWS CloudTrail is configured to use
 the server side encryption (SSE) AWS Key Management Service (AWS KMS) customer master
 key (CMK) encryption. The check will pass if the KmsKeyId is defined.",
  "Remediation": {
    "Recommendation": {
      "Text": "For information on how to correct this issue, consult the AWS Security
 Hub CSPM controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation"
    }
  },
  "ProductFields": {
    "StandardsArn": "arn:aws:securityhub:::standards/service-managed-aws-control-tower/
v/1.0.0",
    "StandardsSubscriptionArn": "arn:aws:securityhub:us-
east-1:123456789012:subscription/service-managed-aws-control-tower/v/1.0.0",
    "ControlId": "CT.CloudTrail.2",
    "RecommendationUrl": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/
remediation",
    "RelatedAWSResources:0/name": "securityhub-cloud-trail-encryption-enabled-
fe95bf3f",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "StandardsControlArn": "arn:aws:securityhub:us-east-1:123456789012:control/service-
managed-aws-control-tower/v/1.0.0/CloudTrail.2",
    "aws/securityhub/ProductName": "Security Hub CSPM",
    "aws/securityhub/CompanyName": "AWS",
    "Resources:0/Id": "arn:aws:cloudtrail:us-east-2:123456789012:trail/AWSMacieTrail-
DO-NOT-EDIT",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/
securityhub/arn:aws:securityhub:us-east-1:123456789012:subscription/service-managed-
aws-control-tower/v/1.0.0/CloudTrail.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "Resources": [
    {
      "Type": "AwsAccount",
      "Id": "AWS::::Account:123456789012",
      "Partition": "aws",
      "Region": "us-east-1"
    }
  ],
```

```
  "Compliance": {
    "Status": "FAILED",
    "SecurityControlId": "CloudTrail.2",
    "AssociatedStandards": [{
      "StandardsId": "standards/service-managed-aws-control-tower/v/1.0.0"
    }]
  },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "MEDIUM",
      "Original": "MEDIUM"
    },
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards"
    ]
  }
}
```

Sample consolidated finding for multiple standards

The following sample provides an example of a finding for a control that applies to multiple enabled standards. In this sample, consolidated control findings is enabled.

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-1:123456789012:security-control/CloudTrail.2/
finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/securityhub",
  "ProductName": "Security Hub",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "security-control/CloudTrail.2",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards"
  ],
  "FirstObservedAt": "2024-08-09T14:57:04.521Z",
  "LastObservedAt": "2025-05-30T03:30:17.407Z",
```

```
  "CreatedAt": "2024-08-09T14:57:04.521Z",
  "UpdatedAt": "2025-05-30T03:30:32.781Z",
  "Severity": {
    "Label": "MEDIUM",
    "Normalized": 40,
    "Original": "MEDIUM"
  },
  "Title": "CloudTrail should have encryption at-rest enabled",
  "Description": "This AWS control checks whether AWS CloudTrail is configured to use
 the server side encryption (SSE) AWS Key Management Service (AWS KMS) customer master
 key (CMK) encryption. The check will pass if the KmsKeyId is defined.",
  "Remediation": {
    "Recommendation": {
      "Text": "For information on how to correct this issue, consult the AWS Security
 Hub CSPM controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation"
    }
  },
  "ProductFields": {
    "RelatedAWSResources:0/name": "securityhub-cloud-trail-encryption-
enabled-01a2b345",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "aws/securityhub/ProductName": "Security Hub",
    "aws/securityhub/CompanyName": "AWS",
    "Resources:0/Id": "arn:aws:cloudtrail:us-east-1:123456789012:trail/TestTrail-DO-
NOT-DELETE",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/
securityhub/arn:aws:securityhub:us-east-1:123456789012:security-control/CloudTrail.2/
finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "Resources": [
    {
      "Type": "AwsCloudTrailTrail",
      "Id": "arn:aws:cloudtrail:us-east-1:123456789012:trail/TestTrail-DO-NOT-DELETE",
      "Partition": "aws",
      "Region": "us-east-1",
      "Details": {
        "AwsCloudTrailTrail": {
          "HasCustomEventSelectors": false,
          "IncludeGlobalServiceEvents": true,
          "LogFileValidationEnabled": true,
          "HomeRegion": "us-east-1",
          "IsMultiRegionTrail": true,
          "S3BucketName": "cloudtrail-awslogs-do-not-delete",
```

```
          "IsOrganizationTrail": false,
          "Name": "TestTrail-DO-NOT-DELETE"
        }
      }
    }
  ],
  "Compliance": {
    "Status": "FAILED",
    "SecurityControlId": "CloudTrail.2",
    "RelatedRequirements": [
      "CIS AWS Foundations Benchmark v1.2.0/2.7",
      "CIS AWS Foundations Benchmark v1.4.0/3.7",
      "CIS AWS Foundations Benchmark v3.0.0/3.5",
      "NIST.800-171.r2/3.3.8",
      "PCI DSS v3.2.1/3.4",
      "PCI DSS v4.0.1/10.3.2"
    ],
    "AssociatedStandards": [
      { "StandardsId": "ruleset/cis-aws-foundations-benchmark/v/1.2.0"},
      { "StandardsId": "standards/aws-foundational-security-best-practices/v/1.0.0"},
      { "StandardsId": "standards/cis-aws-foundations-benchmark/v/1.4.0"},
      { "StandardsId": "standards/cis-aws-foundations-benchmark/v/3.0.0"},
      { "StandardsId": "standards/nist-800-171/v/2.0.0"},
      { "StandardsId": "standards/pci-dss/v/3.2.1"},
      { "StandardsId": "standards/pci-dss/v/4.0.1"}
    ]
  },
  "Workflow": {
    "Status": "NEW"
  },
  "WorkflowState": "NEW",
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory Standards"
    ],
    "Severity": {
      "Normalized": 40,
      "Label": "MEDIUM",
      "Original": "MEDIUM"
    }
  },
  "ProcessedAt": "2025-05-30T03:31:00.831Z"
```

```
}
```

Understanding integrations in Security Hub CSPM

AWS Security Hub CSPM can ingest security findings from several AWS services and supported third-party AWS Partner Network security solutions. These integrations can help you get a comprehensive view of security and compliance across your AWS environment. Security Hub CSPM ingests findings from integrated solutions and converts them to the AWS Security Finding Format (ASFF).

> **Important**
>
> For supported AWS and third-party product integrations, Security Hub CSPM receives and consolidates findings that are generated only after you enable Security Hub CSPM for your AWS accounts. The service doesn't retroactively receive and consolidate security findings that were generated before you enabled Security Hub CSPM.

The **Integrations** page of the Security Hub CSPM console provides access to available AWS and third-party product integrations. The Security Hub CSPM API also has operations for managing integrations.

An integration might not be available in all AWS Regions. If an integration isn't supported in the Region that you are currently signed in to on the Security Hub CSPM console, it doesn't appear on the **Integrations** page of the console. For a list of integrations that are available in the China Regions and AWS GovCloud (US) Regions, see Availability of integrations by Region.

In addition to AWS service and built-in third-party integrations, you can integrate custom security products with Security Hub CSPM. You can then send findings from these products to Security Hub CSPM by using the Security Hub CSPM API. You can also use the API to update existing findings that Security Hub CSPM received from a custom security product.

Topics

- AWS service integrations with Security Hub CSPM
- Third-party product integrations with Security Hub CSPM
- Integrating Security Hub CSPM with custom products

Reviewing a list of Security Hub CSPM integrations

Choose your preferred method, and follow the steps to review a list of integrations in AWS Security Hub CSPM or details about a specific integration.

Security Hub CSPM console

To review integration options and details (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the Security Hub CSPM navigation pane, choose **Integrations**.

On the **Integrations** page, integrations with other AWS services are listed first, followed by integrations with third-party products.

For each integration, the **Integrations** page provides the following information:

- The name of the company
- The name of the product
- A description of the integration
- The categories that the integration applies to
- How to enable the integration
- The current status of the integration

You can filter the list by entering text from the following fields:

- Company name
- Product name
- Integration description
- Categories

Security Hub CSPM API

To review integration options and details (API)

To get a list of integrations, use the DescribeProducts operation. If you're using the AWS CLI, run the describe-products command.

To retrieve details for a specific product integration, use the `ProductArn` parameter to specify the Amazon Resource Name (ARN) of the integration.

For example, the following AWS CLI command retrieves details about the Security Hub CSPM integration with 3CORESec.

```
$ aws securityhub describe-products --product-arn "arn:aws:securityhub:us-east-1::product/3coresec/3coresec"
```

Enabling the flow of findings from a Security Hub CSPM integration

On the **Integrations** page of the AWS Security Hub CSPM console, you can see the required steps to enable each integration.

For most of the integrations with other AWS services, the only required step to enable the integration is to enable the other service. The integration information includes a link to the other service's home page. When you enable the other service, a resource-level permission that allows Security Hub CSPM to receive findings from the service is then automatically created and applied.

For third-party product integrations, you may need to purchase the integration from the AWS Marketplace, and then configure the integration. The integration information provides links to complete these tasks.

If more than one version of a product is available in AWS Marketplace, select the version that you wan to subscribe to, and then choose **Continue to Subscribe**. For example, some products offer a standard version and an AWS GovCloud (US) version.

When you enable a product integration, a resource policy is automatically attached to that product subscription. This resource policy defines the permissions that Security Hub CSPM needs to receive findings from that product.

After you complete any preliminary steps to enable an integration, you can then disable and re-enable the flow of findings from that integration. On the **Integrations** page, for integrations that send findings, the **Status** information indicates whether you are currently accepting findings.

Security Hub CSPM console

To enable the flow of findings from an integration (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the Security Hub CSPM navigation pane, choose **Integrations**.
3. For integrations that send findings, the **Status** information indicates whether Security Hub CSPM is currently accepting findings from that integration.
4. Choose **Accept findings**.

Security Hub CSPM API

Use the EnableImportFindingsForProduct operation. If you're using the AWS CLI, run the enable-import-findings-for-product command. To enable Security Hub to receive findings from an integration, you need the product ARN. To obtain the ARNs for the available integrations, use the DescribeProducts operation. If you're using the AWS CLI, run the describe-products.

For example, the following AWS CLI command enables Security Hub CSPM to receive findings from the CrowdStrike Falcon integration. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub enable-import-findings-for product --product-arn
 "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/crowdstrike-falcon"
```

Disabling the flow of findings from a Security Hub CSPM integration

Choose your preferred method, and follow the steps to disable the flow of findings from an AWS Security Hub CSPM integration.

Security Hub CSPM console

To disable the flow of findings from an integration (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the Security Hub CSPM navigation pane, choose **Integrations**.
3. For integrations that send findings, the **Status** information indicates whether Security Hub CSPM is currently accepting findings from that integration.
4. Choose **Stop accepting findings**.

Security Hub CSPM API

Use the DisableImportFindingsForProduct operation. If you're using the AWS CLI, run the disable-import-findings-for-product command. To disable the flow of findings from an integration, you need the subscription ARN for the enabled integration. To obtain the subscription ARN, use the ListEnabledProductsForImport operation. If you're using the AWS CLI, run the list-enabled-products-for-import.

For example, the following AWS CLI command disables the flow of findings to Security Hub CSPM from the CrowdStrike Falcon integration. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub disable-import-findings-for-product --product-subscription-arn
 "arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/
crowdstrike-falcon"
```

Viewing findings from a Security Hub CSPM integration

When you start accepting findings from an AWS Security Hub CSPM integration, the **Integrations** page of the Security Hub CSPM console displays the **Status** of the integration as **Accepting findings**. To review a list of findings from the integration, choose **See findings**.

The findings list shows the active findings for the selected integration that have a workflow status of `NEW` or `NOTIFIED`.

If you enable cross-Region aggregation, then in the aggregation Region, the list includes findings from the aggregation Region and from linked Regions where the integration is enabled.

Security Hub does not automatically enable integrations based on the cross-Region aggregation configuration.

In other Regions, the finding list for an integration only contains findings from the current Region.

For information on how to configure cross-Region aggregation, see the section called "Aggregating data across Regions".

From the findings list, you can perform the following actions.

- Change the filters and grouping for the list
- View details for individual findings
- Update the workflow status of findings
- Send findings to custom actions

AWS service integrations with Security Hub CSPM

AWS Security Hub CSPM supports integrations with several other AWS services. These integrations can help you get a comprehensive view of security and compliance across your AWS environment.

Unless indicated otherwise below, AWS service integrations that send findings to Security Hub CSPM are activated automatically after you enable Security Hub CSPM and the other service. Integrations that receive Security Hub CSPM findings might require additional steps for activation. Review the information about each integration to learn more.

Some integrations aren't available in all AWS Regions. On the Security Hub CSPM console, an integration doesn't appear on the **Integrations** page if it isn't supported in the current Region. For a list of integrations that are available in the China Regions and AWS GovCloud (US) Regions, see Availability of integrations by Region.

Overview of AWS service integrations with Security Hub CSPM

The following table provides an overview of AWS services that send findings to Security Hub CSPM or receive findings from Security Hub CSPM.

Integrated AWS service	Direction	
AWS Config	Sends findings	

Integrated AWS service	Direction
AWS Firewall Manager	Sends findings
Amazon GuardDuty	Sends findings
AWS Health	Sends findings
AWS Identity and Access Management Access Analyzer	Sends findings
Amazon Inspector	Sends findings
AWS IoT Device Defender	Sends findings
Amazon Macie	Sends findings
Amazon Route 53 Resolver DNS Firewall	Sends findings
AWS Systems Manager Patch Manager	Sends findings
AWS Audit Manager	Receives findings
Amazon Q Developer in chat applications	Receives findings
Amazon Detective	Receives findings
Amazon Security Lake	Receives findings
AWS Systems Manager Explorer and OpsCenter	Receives and updates findings
AWS Trusted Advisor	Receives findings

AWS services that send findings to Security Hub CSPM

The following AWS services integrate with and can send findings to Security Hub CSPM. Security Hub CSPM converts the findings to the AWS Security Finding Format.

AWS Config (Sends findings)

AWS Config is a service that allows you to assess, audit, and evaluate the configurations of your AWS resources. AWS Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations.

By using the integration with AWS Config, you can see the results of AWS Config managed and custom rule evaluations as findings in Security Hub CSPM. These findings can be viewed alongside other Security Hub CSPM findings, providing a comprehensive overview of your security posture.

AWS Config uses Amazon EventBridge to send AWS Config rule evaluations to Security Hub CSPM. Security Hub CSPM transforms the rule evaluations into findings that follow the AWS Security Finding Format. Security Hub CSPM then enriches the findings on a best effort basis by getting more information about the impacted resources, such as the Amazon Resource Name (ARN), resource tags, and creation date.

For more information about this integration, see the following sections.

How AWS Config sends findings to Security Hub CSPM

All findings in Security Hub CSPM use the standard JSON format of ASFF. ASFF includes details about the origin of the finding, the affected resource, and the current status of the finding. AWS Config sends managed and custom rule evaluations to Security Hub CSPM via EventBridge. Security Hub CSPM transforms the rule evaluations into findings that follow ASFF and enriches the findings on a best effort basis.

Types of findings that AWS Config sends to Security Hub CSPM

After the integration is activated, AWS Config sends evaluations of all AWS Config managed rules and custom rules to Security Hub CSPM. Only evaluations that were performed after Security Hub CSPM was enabled are sent. For example, suppose that an AWS Config rule evaluation reveals five failed resources. If I enable Security Hub CSPM after that, and the rule then reveals a sixth failed resource, AWS Config sends only the sixth resource evaluation to Security Hub CSPM.

Evaluations from service-linked AWS Config rules, such as those used to run checks on Security Hub CSPM controls, are excluded.

Sending AWS Config findings to Security Hub CSPM

When the integration is activated, Security Hub CSPM will automatically assign the permissions necessary to receive findings from AWS Config. Security Hub CSPM uses service-to-service level permissions that provide you with a safe way to activate this integration and import findings from AWS Config via Amazon EventBridge.

Latency for sending findings

When AWS Config creates a new finding, you can usually view the finding in Security Hub CSPM within five minutes.

Retrying when Security Hub CSPM is not available

AWS Config sends findings to Security Hub CSPM on a best-effort basis through EventBridge. When an event isn't successfully delivered to Security Hub CSPM, EventBridge retries delivery for up to 24 hours or 185 times, whichever comes first.

Updating existing AWS Config findings in Security Hub CSPM

After AWS Config sends a finding to Security Hub CSPM, it can send updates to the same finding to Security Hub CSPM to reflect additional observations of the finding activity. Updates are only sent for `ComplianceChangeNotification` events. If no compliance change occurs, updates aren't sent to Security Hub CSPM. Security Hub CSPM deletes findings 90 days after the most recent update or 90 days after creation if no update occurs.

Security Hub CSPM doesn't archive findings that are sent from AWS Config even if you delete the associated resource.

Regions in which AWS Config findings exist

AWS Config findings occur on a Regional basis. AWS Config sends findings to Security Hub CSPM in the same Region or Regions where the findings occur.

Viewing AWS Config findings in Security Hub CSPM

To view your AWS Config findings, choose **Findings** from the Security Hub CSPM navigation pane. To filter the findings to display only AWS Config findings, choose **Product name** in the search bar drop down. Enter **Config**, and choose **Apply**.

Interpreting AWS Config finding names in Security Hub CSPM

Security Hub CSPM transforms AWS Config rule evaluations into findings that follow the AWS Security Finding Format (ASFF). AWS Config rule evaluations use a different event pattern compared to ASFF. The following table maps the AWS Config rule evaluation fields with their ASFF counterpart as they appear in Security Hub CSPM.

Config rule evaluation finding type	ASFF finding type	Hardcoded value
detail.awsAccountId	AwsAccountId	
detail.newEvaluationResult.resultRecordedTime	CreatedAt	
detail.newEvaluationResult.resultRecordedTime	UpdatedAt	
	ProductArn	"arn:<partition>:securityhub:<region>::product/aws/config"
	ProductName	"Config"
	CompanyName	"AWS"
	Region	"eu-central-1"
configRuleArn	GeneratorId, ProductFields	
detail.ConfigRuleARN/finding/hash	Id	
detail.configRuleName	Title, ProductFields	
detail.configRuleName	Description	"This finding is created for a resource compliance change for config rule: `${detail.ConfigRuleName}` "

Config rule evaluation finding type	ASFF finding type	Hardcoded value
Configuration Item "ARN" or Security Hub CSPM computed ARN	Resources[i].id	
detail.resourceType	Resources[i].Type	"AwsS3Bucket"
	Resources[i].Partition	"aws"
	Resources[i].Region	"eu-central-1"
Configuration Item "configuration"	Resources[i].Details	
	SchemaVersion	"2018-10-08"
	Severity.Label	See "Interpreting Severity Label" below
	Types	["Software and Configuration Checks"]
detail.newEvaluationResult.complianceType	Compliance.Status	"FAILED", "NOT_AVAILABLE", "PASSED", or "WARNING"
	Workflow.Status	"RESOLVED" if an AWS Config finding is generated with a Compliance.Status of "PASSED," or if the Compliance.Status changes from "FAILED" to "PASSED." Otherwise, Workflow.Status will be "NEW." You can change this value with the BatchUpdateFindings API operation.

Interpreting severity label

All findings from AWS Config rule evaluations have a default severity label of **MEDIUM** in the ASFF. You can update the severity label of a finding with the `BatchUpdateFindings` API operation.

Typical finding from AWS Config

Security Hub CSPM transforms AWS Config rule evaluations into findings that follow the ASFF. The following is an example of a typical finding from AWS Config in the ASFF.

> **Note**
>
> If the description is more than 1,024 characters, it will be truncated to 1,024 characters and will say "(truncated)" at the end.


```
{
 "SchemaVersion": "2018-10-08",
 "Id": "arn:aws:config:eu-central-1:123456789012:config-rule/config-rule-mburzq/
finding/45g070df80cb50b68fa6a43594kc6fda1e517932",
 "ProductArn": "arn:aws:securityhub:eu-central-1::product/aws/config",
 "ProductName": "Config",
 "CompanyName": "AWS",
 "Region": "eu-central-1",
 "GeneratorId": "arn:aws:config:eu-central-1:123456789012:config-rule/config-rule-
mburzq",
 "AwsAccountId": "123456789012",
 "Types": [
  "Software and Configuration Checks"
 ],
 "CreatedAt": "2022-04-15T05:00:37.181Z",
 "UpdatedAt": "2022-04-19T21:20:15.056Z",
 "Severity": {
  "Label": "MEDIUM",
  "Normalized": 40
 },
 "Title": "s3-bucket-level-public-access-prohibited-config-integration-demo",
 "Description": "This finding is created for a resource compliance change for config
 rule: s3-bucket-level-public-access-prohibited-config-integration-demo",
 "ProductFields": {
  "aws/securityhub/ProductName": "Config",
  "aws/securityhub/CompanyName": "AWS",
```

```
  "aws/securityhub/FindingId": "arn:aws:securityhub:eu-central-1::product/aws/
config/arn:aws:config:eu-central-1:123456789012:config-rule/config-rule-mburzq/
finding/46f070df80cd50b68fa6a43594dc5fda1e517902",
  "aws/config/ConfigRuleArn": "arn:aws:config:eu-central-1:123456789012:config-rule/
config-rule-mburzq",
  "aws/config/ConfigRuleName": "s3-bucket-level-public-access-prohibited-config-
integration-demo",
  "aws/config/ConfigComplianceType": "NON_COMPLIANT"
 },
 "Resources": [{
  "Type": "AwsS3Bucket",
  "Id": "arn:aws:s3:::amzn-s3-demo-bucket",
  "Partition": "aws",
  "Region": "eu-central-1",
  "Details": {
   "AwsS3Bucket": {
    "OwnerId": "4edbba300f1caa608fba2aad2c8fcfe30c32ca32777f64451eec4fb2a0f10d8c",
    "CreatedAt": "2022-04-15T04:32:53.000Z"
   }
  }
 }],
 "Compliance": {
  "Status": "FAILED"
 },
 "WorkflowState": "NEW",
 "Workflow": {
  "Status": "NEW"
 },
 "RecordState": "ACTIVE",
 "FindingProviderFields": {
  "Severity": {
   "Label": "MEDIUM"
  },
  "Types": [
   "Software and Configuration Checks"
  ]
 }
}
```

Enabling and configuring the integration

After you enable Security Hub CSPM, this integration is activated automatically. AWS Config immediately begins to send findings to Security Hub CSPM.

Stopping the publication of findings to Security Hub CSPM

To stop sending findings to Security Hub CSPM, you can use the Security Hub CSPM console or Security Hub CSPM API.

For instructions on stopping the flow of findings, see Enabling the flow of findings from a Security Hub CSPM integration.

AWS Firewall Manager (Sends findings)

Firewall Manager sends findings to Security Hub CSPM when a web application firewall (WAF) policy for resources or a web access control list (web ACL) rule is not in compliance. Firewall Manager also sends findings when AWS Shield Advanced is not protecting resources, or when an attack is identified.

After you enable Security Hub CSPM, this integration is automatically activated. Firewall Manager immediately begins to send findings to Security Hub CSPM.

To learn more about the integration, view the **Integrations** page in the Security Hub CSPM console.

To learn more about Firewall Manager, see the *AWS WAF Developer Guide*.

Amazon GuardDuty (Sends findings)

GuardDuty sends all of the finding types that it generates to Security Hub CSPM. Some finding types have prerequisites, enablement requirements, or Regional limitations. For more information, see GuardDuty finding types in the *Amazon GuardDuty User Guide*.

New findings from GuardDuty are sent to Security Hub CSPM within five minutes. Updates to findings are sent based on the **Updated findings** setting for Amazon EventBridge in GuardDuty settings.

When you generate GuardDuty sample findings using the GuardDuty **Settings** page, Security Hub CSPM receives the sample findings and omits the prefix `[Sample]` in the finding type. For example, the sample finding type in GuardDuty `[SAMPLE] Recon:IAMUser/ResourcePermissions` is displayed as `Recon:IAMUser/ResourcePermissions` in Security Hub CSPM.

After you enable Security Hub CSPM, this integration is automatically activated. GuardDuty immediately begins to send findings to Security Hub CSPM.

For more information about the GuardDuty integration, see Integrating with AWS Security Hub CSPM in the *Amazon GuardDuty User Guide*.

AWS Health (Sends findings)

AWS Health provides ongoing visibility into your resource performance and the availability of your AWS services and AWS accounts. You can use AWS Health events to learn how service and resource changes might affect your applications that run on AWS.

The integration with AWS Health does not use `BatchImportFindings`. Instead, AWS Health uses service-to-service event messaging to send findings to Security Hub CSPM.

For more information about the integration, see the following sections.

How AWS Health sends findings to Security Hub CSPM

In Security Hub CSPM, security issues are tracked as findings. Some findings come from issues that are detected by other AWS services or by third-party partners. Security Hub CSPM also has a set of rules that it uses to detect security issues and generate findings.

Security Hub CSPM provides tools to manage findings from across all of these sources. You can view and filter lists of findings and view details for a finding. See Reviewing finding details and history in Security Hub CSPM. You can also track the status of an investigation into a finding. See Setting the workflow status of findings in Security Hub CSPM.

All findings in Security Hub CSPM use a standard JSON format called the AWS Security Finding Format (ASFF). ASFF includes details about the source of the issue, the affected resources, and the current status of the finding.

AWS Health is one of the AWS services that sends findings to Security Hub CSPM.

Types of findings that AWS Health sends to Security Hub CSPM

After the integration is enabled, AWS Health sends findings that meet one or more of the listed specifications to Security Hub CSPM. Security Hub CSPM ingests the findings in the AWS Security Finding Format (ASFF).

- Findings that contain any of the following values for AWS service:
  - `RISK`
  - `ABUSE`
  - `ACM`

- CLOUDHSM
- CLOUDTRAIL
- CONFIG
- CONTROLTOWER
- DETECTIVE
- EVENTS
- GUARDDUTY
- IAM
- INSPECTOR
- KMS
- MACIE
- SES
- SECURITYHUB
- SHIELD
- SSO
- COGNITO
- IOTDEVICEDEFENDER
- NETWORKFIREWALL
- ROUTE53
- WAF
- FIREWALLMANAGER
- SECRETSMANAGER
- BACKUP
- AUDITMANAGER
- ARTIFACT
- CLOUDENDURE
- CODEGURU
- ORGANIZATIONS
- DIRECTORYSERVICE
- RESOURCEMANAGER

- CLOUDWATCH
- DRS
- INSPECTOR2
- RESILIENCEHUB

- Findings with the words security, abuse, or certificate in the AWS Health typeCode field
- Findings where the AWS Health service is risk or abuse

Sending AWS Health findings to Security Hub CSPM

When you choose to accept findings from AWS Health, Security Hub CSPM will automatically assign the permissions necessary to receive the findings from AWS Health. Security Hub CSPM uses service-to-service level permissions that provide you with a safe, easy way to enable this integration and import findings from AWS Health via Amazon EventBridge on your behalf. Choosing **Accept Findings** grants Security Hub CSPM permission to consume findings from AWS Health.

Latency for sending findings

When AWS Health creates a new finding, it is usually sent to Security Hub CSPM within five minutes.

Retrying when Security Hub CSPM is not available

AWS Health sends findings to Security Hub CSPM on a best-effort basis through EventBridge. When an event isn't successfully delivered to Security Hub CSPM, EventBridge retries sending the event for 24 hours.

Updating existing findings in Security Hub CSPM

After AWS Health sends a finding to Security Hub CSPM, it can send updates to the same finding to reflect additional observations of the finding activity to Security Hub CSPM.

Regions in which findings exist

For global events, AWS Health sends findings to Security Hub CSPM in us-east-1 (AWS partition), cn-northwest-1 (China partition), and gov-us-west-1 (GovCloud partition). AWS Health sends Region-specific events to Security Hub CSPM in the same Region or Regions where the events occur.

Viewing AWS Health findings in Security Hub CSPM

To view your AWS Health findings in Security Hub CSPM, choose **Findings** from the navigation panel. To filter the findings to display only AWS Health findings, choose **Health** from the **Product name** field.

Interpreting AWS Health finding names in Security Hub CSPM

AWS Health sends the findings to Security Hub CSPM using the AWS Security Finding Format (ASFF). AWS Health finding uses a different event pattern compared to Security Hub CSPM ASFF format. The table below details all the AWS Health finding fields with their ASFF counterpart as they appear in Security Hub CSPM.

Health finding type	ASFF finding type	Hardcoded value
account	AwsAccountId	
detail.startTime	CreatedAt	
detail.eventDescription.latestDescription	Description	
detail.eventTypeCode	GeneratorId	
detail.eventArn (including account) + hash of detail.startTime	Id	
"arn:aws:securityhub:<region>::product/aws/health"	ProductArn	
account or resourceId	Resources[i].id	
	Resources[i].Type	"Other"
	SchemaVersion	"2018-10-08"
	Severity.Label	See "Interpreting Severity Label" below

Health finding type	ASFF finding type	Hardcoded value
“AWS Health -" detail.eventTypeCode	Title	
-	Types	["Software and Configuration Checks"]
event.time	UpdatedAt	
URL of the event on Health console	SourceUrl	

Interpreting severity label

The severity label in the ASFF finding is determined using the following logic:

- Severity **CRITICAL** if:
  - The `service` field in the AWS Health finding has the value `Risk`
  - The `typeCode` field in the AWS Health finding has the value `AWS_S3_OPEN_ACCESS_BUCKET_NOTIFICATION`
  - The `typeCode` field in the AWS Health finding has the value `AWS_SHIELD_INTERNET_TRAFFIC_LIMITATIONS_PLACED_IN_RESPONSE_TO_DDOS_ATTACK`
  - The `typeCode` field in the AWS Health finding has the value `AWS_SHIELD_IS_RESPONDING_TO_A_DDOS_ATTACK_AGAINST_YOUR_AWS_RESOURCES`

  Severity **HIGH** if:
  - The `service` field in the AWS Health finding has the value `Abuse`
  - The `typeCode` field in the AWS Health finding contains the value `SECURITY_NOTIFICATION`
  - The `typeCode` field in the AWS Health finding contains the value `ABUSE_DETECTION`

  Severity **MEDIUM** if:
  - The `service` field in the finding is any of the following: `ACM`, `ARTIFACT`, `AUDITMANAGER`, `BACKUP`,`CLOUDENDURE`, `CLOUDHSM`, `CLOUDTRAIL`, `CLOUDWATCH`, `CODEGURGU`, `COGNITO`, `CONFIG`, `CONTROLTOWER`, `DETECTIVE`, `DIRECTORYSERVICE`, `DRS`, `EVENTS`, `FIREWALLMANAGER`, `GUARDDUTY`, `IAM`, `INSPECTOR`, `INSPECTOR2`, `IOTDEVICEDEFENDER`,

KMS, MACIE, NETWORKFIREWALL, ORGANIZATIONS, RESILIENCEHUB, RESOURCEMANAGER, ROUTE53, SECURITYHUB, SECRETSMANAGER, SES, SHIELD, SSO, or WAF

- The **typeCode** field in the AWS Health finding contains the value CERTIFICATE
- The **typeCode** field in the AWS Health finding contains the value END_OF_SUPPORT

Typical finding from AWS Health

AWS Health sends findings to Security Hub CSPM using the AWS Security Finding Format (ASFF). The following is an example of a typical finding from AWS Health.

> **Note**
>
> If the description is more than 1024 characters, it will be truncated to 1024 characters and will say *(truncated)* at the end.


```
{
            "SchemaVersion": "2018-10-08",
            "Id": "arn:aws:health:us-east-1:123456789012:event/SES/
AWS_SES_CMF_PENDING_TO_SUCCESS/
AWS_SES_CMF_PENDING_TO_SUCCESS_303388638044_33fe2115-8dad-40ce-
b533-78e29f49de96/101F7FBAEFC663977DA09CFF56A29236602834D2D361E6A8CA5140BFB3A69B30",
            "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/health",
            "GeneratorId": "AWS_SES_CMF_PENDING_TO_SUCCESS",
            "AwsAccountId": "123456789012",
            "Types": [
                "Software and Configuration Checks"
            ],
            "CreatedAt": "2022-01-07T16:34:04.000Z",
            "UpdatedAt": "2022-01-07T19:17:43.000Z",
            "Severity": {
                "Label": "MEDIUM",
                "Normalized": 40
            },
            "Title": "AWS Health - AWS_SES_CMF_PENDING_TO_SUCCESS",
            "Description": "Congratulations! Amazon SES has successfully detected the
 MX record required to use 4557227d-9257-4e49-8d5b-18a99ced4be9.cmf.pinpoint.sysmon-
iad.adzel.com as a custom MAIL FROM domain for verified identity cmf.pinpoint.sysmon-
iad.adzel.com in AWS Region US East (N. Virginia).\\n\\nYou can now use this MAIL
 FROM domain with cmf.pinpoint.sysmon-iad.adzel.com and any other verified identity
```

```
 that is configured to use it. For information about how to configure a verified
 identity to use a custom MAIL FROM domain, see http://docs.aws.amazon.com/ses/latest/
DeveloperGuide/mail-from-set.html .\\n\\nPlease note that this email only applies to
 AWS Region US East (N. Virginia).",
            "SourceUrl": "https://phd.aws.amazon.com/phd/home#/event-log?
eventID=arn:aws:health:us-east-1::event/SES/AWS_SES_CMF_PENDING_TO_SUCCESS/
AWS_SES_CMF_PENDING_TO_SUCCESS_303388638044_33fe2115-8dad-40ce-b533-78e29f49de96",
            "ProductFields": {
                "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/
aws/health/arn:aws:health:us-east-1::event/SES/AWS_SES_CMF_PENDING_TO_SUCCESS/
AWS_SES_CMF_PENDING_TO_SUCCESS_303388638044_33fe2115-8dad-40ce-b533-78e29f49de96",
                "aws/securityhub/ProductName": "Health",
                "aws/securityhub/CompanyName": "AWS"
            },
            "Resources": [
                {
                    "Type": "Other",
                    "Id": "4557227d-9257-4e49-8d5b-18a99ced4be9.cmf.pinpoint.sysmon-
iad.adzel.com"
                }
            ],
            "WorkflowState": "NEW",
            "Workflow": {
                "Status": "NEW"
            },
            "RecordState": "ACTIVE",
            "FindingProviderFields": {
                "Severity": {
                    "Label": "MEDIUM"
                },
                "Types": [
                    "Software and Configuration Checks"
                ]
            }
        }
    ]
}
```

Enabling and configuring the integration

After you enable Security Hub CSPM, this integration is automatically activated. AWS Health immediately begins to send findings to Security Hub CSPM.

Stopping the publication of findings to Security Hub CSPM

To stop sending findings to Security Hub CSPM, you can use the Security Hub CSPM console or Security Hub CSPM API.

For instructions on stopping the flow of findings, see Enabling the flow of findings from a Security Hub CSPM integration.

AWS Identity and Access Management Access Analyzer (Sends findings)

With IAM Access Analyzer, all findings are sent to Security Hub CSPM.

IAM Access Analyzer uses logic-based reasoning to analyze resource-based policies that are applied to supported resources in your account. IAM Access Analyzer generates a finding when it detects a policy statement that lets an external principal access a resource in your account.

In IAM Access Analyzer, only the administrator account can see findings for analyzers that apply to an organization. For organization analyzers, the `AwsAccountId` ASFF field reflects the administrator account ID. Under `ProductFields`, the `ResourceOwnerAccount` field indicates the account in which the finding was discovered. If you enable analyzers individually for each account, Security Hub CSPM generates multiple findings, one that identifies the administrator account ID and one that identifies the resource account ID.

For more information, see Integration with AWS Security Hub CSPM in the *IAM User Guide*.

Amazon Inspector (Sends findings)

Amazon Inspector is a vulnerability management service that continuously scans your AWS workloads for vulnerabilities. Amazon Inspector automatically discovers and scans Amazon EC2 instances and container images that reside in the Amazon Elastic Container Registry. The scan looks for software vulnerabilities and unintended network exposure.

After you enable Security Hub CSPM, this integration is automatically activated. Amazon Inspector immediately begins to send all of the findings that it generates to Security Hub CSPM.

For more information about the integration, see Integration with AWS Security Hub CSPM in the *Amazon Inspector User Guide*.

Security Hub CSPM can also receive findings from Amazon Inspector Classic. Amazon Inspector Classic sends findings to Security Hub CSPM that are generated through assessment runs for all supported rules packages.

For more information about the integration, see Integration with AWS Security Hub CSPM in the *Amazon Inspector Classic User Guide*.

Findings for Amazon Inspector and Amazon Inspector Classic use the same product ARN. Amazon Inspector findings have the following entry in `ProductFields`:

```
"aws/inspector/ProductVersion": "2",
```

> **Note**
>
> Security findings generated by Amazon Inspector Code Security are not available for this integration. However, you can access these particular findings in the Amazon Inspector console and through the Amazon Inspector API.

AWS IoT Device Defender (Sends findings)

AWS IoT Device Defender is a security service that audits the configuration of your IoT devices, monitors connected devices to detect abnormal behavior, and helps mitigate security risks.

After enabling both AWS IoT Device Defender and Security Hub CSPM, visit the Integrations page of the Security Hub CSPM console, and choose **Accept findings** for Audit, Detect, or both. AWS IoT Device Defender Audit and Detect begin to send all findings to Security Hub CSPM.

AWS IoT Device Defender Audit sends check summaries to Security Hub CSPM, which contain general information for a specific audit check type and audit task. AWS IoT Device Defender Detect sends violation findings for machine learning (ML), statistical, and static behaviors to Security Hub CSPM. Audit also sends finding updates to Security Hub CSPM.

For more information about this integration, see Integration with AWS Security Hub CSPM in the *AWS IoT Developer Guide*.

Amazon Macie (Sends findings)

Amazon Macie is a data security service that discovers sensitive data by using machine learning and pattern matching, provides visibility into data security risks, and enables automated protection against those risks. A finding from Macie can indicate that a potential policy violation or sensitive data exists in your Amazon S3 data estate.

After you enable Security Hub CSPM, Macie automatically starts sending policy findings to Security Hub CSPM. You can configure the integration to also send sensitive data findings to Security Hub CSPM.

In Security Hub CSPM, the finding type for a policy or sensitive data finding is changed to a value that is compatible with ASFF. For example, the `Policy:IAMUser/S3BucketPublic` finding type in Macie is displayed as `Effects/Data Exposure/Policy:IAMUser-S3BucketPublic` in Security Hub CSPM.

Macie also sends generated sample findings to Security Hub CSPM. For sample findings, the name of the affected resource is `macie-sample-finding-bucket` and the value for the `Sample` field is `true`.

For more information, see Evaluating Macie findings with Security Hub in the *Amazon Macie User Guide*.

Amazon Route 53 Resolver DNS Firewall (Sends findings)

With Amazon Route 53 Resolver DNS Firewall, you can filter and regulate outbound DNS traffic for your virtual private cloud (VPC). You do this by creating reusable collections of filtering rules in DNS Firewall rule groups, associating the rule groups with your VPC, and then monitoring activity in DNS Firewall logs and metrics. Based on the activity, you can adjust DNS Firewall behavior. DNS Firewall is a feature of Route 53 Resolver.

Route 53 Resolver DNS Firewall can send several types of findings to Security Hub CSPM:

- Findings related to queries blocked or alerted on for domains associated with AWS Managed Domain Lists, which are domain lists that AWS manages.
- Findings related to queries blocked or alerted on for domains associated with a custom domain list that you define.
- Findings related to queries blocked or alerted on by DNS Firewall Advanced, which is a Route 53 Resolver feature that can detect queries associated with advanced DNS threats such as Domain Generation Algorithms (DGAs) and DNS Tunneling.

After you enable Security Hub CSPM and Route 53 Resolver DNS Firewall, DNS Firewall automatically starts sending findings for AWS Managed Domain Lists and DNS Firewall Advanced to Security Hub CSPM. To also send findings for a custom domain list to Security Hub CSPM, manually enable the integration in Security Hub CSPM.

In Security Hub CSPM, all findings from Route 53 Resolver DNS Firewall have the following type: `TTPs/Impact/Impact:Runtime-MaliciousDomainRequest.Reputation`.

For more information, see Sending findings from Route 53 Resolver DNS Firewall to Security Hub in the *Amazon Route 53 Developer Guide*.

AWS Systems Manager Patch Manager (Sends findings)

AWS Systems Manager Patch Manager sends findings to Security Hub CSPM when instances in a customer's fleet go out of compliance with their patch compliance standard.

Patch Manager automates the process of patching managed instances with both security related and other types of updates.

After you enable Security Hub CSPM, this integration is automatically activated. Systems Manager Patch Manager immediately begins to send findings to Security Hub CSPM.

For more information about using Patch Manager, see AWS Systems Manager Patch Manager in the *AWS Systems Manager User Guide*.

AWS services that receive findings from Security Hub CSPM

The following AWS services are integrated with Security Hub CSPM and receive findings from Security Hub CSPM. Where noted, the integrated service may also update findings. In this case, finding updates that you make in the integrated service will also be reflected in Security Hub CSPM.

AWS Audit Manager (Receives findings)

AWS Audit Manager receives findings from Security Hub CSPM. These findings help Audit Manager users to prepare for audits.

To learn more about Audit Manager, see the *AWS Audit Manager User Guide*. AWS Security Hub CSPM checks supported by AWS Audit Manager lists the controls for which Security Hub CSPM sends findings to Audit Manager.

Amazon Q Developer in chat applications (Receives findings)

Amazon Q Developer in chat applications is an interactive agent that helps you to monitor and interact with your AWS resources in your Slack channels and Amazon Chime chat rooms.

Amazon Q Developer in chat applications receives findings from Security Hub CSPM.

To learn more about the Amazon Q Developer in chat applications integration with Security Hub CSPM, see the Security Hub CSPM integration overview in the *Amazon Q Developer in chat applications Administrator Guide*.

Amazon Detective (Receives findings)

Detective automatically collects log data from your AWS resources and uses machine learning, statistical analysis, and graph theory to help you visualize and conduct faster and more efficient security investigations.

The Security Hub CSPM integration with Detective allows you to pivot from Amazon GuardDuty findings in Security Hub CSPM into Detective. You can then use the Detective tools and visualizations to investigate them. The integration does not require any additional configuration in Security Hub CSPM or Detective.

For findings received from other AWS services, the finding details panel on the Security Hub CSPM console includes an **Investigate in Detective** subsection. That subsection contains a link to Detective where you can further investigate the security issue that the finding flagged. You can also build a behavior graph in Detective based on Security Hub CSPM findings to conduct more effective investigations. For more information, see AWS security findings in the *Amazon Detective Administration Guide*.

If cross-Region aggregation is enabled, then when you pivot from the aggregation Region, Detective opens in the Region where the finding originated.

If a link does not work, then for troubleshooting advice, see Troubleshooting the pivot.

Amazon Security Lake (Receives findings)

Security Lake is a fully-managed security data lake service. You can use Security Lake to automatically centralize security data from cloud, on-premises, and custom sources into a data lake that's stored in your account. Subscribers can consume data from Security Lake for investigative and analytics use cases.

To activate this integration, you must enable both services and add Security Hub CSPM as a source in the Security Lake console, Security Lake API, or AWS CLI. Once you complete these steps, Security Hub CSPM begins to send all findings to Security Lake.

Security Lake automatically normalizes Security Hub CSPM findings and converts them to a standardized open-source schema called Open Cybersecurity Schema Framework (OCSF). In Security Lake, you can add one or more subscribers to consume Security Hub CSPM findings.

For more information about this integration, including instructions on adding Security Hub CSPM as a source and creating subscribers, see Integration with AWS Security Hub CSPM in the *Amazon Security Lake User Guide*.

AWS Systems Manager Explorer and OpsCenter (Receives and updates findings)

AWS Systems Manager Explorer and OpsCenter receive findings from Security Hub CSPM, and update those findings in Security Hub CSPM.

Explorer provides you with a customizable dashboard, providing key insights and analysis into the operational health and performance of your AWS environment.

OpsCenter provides you with a central location to view, investigate, and resolve operational work items.

For more information about Explorer and OpsCenter, see Operations management in the *AWS Systems Manager User Guide*.

AWS Trusted Advisor (Receives findings)

Trusted Advisor draws upon best practices learned from serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment, and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps.

When you enable both Trusted Advisor and Security Hub CSPM, the integration is updated automatically.

Security Hub CSPM sends the results of its AWS Foundational Security Best Practices checks to Trusted Advisor.

For more information about the Security Hub CSPM integration with Trusted Advisor, see Viewing AWS Security Hub CSPM controls in AWS Trusted Advisor in the *AWS Support User Guide*.

Third-party product integrations with Security Hub CSPM

AWS Security Hub CSPM integrates with multiple third-party partner products. An integration can perform one or more of the following actions:

- Send findings that it generates to Security Hub CSPM

- Receive findings from Security Hub CSPM
- Update findings in Security Hub CSPM

Integrations that send findings to Security Hub CSPM have an Amazon Resource Name (ARN).

An integration might not be available in all AWS Regions. If an integration isn't supported in the Region that you are currently signed in to on the Security Hub CSPM console, it doesn't appear on the **Integrations** page of the console. For a list of integrations that are available in the China Regions and AWS GovCloud (US) Regions, see Availability of integrations by Region.

If you have a security solution and are interested in becoming a Security Hub CSPM partner, send email to <securityhub-partners@amazon.com>. For more information, see the Partner Integration Guide.

Overview of third-party integrations with Security Hub CSPM

The following table provides an overview of the third-party integrations that can send findings to Security Hub CSPM or receive findings from Security Hub CSPM.

Integration	Direction	ARN (if applicable)
3CORESec – 3CORESec NTA	Sends findings	arn:aws:securityhub: *<REGION>*::product/3coresec/3coresec
Alert Logic – SIEMless Threat Management	Sends findings	arn:aws:securityhub: *<REGION>*:733251395267:product/alertlogic/althreatmanagement
Aqua Security – Aqua Cloud Native Security Platform	Sends findings	arn:aws:securityhub: *<REGION>*::product/aquasecurity/aquasecurity
Aqua Security – Kube-bench	Sends findings	arn:aws:securityhub: *<REGION>*::product/

Integration	Direction	ARN (if applicable)
		aqua-security/kube-bench
Armor – Armor Anywhere	Sends findings	arn:aws:securityhub: *<REGION>*:679703615338:product/armordefense/armoranywhere
AttackIQ – AttackIQ	Sends findings	arn:aws:securityhub: *<REGION>*::product/attackiq/attackiq-platform
Barracuda Networks – Cloud Security Guardian	Sends findings	arn:aws:securityhub: *<REGION>*:151784055945:product/barracuda/cloudsecurityguardian
BigID – BigID Enterprise	Sends findings	arn:aws:securityhub: *<REGION>*::product/bigid/bigid-enterprise
Blue Hexagon – Blue Hexagon forAWS	Sends findings	arn:aws:securityhub: *<REGION>*::product/blue-hexagon/blue-hexagon-for-aws
Check Point – CloudGuard IaaS	Sends findings	arn:aws:securityhub: *<REGION>*:758245563457:product/checkpoint/cloudguard-iaas

Integration	Direction	ARN (if applicable)
Check Point – CloudGuard Posture Management	Sends findings	arn:aws:securityhub: *<REGION>*:634729597623:product/checkpoint/dome9-arc
Claroty – xDome	Sends findings	arn:aws:securityhub: *<REGION>*::product/claroty/xdome
Cloud Storage Security – Antivirus for Amazon S3	Sends findings	arn:aws:securityhub: *<REGION>*::product/cloud-storage-security/antivirus-for-amazon-s3
Contrast Security	Sends findings	arn:aws:securityhub: *<REGION>*::product/contrast-security/security-assess
CrowdStrike – CrowdStrike Falcon	Sends findings	arn:aws:securityhub: *<REGION>*:517716713836:product/crowdstrike/crowdstrike-falcon
CyberArk – Privileged Threat Analytics	Sends findings	arn:aws:securityhub: *<REGION>*:749430749651:product/cyberark/cyberark-pta
Data Theorem – Data Theorem	Sends findings	arn:aws:securityhub: *<REGION>*::product/data-theorem/api-cloud-web-secure

Integration	Direction	ARN (if applicable)
Drata	Sends findings	arn:aws:securityhub: *<REGION>*::product/drata/drata-integration
Forcepoint – Forcepoint CASB	Sends findings	arn:aws:securityhub: *<REGION>*:365761988620:product/forcepoint/forcepoint-casb
Forcepoint – Forcepoint Cloud Security Gateway	Sends findings	arn:aws:securityhub: *<REGION>*::product/forcepoint/forcepoint-cloud-security-gateway
Forcepoint – Forcepoint DLP	Sends findings	arn:aws:securityhub: *<REGION>*:365761988620:product/forcepoint/forcepoint-dlp
Forcepoint – Forcepoint NGFW	Sends findings	arn:aws:securityhub: *<REGION>*:365761988620:product/forcepoint/forcepoint-ngfw
Fugue – Fugue	Sends findings	arn:aws:securityhub: *<REGION>*::product/fugue/fugue

Integration	Direction	ARN (if applicable)
Guardicore – Centra 4.0	Sends findings	arn:aws:securityhub: *<REGION>*::product/guardicore/guardicore
HackerOne – Vulnerability Intelligence	Sends findings	arn:aws:securityhub: *<REGION>*::product/hackerone/vulnerability-intelligence
JFrog – Xray	Sends findings	arn:aws:securityhub: *<REGION>*::product/jfrog/jfrog-xray
Juniper Networks – vSRX Next Generation Firewall	Sends findings	arn:aws:securityhub: *<REGION>*::product/juniper-networks/vsrx-next-generation-firewall
k9 Security – Access Analyzer	Sends findings	arn:aws:securityhub: *<REGION>*::product/k9-security/access-analyzer
Lacework – Lacework	Sends findings	arn:aws:securityhub: *<REGION>*::product/lacework/lacework
McAfee – MVISION Cloud Native Application Protection Platform (CNAPP)	Sends findings	arn:aws:securityhub: *<REGION>*::product/mcafee-skyhigh/mcafee-mvision-cloud-aws

Integration	Direction	ARN (if applicable)
NETSCOUT – NETSCOUT Cyber Investigator	Sends findings	arn:aws:securityhub:us-east-1::product/netscout/netscout-cyber-investigator
Orca Cloud Security Platform	Sends findings	arn:aws:securityhub: *<REGION>*::product/orca-security/orca-security
Palo Alto Networks – Prisma Cloud Compute	Sends findings	arn:aws:securityhub: *<REGION>*:496947949261:product/twistlock/twistlock-enterprise
Palo Alto Networks – Prisma Cloud Enterprise	Sends findings	arn:aws:securityhub: *<REGION>*:188619942792:product/paloaltonetworks/redlock
Plerion – Cloud Security Platform	Sends findings	arn:aws:securityhub: *<REGION>*::product/plerion/cloud-security-platform
Prowler – Prowler	Sends findings	arn:aws:securityhub: *<REGION>*::product/prowler/prowler
Qualys – Vulnerability Management	Sends findings	arn:aws:securityhub: *<REGION>*:805950163170:product/qualys/qualys-vm

Integration	Direction	ARN (if applicable)
Rapid7 – InsightVM	Sends findings	arn:aws:securityhub: *<REGION>*:336818582268:product/rapid7/insightvm
SecureCloudDB – SecureCloudDB	Sends findings	arn:aws:securityhub: *<REGION>*::product/securecloudd b/securecloudd b
SentinelOne – SentinelOne	Sends findings	arn:aws:securityhub: *<REGION>*::product/sentinelone/endpoint-protection
Snyk	Sends findings	arn:aws:securityhub: *<region>*::product/snyk/snyk
Sonrai Security – Sonrai Dig	Sends findings	arn:aws:securityhub: *<REGION>*::product/sonrai-security/sonrai-dig
Sophos – Server Protection	Sends findings	arn:aws:securityhub: *<REGION>*:062897671886:product/sophos/sophos-server-protection
StackRox – StackRox Kubernetes Security	Sends findings	arn:aws:securityhub: *<REGION>*::product/stackrox/kubernetes-security

Integration	Direction	ARN (if applicable)
Sumo Logic – Machine Data Analytics	Sends findings	arn:aws:securityhub: *<REGION>*:956882708938:product/sumologicinc/sumologic-mda
Symantec – Cloud Workload Protection	Sends findings	arn:aws:securityhub: *<REGION>*:754237914691:product/symantec-corp/symantec-cwp
Tenable – Tenable.io	Sends findings	arn:aws:securityhub: *<REGION>*:422820575223:product/tenable/tenable-io
Trend Micro – Cloud One	Sends findings	arn:aws:securityhub: *<REGION>*::product/trend-micro/cloud-one
Vectra – Cognito Detect	Sends findings	arn:aws:securityhub: *<REGION>*:978576646331:product/vectra-ai/cognito-detect
Wiz	Sends findings	arn:aws:securityhub: *<REGION>*::product/wiz-security/wiz-security
Atlassian - Jira Service Management	Receives and updates findings	Not applicable

Integration	Direction	ARN (if applicable)
Atlassian - Jira Service Management Cloud	Receives and updates findings	Not applicable
Atlassian – Opsgenie	Receives findings	Not applicable
Dynatrace	Receives findings	Not applicable
Fortinet – FortiCNP	Receives findings	Not applicable
IBM – QRadar	Receives findings	Not applicable
Logz.io Cloud SIEM	Receives findings	Not applicable
MetricStream	Receives findings	Not applicable
MicroFocus – MicroFocus Arcsight	Receives findings	Not applicable
New Relic Vulnerability Management	Receives findings	Not applicable
PagerDuty – PagerDuty	Receives findings	Not applicable
Palo Alto Networks – Cortex XSOAR	Receives findings	Not applicable
Palo Alto Networks – VM-Series	Receives findings	Not applicable
Rackspace Technology – Cloud Native Security	Receives findings	Not applicable
Rapid7 – InsightConnect	Receives findings	Not applicable
RSA – RSA Archer	Receives findings	Not applicable
ServiceNow – ITSM	Receives and updates findings	Not applicable
Slack – Slack	Receives findings	Not applicable

Integration	Direction	ARN (if applicable)
Splunk – Splunk Enterprise	Receives findings	Not applicable
Splunk – Splunk Phantom	Receives findings	Not applicable
ThreatModeler	Receives findings	Not applicable
Trellix – Trellix Helix	Receives findings	Not applicable
Caveonix – Caveonix Cloud	Sends and receives findings	`arn:aws:securityhub: <REGION>::product/caveonix/caveonix-cloud`
Cloud Custodian – Cloud Custodian	Sends and receives findings	`arn:aws:securityhub: <REGION>::product/cloud-custodian/cloud-custodian`
DisruptOps, Inc. – DisruptOPS	Sends and receives findings	`arn:aws:securityhub: <REGION>::product/disruptops-inc/disruptops`
Kion	Sends and receives findings	`arn:aws:securityhub: <REGION>::product/cloudtamerio/cloudtamerio`
Turbot – Turbot	Sends and receives findings	`arn:aws:securityhub: <REGION>:453761072151:product/turbot/turbot`

Third-party integrations that send findings to Security Hub CSPM

The following third-party partner product integrations can send findings to Security Hub CSPM. Security Hub CSPM transforms the findings into the AWS Security Finding Format.

3CORESec – 3CORESec NTA

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/3coresec/3coresec`

3CORESec provides managed detection services for both on-premises and AWS systems. Their integration with Security Hub CSPM allows visibility into threats such as malware, privilege escalation, lateral movement, and improper network segmentation.

Product link

Partner documentation

Alert Logic – SIEMless Threat Management

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:733251395267:product/alertlogic/althreatmanagement`

Get the right level of coverage: vulnerability and asset visibility, threat detection and incident management, AWS WAF, and assigned SOC analyst options.

Product link

Partner documentation

Aqua Security – Aqua Cloud Native Security Platform

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/aquasecurity/aquasecurity`

Aqua Cloud Native Security Platform (CSP) provides full lifecycle security for container-based and serverless applications, from your CI/CD pipeline to runtime production environments.

Product link

Partner documentation

Aqua Security – Kube-bench

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/aqua-security/kube-bench`

Kube-bench is an open-source tool that runs the Center for Internet Security (CIS) Kubernetes Benchmark against your environment.

Product link

Partner documentation

Armor – Armor Anywhere

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:679703615338:product/armordefense/armoranywhere`

Armor Anywhere delivers managed security and compliance for AWS.

Product link

Partner documentation

AttackIQ – AttackIQ

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/attackiq/attackiq-platform`

AttackIQ Platform emulates real adversarial behavior aligned with the MITRE ATT&CK Framework to help validate and improve your overall security posture.

Product link

Partner documentation

Barracuda Networks – Cloud Security Guardian

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:151784055945:product/barracuda/cloudsecurityguardian`

Barracuda Cloud Security Sentry helps organizations stay secure while building applications in, and moving workloads to, the public cloud.

AWS Marketplace link

Product link

BigID – BigID Enterprise

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/bigid/bigid-enterprise`

The BigID Enterprise Privacy Management Platform helps companies manage and protect sensitive data (PII) across all their systems.

Product link

Partner documentation

Blue Hexagon – Blue Hexagon for AWS

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/blue-hexagon/blue-hexagon-for-aws`

Blue Hexagon is a real time threat detection platform. It uses deep learning principles to detect known and unknown threats, including malware and network anomalies.

AWS Marketplace link

Partner documentation

Check Point – CloudGuard IaaS

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:758245563457:product/checkpoint/cloudguard-iaas`

Check Point CloudGuard easily extends comprehensive threat prevention security to AWS while protecting assets in the cloud.

Product link

Partner documentation

Check Point – CloudGuard Posture Management

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:634729597623:product/checkpoint/dome9-arc`

A SaaS platform that delivers verifiable cloud network security, advanced IAM protection, and comprehensive compliance and governance.

Product link

Partner documentation

Claroty – xDome

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/claroty/xdome`

Claroty xDome helps organizations secure their cyber-physical systems across the Extended Internet of Things (XIoT) within industrial (OT), healthcare (IoMT), and enterprise (IoT) environments.

Product link

Partner documentation

Cloud Storage Security – Antivirus for Amazon S3

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/cloud-storage-security/antivirus-for-amazon-s3`

Cloud Storage Security provides cloud native anti-malware and antivirus scanning for Amazon S3 objects.

Antivirus for Amazon S3 offers real time and scheduled scans of objects and files in Amazon S3 for malware and threats. It provides visibility and remediation for problem and infected files.

Product link

Partner documentation

Contrast Security – Contrast Assess

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/contrast-security/security-assess`

Contrast Security Contrast Assess is an IAST tool that offers real-time vulnerability detection in web apps, APIs, and microservices. Contrast Assess integrates with Security Hub CSPM to help provide centralized visibility and response for all your workloads.

Product link

Partner documentation

CrowdStrike – CrowdStrike Falcon

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:517716713836:product/crowdstrike/crowdstrike-falcon`

The CrowdStrike Falcon single, lightweight sensor unifies next-generation antivirus, endpoint detection and response, and 24/7 managed hunting through the cloud.

AWS Marketplace link

Partner documentation

CyberArk – Privileged Threat Analytics

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:749430749651:product/cyberark/cyberark-pta`

Privileged Threat Analytics collect, detect, alert, and respond to high-risk activity and behavior of privileged accounts to contain in-progress attacks.

Product link

Partner documentation

Data Theorem – Data Theorem

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/data-theorem/api-cloud-web-secure`

Data Theorem continuously scans web applications, APIs, and cloud resources in search of security flaws and data privacy gaps to prevent AppSec data breaches.

Product link

Partner documentation

Drata

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/drata/drata-integration`

Drata is a compliance automation platform that helps you achieve and maintain compliance with various frameworks, such as SOC2, ISO, and GDPR. The integration between Drata and Security Hub CSPM helps you centralize your security findings in one location.

AWS Marketplace link

Partner documentation

Forcepoint – Forcepoint CASB

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:365761988620:product/forcepoint/forcepoint-casb`

Forcepoint CASB allows you to discover cloud application use, analyze risk, and enforce appropriate controls for SaaS and custom applications.

Product link

Partner documentation

Forcepoint – Forcepoint Cloud Security Gateway

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/forcepoint/forcepoint-cloud-security-gateway`

Forcepoint Cloud Security Gateway is a converged cloud security service that provides visibility, control, and threat protection for users and data, wherever they are.

Product link

Partner documentation

Forcepoint – Forcepoint DLP

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:365761988620:product/forcepoint/forcepoint-dlp`

Forcepoint DLP addresses human-centric risk with visibility and control everywhere your people work and everywhere your data resides.

Product link

Partner documentation

Forcepoint – Forcepoint NGFW

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:365761988620:product/forcepoint/forcepoint-ngfw`

Forcepoint NGFW lets you connect your AWS environment into your enterprise network with the scalability, protection, and insights needed to manage your network and respond to threats.

Product link

Partner documentation

Fugue – Fugue

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/fugue/fugue`

Fugue is an agent-less, scalable cloud-native platform that automates the continuous validation of infrastructure-as-code and cloud runtime environments using the same policies.

Product link

Partner documentation

Guardicore – Centra 4.0

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/guardicore/guardicore`

Guardicore Centra provides flow visualization, micro-segmentation, and breach detection for workloads in modern data centers and clouds.

Product link

Partner documentation

HackerOne – Vulnerability Intelligence

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/hackerone/vulnerability-intelligence`

The HackerOne platform partners with the global hacker community to uncover the most relevant security issues. Vulnerability Intelligence enables your organization to go beyond automated scanning. It shares vulnerabilities that HackerOne ethical hackers have validated and provided steps to reproduce.

AWS marketplace link

Partner documentation

JFrog – Xray

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/jfrog/jfrog-xray`

JFrog Xray is a universal application security Software Composition Analysis (SCA) tool that continuously scans binaries for license compliance and security vulnerabilities so that you can run a secure software supply chain.

AWS Marketplace link

Partner documentation

Juniper Networks – vSRX Next Generation Firewall

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/juniper-networks/vsrx-next-generation-firewall`

Juniper Networks' vSRX Virtual Next Generation Firewall delivers a complete cloud-based virtual firewall with advanced security, secure SD-WAN, robust networking, and built-in automation.

AWS Marketplace link

Partner documentation

Product link

k9 Security – Access Analyzer

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/k9-security/access-analyzer`

k9 Security notifies you when important access changes occur in your AWS Identity and Access Management account. With k9 Security, you can understand the access that users and IAM roles have to critical AWS services and your data.

k9 Security is built for continuous delivery, allowing you to operationalize IAM with actionable access audits and simple policy automation for AWS CDK and Terraform.

Product link

Partner documentation

Lacework – Lacework

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/lacework/lacework`

Lacework is the data-driven security platform for the cloud. The Lacework Cloud Security Platform automates cloud security at scale so you can innovate with speed and safety.

Product link

Partner documentation

McAfee – MVISION Cloud Native Application Protection Platform (CNAPP)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/mcafee-skyhigh/mcafee-mvision-cloud-aws`

McAfee MVISION Cloud Native Application Protection Platform (CNAPP) offers Cloud Security Posture Management (CSPM) and Cloud Workload Protection Platform (CWPP) for your AWS environment.

Product link

Partner documentation

NETSCOUT – NETSCOUT Cyber Investigator

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/netscout/netscout-cyber-investigator`

NETSCOUT Cyber Investigator is an enterprise-wide network threat, risk investigation, and forensic analysis platform that helps to reduce the impact of cyber threats on businesses.

Product link

Partner documentation

Orca Cloud Security Platform

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/orca-security/orca-security`

The Orca Cloud Security Platform identifies, prioritizes, and remediates risks and compliance issues across your entire cloud estate. Orca's agentless-first, AI-driven platform offers comprehensive coverage detecting vulnerabilities, misconfigurations, lateral movement, API risks, sensitive data, anomalous events and behaviors, and overly permissive identities.

Orca integrates with Security Hub CSPM to bring deep cloud security telemetry into Security Hub CSPM. Orca, using its SideScanning technology, prioritizes risk across cloud infrastructure, workloads, applications, data, APIs, identities, and more.

Product link

Partner documentation

Palo Alto Networks – Prisma Cloud Compute

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:496947949261:product/twistlock/twistlock-enterprise`

Prisma Cloud Compute is a cloud native cybersecurity platform that protects VMs, containers, and serverless platforms.

Product link

Partner documentation

Palo Alto Networks – Prisma Cloud Enterprise

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:188619942792:product/paloaltonetworks/redlock`

Protects your AWS deployment with cloud security analytics, advanced threat detection, and compliance monitoring.

Product link

Partner documentation

Plerion – Cloud Security Platform

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/plerion/cloud-security-platform`

Plerion is a Cloud Security Platform with a unique threat-led, risk-driven approach that offers preventative, detective, and corrective action across your workloads. The integration between Plerion and Security Hub CSPM allows customers to centralize and act upon their security findings in one place.

AWS Marketplace link

Partner documentation

Prowler – Prowler

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/prowler/prowler`

Prowler is an open source security tool to perform AWS checks related to security best practices, hardening, and continuous monitoring.

Product link

Partner documentation

Qualys – Vulnerability Management

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:805950163170:product/qualys/qualys-vm`

Qualys Vulnerability Management (VM) continuously scans and identifies vulnerabilities, protecting your assets.

Product link

Partner documentation

Rapid7 – InsightVM

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:336818582268:product/rapid7/insightvm`

Rapid7 InsightVM provides vulnerability management for modern environments, allowing you to efficiently find, prioritize, and remediate vulnerabilities.

Product link

Partner documentation

SecureCloudDB – SecureCloudDB

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/securecloudb/securecloudb`

SecureCloudDB is a cloud native database security tool that provides comprehensive visibility of internal and external security postures and activity. It flags security violations and provides remediation on exploitable database vulnerabilities.

Product link

Partner documentation

SentinelOne – SentinelOne

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/sentinelone/endpoint-protection`

SentinelOne is an autonomous extended detection and response (XDR) platform encompassing AI-powered prevention, detection, response, and hunting across endpoints, containers, cloud workloads, and IoT devices.

AWS Marketplace link

Product link

Snyk

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/snyk/snyk`

Snyk provides a security platform that scans app components for security risks in workloads running on AWS. These risks are sent to Security Hub CSPM as findings, helping developers and security teams visualize and prioritize them along with the rest of their AWS security findings.

AWS Marketplace link

Partner documentation

Sonrai Security – Sonrai Dig

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/sonrai-security/sonrai-dig`

Sonrai Dig monitors and remediates cloud misconfigurations and policy violations, so you can improve your security and compliance posture.

Product link

Partner documentation

Sophos – Server Protection

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:062897671886:product/sophos/sophos-server-protection`

Sophos Server Protection defends the critical applications and data at the core of your organization, using comprehensive defense-in-depth techniques.

Product link

StackRox – StackRox Kubernetes Security

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/stackrox/kubernetes-security`

StackRox helps enterprises secure their container and Kubernetes deployments at scale by enforcing their compliance and security policies across the entire container life cycle – build, deploy, and run.

Product link

Partner documentation

Sumo Logic – Machine Data Analytics

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:956882708938:product/sumologicinc/sumologic-mda`

Sumo Logic is a secure, machine data analytics platform that enables development and security operations teams to build, run, and secure their AWS applications.

Product link

Partner documentation

Symantec – Cloud Workload Protection

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:754237914691:product/symantec-corp/symantec-cwp`

Cloud Workload Protection provides complete protection for your Amazon EC2 instances with antimalware, intrusion prevention, and file integrity monitoring.

Product link

Partner documentation

Tenable – Tenable.io

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:422820575223:product/tenable/tenable-io`

Accurately identify, investigate, and prioritize vulnerabilities. Managed in the cloud.

Product link

Partner documentation

Trend Micro – Cloud One

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/trend-micro/cloud-one`

Trend Micro Cloud One provides the right security information to teams at the right time and place. This integration sends security findings to Security Hub CSPM in real time, enhancing visibility into your AWS resources and Trend Micro Cloud One event details in Security Hub CSPM.

AWS Marketplace link

Partner documentation

Vectra – Cognito Detect

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:978576646331:product/vectra-ai/cognito-detect`

Vectra is transforming cybersecurity by applying advanced AI to detect and respond to hidden cyberattackers before they can steal or cause damage.

AWS Marketplace link

Partner documentation

Wiz – Wiz Security

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/wiz-security/wiz-security`

Wiz continuously analyzes configurations, vulnerabilities, networks, IAM settings, secrets, and more across your AWS accounts, users, and workloads to discover critical issues that represent actual risk. Integrate Wiz with Security Hub CSPM to visualize and respond to issues that Wiz detects from the Security Hub CSPM console.

AWS Marketplace link

Partner documentation

Third-party integrations that receive findings from Security Hub CSPM

The following third-party partner product integrations can receive findings from Security Hub CSPM. Where noted, the product might also update findings. In this case, updates that you make to findings in the partner product are also reflected in Security Hub CSPM.

Atlassian - Jira Service Management

Integration type: Receive and update

The AWS Service Management Connector for Jira sends findings from Security Hub CSPM to Jira. Jira issues are created based on the findings. When the Jira issues are updated, the corresponding findings are updated in Security Hub CSPM.

The integration only supports Jira Server and Jira Data Center.

For an overview of the integration and how it works, watch the video AWS Security Hub CSPM – Bidirectional integration with Atlassian Jira Service Management.

Product link

Partner documentation

Atlassian - Jira Service Management Cloud

Integration type: Receive and update

Jira Service Management Cloud is the cloud component of Jira Service Management.

The AWS Service Management Connector for Jira sends findings from Security Hub CSPM to Jira. The findings trigger the creation of issues in Jira Service Management Cloud. When you update

those issues in Jira Service Management Cloud, the corresponding findings are also updated in Security Hub CSPM.

Product link

Partner documentation

Atlassian – Opsgenie

Integration type: Receive

Opsgenie is a modern incident management solution for operating always-on services, empowering development and operations teams to plan for service disruptions and stay in control during incidents.

Integrating with Security Hub CSPM ensures that mission critical security-related incidents are routed to the appropriate teams for immediate resolution.

Product link

Partner documentation

Dynatrace

Integration type: Receive

The Dynatrace integration with Security Hub CSPM helps to unify, visualize, and automate security findings across tools and environments. Adding Dynatrace runtime context to security findings allows smarter prioritization, helps reduce noise from alerts, and focuses your DevSecOps teams on efficiently remedying the critical issues that affect your production environments and applications.

Product link

Partner documentation

Fortinet – FortiCNP

Integration type: Receive

FortiCNP is a Cloud Native Protection product that aggregates security findings into actionable insights and prioritizes security insights based on risk score to reduce alert fatigue and accelerate remediation.

AWS Marketplace link

Partner documentation

IBM – QRadar

Integration type: Receive

IBM QRadar SIEM provides security teams with the ability to quickly and accurately detect, prioritize, investigate, and respond to threats.

Product link

Partner documentation

Logz.io Cloud SIEM

Integration type: Receive

Logz.io is a provider of Cloud SIEM that provides advanced correlation of log and event data to help security teams to detect, analyze, and respond to security threats in real time.

Product link

Partner documentation

MetricStream – CyberGRC

Integration type: Receive

MetricStream CyberGRC helps you manage, measure, and mitigate cybersecurity risks. By receiving Security Hub CSPM findings, CyberGRC provides more visibility into these risks, so you can prioritize cybersecurity investments and comply with IT policies.

AWS Marketplace link

Product link

MicroFocus – MicroFocus Arcsight

Integration type: Receive

ArcSight accelerates effective threat detection and response in real time, integrating event correlation and supervised and unsupervised analytics with response automation and orchestration.

Product link

Partner documentation

New Relic Vulnerability Management

Integration type: Receive

New Relic Vulnerability Management receives security findings from Security Hub CSPM, so you can get a centralized view of security alongside performance telemetry in context across your stack.

AWS Marketplace link

Partner documentation

PagerDuty – PagerDuty

Integration type: Receive

The PagerDuty digital operations management platform empowers teams to proactively mitigate customer-impacting issues by automatically turning any signal into the right insight and action.

AWS users can use the PagerDuty set of AWS integrations to scale their AWS and hybrid environments with confidence.

When coupled with Security Hub CSPM aggregated and organized security alerts, PagerDuty allows teams to automate their threat response process and quickly set up custom actions to prevent potential issues.

PagerDuty users who are undertaking a cloud migration project can move quickly, while decreasing the impact of issues that occur throughout the migration lifecycle.

Product link

Partner documentation

Palo Alto Networks – Cortex XSOAR

Integration type: Receive

Cortex XSOAR is a Security Orchestration, Automation, and Response (SOAR) platform that integrates with your entire security product stack to accelerate incident response and security operations.

Product link

Partner documentation

Palo Alto Networks – VM-Series

Integration type: Receive

Palo Alto VM-Series integration with Security Hub CSPM collects threat intelligence and sends it to the VM-Series next-generation firewall as an automatic security policy update that blocks malicious IP address activity.

Product link

Partner documentation

Rackspace Technology – Cloud Native Security

Integration type: Receive

Rackspace Technology provides managed security services on top of native AWS security products for 24x7x365 monitoring by Rackspace SOC, advanced analysis, and threat remediation.

Product link

Rapid7 – InsightConnect

Integration type: Receive

Rapid7 InsightConnect is a security orchestration and automation solution that enables your team to optimize SOC operations with little to no code.

Product link

Partner documentation

RSA – RSA Archer

Integration type: Receive

RSA Archer IT and Security Risk Management allows you to determine which assets are critical to your business, establish and communicate security policies and standards, detect and respond to attacks, identify and remediate security deficiencies, and establish clear IT risk management best practices.

Product link

Partner documentation

ServiceNow – ITSM

Integration type: Receive and update

The ServiceNow integration with Security Hub CSPM allows security findings from Security Hub CSPM to be viewed within ServiceNow ITSM. You can also configure ServiceNow to automatically create an incident or problem when it receives a finding from Security Hub CSPM.

Any updates to these incidents and problems result in updates to the findings in Security Hub CSPM.

For an overview of the integration and how it works, watch the video AWS Security Hub CSPM - Bidirectional integration with ServiceNow ITSM.

Product link

Partner documentation

Slack – Slack

Integration type: Receive

Slack is a layer of the business technology stack that brings together people, data, and applications. It is a single place where people can effectively work together, find important information, and access hundreds of thousands of critical applications and services to do their best work.

Product link

Partner documentation

Splunk – Splunk Enterprise

Integration type: Receive

Splunk uses Amazon CloudWatch Events as a consumer of Security Hub CSPM findings. Send your data to Splunk for advanced security analytics and SIEM.

Product link

Partner documentation

Splunk – Splunk Phantom

Integration type: Receive

With the Splunk Phantom application for AWS Security Hub CSPM, findings are sent to Phantom for automated context enrichment with additional threat intelligence information or to perform automated response actions.

Product link

Partner documentation

ThreatModeler

Integration type: Receive

ThreatModeler is an automated threat modeling solution that secures and scales the enterprise software and cloud development life cycle.

Product link

Partner documentation

Trellix – Trellix Helix

Integration type: Receive

Trellix Helix is a cloud-hosted security operations platform that allows organizations to take control of any incident from alert to fix.

Product link

Partner documentation

Third-party integrations that send findings to and receive findings from Security Hub CSPM

The following third-party partner product integrations can send findings to and receive findings from Security Hub CSPM.

Caveonix – Caveonix Cloud

Integration type: Send and receive

Product ARN: `arn:aws:securityhub:<REGION>::product/caveonix/caveonix-cloud`

The Caveonix AI-powered platform automates visibility, assessment, and mitigation in hybrid clouds, covering cloud-native services, VMs, and containers. Integrated with AWS Security Hub CSPM, Caveonix merges AWS data and advanced analytics for insights into security alerts and compliance.

AWS Marketplace link

Partner documentation

Cloud Custodian – Cloud Custodian

Integration type: Send and receive

Product ARN: `arn:aws:securityhub:<REGION>::product/cloud-custodian/cloud-custodian`

Cloud Custodian enables users to be well managed in the cloud. The simple YAML DSL allows easily defined rules to enable a well-managed cloud infrastructure that's both secure and cost optimized.

Product link

Partner documentation

DisruptOps, Inc. – DisruptOPS

Integration type: Send and receive

Product ARN: `arn:aws:securityhub:<REGION>::product/disruptops-inc/disruptops`

The DisruptOps Security Operations Platform helps organizations maintain best security practices in your cloud through the use of automated guardrails.

Product link

Partner documentation

Kion

Integration type: Send and receive

Product ARN: `arn:aws:securityhub:<REGION>::product/cloudtamerio/cloudtamerio`

Kion (formerly cloudtamer.io) is a complete cloud governance solution for AWS. Kion gives stakeholders visibility into cloud operations and helps cloud users manage accounts, control budget and cost, and ensure continuous compliance.

Product link

Partner documentation

Turbot – Turbot

Integration type: Send and receive

Product ARN: `arn:aws:securityhub:<REGION>::product/turbot/turbot`

Turbot ensures that your cloud infrastructure is secure, compliant, scalable, and cost optimized.

Product link

Partner documentation

Integrating Security Hub CSPM with custom products

In addition to findings generated by integrated AWS services and third-party products, AWS Security Hub CSPM can consume findings that are generated by other custom security products.

You can send these findings to Security Hub CSPM by using the BatchImportFindings operation of the Security Hub CSPM API. You can use the same operation to update findings from custom products that you already sent to Security Hub CSPM.

When setting up the custom integration, use the guidelines and checklists provided in the *Security Hub CSPM Partner Integration Guide*.

Requirements and recommendations for custom product integrations

Before you can successfully invoke the `BatchImportFindings` API operation, you must enable Security Hub CSPM.

You must also provide finding details for the custom product using the the section called “Finding format: ASFF”. Review the following requirements and recommendations for custom product integrations:

Setting the product ARN

When you enable Security Hub CSPM, a default product Amazon Resource Name (ARN) for Security Hub CSPM is generated in your current account.

This product ARN has the following format: `arn:aws:securityhub:<region>:<account-id>:product/<account-id>/default`. For example, `arn:aws:securityhub:us-west-2:123456789012:product/123456789012/default`.

Use this product ARN as the value for the `ProductArn` attribute when invoking the `BatchImportFindings` API operation.

Setting the company and product names

You can use `BatchImportFindings` to set a preferred company name and product name for the custom integration that is sending findings to Security Hub CSPM.

Your specified names replace the preconfigured company name and product name, called personal name and default name respectively, and appear in the Security Hub CSPM console and the JSON of each finding. See BatchImportFindings for finding providers.

Setting the finding IDs

You must supply, manage, and increment your own finding IDs, using the `Id` attribute.

Each new finding should have a unique finding ID. If the custom product sends multiple findings with the same finding ID, Security Hub CSPM only processes the first finding.

Setting the account ID

You must specify your own account ID, using the `AwsAccountId` attribute.

Setting the created at and updated at dates

You must supply your own timestamps for the `CreatedAt` and `UpdatedAt` attributes.

Updating findings from custom products

In addition to sending new findings from custom products, you can also use the `BatchImportFindings` API operation to update existing findings from custom products.

To update existing findings, use the existing finding ID (via the `Id` attribute). Resend the full finding with the appropriate information updated in the request, including a modified `UpdatedAt` timestamp.

Example custom integrations

You can use the following example custom product integrations as a guide to create your own custom solutions:

Sending findings from Chef InSpec scans to Security Hub CSPM

You can create an AWS CloudFormation template that runs a Chef InSpec compliance scan and then sends findings to Security Hub CSPM.

For more details, see Continuous compliance monitoring with Chef InSpec and AWS Security Hub CSPM.

Sending container vulnerabilities detected by Trivy to Security Hub CSPM

You can create an AWS CloudFormation template that uses AquaSecurity Trivy to scan containers for vulnerabilities, and then sends those vulnerability findings to Security Hub CSPM.

For more details, see How to build a CI/CD pipeline for container vulnerability scanning with Trivy andAWS Security Hub CSPM.

Creating and updating findings in Security Hub CSPM

In AWS Security Hub CSPM, a *finding* is an observable record of a security check or security-related detection. A finding can originate from one of the following sources:

- A security check for a control in Security Hub CSPM.
- An integration with another AWS service.
- An integration with a third-party product.
- A custom integration.

Security Hub CSPM normalizes findings from all sources into a standard syntax and format called the *AWS Security Finding Format (ASFF)*. For detailed information about this format, including descriptions of individual ASFF fields, see AWS Security Finding Format (ASFF). If you enable cross-Region aggregation, Security Hub CSPM also aggregates new and updated findings automatically from all linked Regions to an aggregation Region that you specify. For more information, see Understanding cross-Region aggregation in Security Hub CSPM.

After a finding is created, it can be updated as follows:

- A finding provider can use the BatchImportFindings operation of the Security Hub CSPM API to update general information about the finding. Finding providers can only update findings that they created.
- A customer can use the Security Hub CSPM console or the BatchUpdateFindings operation of the Security Hub CSPM API to update the status of the investigation into the finding. The `BatchUpdateFindings` operation can also be used by a SIEM, ticketing, incident management, SOAR, or other type of tool on behalf of a customer.

To reduce finding noise and streamline tracking and analysis of individual findings, Security Hub CSPM automatically deletes findings that haven't been updated recently. The timing with which Security Hub CSPM does this depends on whether a finding is active or archived:

- An *active finding* is a finding whose record state (`RecordState`) is `ACTIVE`. Security Hub CSPM stores active findings for 90 days. If an active finding hasn't been updated for 90 days, it expires and Security Hub CSPM permanently deletes it.
- An *archived finding* is a finding whose record state (`RecordState`) is `ARCHIVED`. Security Hub CSPM stores archived findings for 30 days. If an archived finding hasn't been updated for 30 days, it expires and Security Hub CSPM permanently deletes it.

For control findings, which are findings that Security Hub CSPM generates from security checks for controls, Security Hub CSPM determines whether a finding has expired based on the value for the `UpdatedAt` field of the finding. If this value was more than 90 days ago for an active finding, Security Hub CSPM permanently deletes the finding. If this value was more than 30 days ago for an archived finding, Security Hub CSPM permanently deletes the finding.

For all other types of findings, Security Hub CSPM determines whether a finding has expired based on the values for the `ProcessedAt` and `UpdatedAt` fields of the finding. Security Hub CSPM compares the values for these fields and determines which is more recent. If the more recent value was more than 90 days ago for an active finding, Security Hub CSPM permanently deletes the finding. If the more recent value was more than 30 days ago for an archived finding, Security Hub CSPM permanently deletes the finding. Finding providers can change the value for the `UpdatedAt` field of one or more findings by using the BatchImportFindings operation of the Security Hub CSPM API.

For longer-term retention of findings, you can export findings to an S3 bucket. You can do this by using a custom action with an Amazon EventBridge rule. For more information, see Using EventBridge for automated response and remediation.

Topics

BatchImportFindings for finding providers

Finding providers can use the BatchImportFindings operation to create new findings in AWS Security Hub CSPM. They can also use this operation to update findings that they created. Finding providers can't update findings that they didn't create.

Customers, SIEMs, ticketing, SOAR, and other types of tools must use the BatchUpdateFindings operation to make updates related to their investigation of findings from finding providers. For more information, see the section called “BatchUpdateFindings for customers”.

When Security Hub CSPM receives a `BatchImportFindings` request to create or update a finding, it automatically generates a **Security Hub Findings - Imported** event in Amazon EventBridge. You can take automated action on that event. For more information, see the section called “Automated response and remediation”.

Prerequisites for using `BatchImportFindings`

`BatchImportFindings` must be called by one of the following:

- The account that is associated with the findings. The identifier of the associated account must match the value of the `AwsAccountId` attribute for the finding.
- An account that is allow-listed as an official Security Hub CSPM partner integration.

Security Hub CSPM can only accept finding updates for accounts that have Security Hub CSPM enabled. The finding provider also must be enabled. If Security Hub CSPM is disabled, or the

finding provider integration is not enabled, then the findings are returned in the `FailedFindings` list, with an `InvalidAccess` error.

Determining whether to create or update a finding

To determine whether to create or update a finding, Security Hub CSPM checks the `ID` field. If the value of `ID` doesn't match an existing finding, Security Hub CSPM creates a new finding.

If `ID` matches an existing finding, Security Hub CSPM checks the `UpdatedAt` field for the update, and proceeds as follows:

- If `UpdatedAt` on the update matches or occurs before `UpdatedAt` on the existing finding, Security Hub CSPM ignores the update request.
- If `UpdatedAt` on the update occurs after `UpdatedAt` on the existing finding, Security Hub CSPM updates the existing finding.

Restrictions on finding updates with `BatchImportFindings`

Finding providers can't use `BatchImportFindings` to update the following attributes of an existing finding:

- `Note`
- `UserDefinedFields`
- `VerificationState`
- `Workflow`

Security Hub CSPM ignores any content provided in a `BatchImportFindings` request for these attributes. Customers, or entities acting on their behalf (such as ticketing tools), can use `BatchUpdateFindings` to update these attributes.

Updating findings with FindingProviderFields

Finding providers also shouldn't use `BatchImportFindings` to update the following top-level attributes in the AWS Security Finding Format (ASFF):

- `Confidence`
- `Criticality`

- `RelatedFindings`
- `Severity`
- `Types`

Instead, finding providers should use the `FindingProviderFields` object to provide values for these attributes.

Example

```
"FindingProviderFields": {
    "Confidence": 42,
    "Criticality": 99,
    "RelatedFindings":[
      {
        "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",
        "Id": "123e4567-e89b-12d3-a456-426655440000"
      }
    ],
    "Severity": {
        "Label": "MEDIUM",
        "Original": "MEDIUM"
    },
    "Types": [ "Software and Configuration Checks/Vulnerabilities/CVE" ]
}
```

For `BatchImportFindings` requests, Security Hub CSPM handles values in the top-level attributes and in `FindingProviderFields` as follows.

(Preferred) `BatchImportFindings` provides a value for an attribute in `FindingProviderFields`, but does not provide a value for the corresponding top-level attribute.

For example, `BatchImportFindings` provides `FindingProviderFields.Confidence`, but does not provide `Confidence`. This is the preferred option for `BatchImportFindings` requests.

Security Hub CSPM updates the value of the attribute in `FindingProviderFields`.

It replicates the value to the top-level attribute only if the attribute wasn't already updated by `BatchUpdateFindings`.

`BatchImportFindings` provides a value for a top-level attribute, but does not provide a value for the corresponding attribute in `FindingProviderFields`.

For example, `BatchImportFindings` provides `Confidence`, but does not provide `FindingProviderFields.Confidence`.

Security Hub CSPM uses the value to update the attribute in `FindingProviderFields`. It overwrites any existing value.

Security Hub CSPM updates the top-level attribute only if the attribute was not already updated by `BatchUpdateFindings`.

`BatchImportFindings` provides a value for both a top-level attribute and the corresponding attribute in `FindingProviderFields`.

For example, `BatchImportFindings` provides both `Confidence` and `FindingProviderFields.Confidence`.

For a new finding, Security Hub CSPM uses the value in `FindingProviderFields` to populate both the top-level attribute and the corresponding attribute in `FindingProviderFields`. It doesn't use the provided top-level attribute value.

For an existing finding, Security Hub CSPM uses both values. However, it updates the top-level attribute value only if the attribute was not already updated by `BatchUpdateFindings`.

BatchUpdateFindings for customers

AWS Security Hub CSPM customers, and entities acting on their behalf, can use the BatchUpdateFindings operation to update information related to the processing of Security Hub CSPM findings from finding providers. As a customer, you can use this operation directly. SIEM, ticketing, incident management, and SOAR tools can also use this operation on behalf of a customer.

You can't use the `BatchUpdateFindings` operation to create new findings. However, you can use it to update up to 100 existing findings at a time. In a `BatchUpdateFindings` request, you specify which findings to update, which AWS Security Finding Format (ASFF) fields to update for the findings, and the new values for the fields. Security Hub CSPM then updates the findings as specified in your request. This process can take several minutes. If you update findings by using the `BatchUpdateFindings` operation, your updates don't affect existing values for the `UpdatedAt` field of the findings.

When Security Hub CSPM receives a `BatchUpdateFindings` request to update a finding, it automatically generates a **Security Hub Findings – Imported** event in Amazon EventBridge. You can optionally use this event to take automated action on the specified finding. For more information, see the section called "Automated response and remediation".

Available fields for BatchUpdateFindings

If you are signed in to a Security Hub CSPM administrator account, you can use `BatchUpdateFindings` to update findings that were generated by the administrator account or member accounts. Member accounts can use `BatchUpdateFindings` to update findings for their account only.

Customers can use `BatchUpdateFindings` to update the following fields and objects:

- `Confidence`
- `Criticality`
- `Note`
- `RelatedFindings`
- `Severity`
- `Types`
- `UserDefinedFields`
- `VerificationState`
- `Workflow`

Configuring access to BatchUpdateFindings

You can configure AWS Identity and Access Management (IAM) policies to restrict access to using `BatchUpdateFindings` to update finding fields and field values.

In a statement to restrict access to `BatchUpdateFindings`, use the following values:

- `Action` is `securityhub:BatchUpdateFindings`
- `Effect` is `Deny`
- For `Condition`, you can deny a `BatchUpdateFindings` request based on the following:
  - The finding includes a specific field.
  - The finding includes a specific field value.

Condition keys

These are the condition keys for restricting access to `BatchUpdateFindings`.

ASFF field

The condition key for an ASFF field is as follows:

```
securityhub:ASFFSyntaxPath/<fieldName>
```

Replace *<fieldName>* with the ASFF field. When configuring access to `BatchUpdateFindings`, include one or more specific ASFF fields in your IAM policy rather than a parent-level field. For example, to restrict access to the `Workflow.Status` field, you must include `securityhub:ASFFSyntaxPath/Workflow.Status` in your policy instead of the `Workflow` parent-level field.

Disallowing all updates to a field

To prevent a user from making any update to a specific field, use a condition like this:

```
 "Condition": {
                "Null": {
                    "securityhub:ASFFSyntaxPath/<fieldName>": "false"
               }
}
```

For example, the following statement indicates that `BatchUpdateFindings` can't be used to update the `Workflow.Status` field of findings.

```
{
    "Sid": "VisualEditor0",
    "Effect": "Deny",
    "Action": "securityhub:BatchUpdateFindings",
    "Resource": "*",
    "Condition": {
        "Null": {
            "securityhub:ASFFSyntaxPath/Workflow.Status": "false"
        }
    }
}
```

Disallowing specific field values

To prevent a user from setting a field to a specific value, use a condition like this:

```
"Condition": {
                "StringEquals": {
                    "securityhub:ASFFSyntaxPath/<fieldName>": "<fieldValue>"
                }
}
```

For example, the following statement indicates that `BatchUpdateFindings` can't be used to set `Workflow.Status` to `SUPPRESSED`.

```
{
    "Sid": "VisualEditor0",
    "Effect": "Deny",
    "Action": "securityhub:BatchUpdateFindings",
    "Resource": "*",
    "Condition": {
    "StringEquals": {
        "securityhub:ASFFSyntaxPath/Workflow.Status": "SUPPRESSED"
    }
}
```

You can also provide a list of values that are not permitted.

```
 "Condition": {
                "StringEquals": {
                    "securityhub:ASFFSyntaxPath/<fieldName>": [ "<fieldValue1>",
 "<fieldValue2>", "<fieldValuen>" ]
                }
}
```

For example, the following statement indicates that `BatchUpdateFindings` can't be used to set `Workflow.Status` to either `RESOLVED` or `SUPPRESSED`.

```
{
    "Sid": "VisualEditor0",
    "Effect": "Deny",
    "Action": "securityhub:BatchUpdateFindings",
    "Resource": "*",
```

```
    "Condition": {
    "StringEquals": {
        "securityhub:ASFFSyntaxPath/Workflow.Status": [
            "RESOLVED",
            "NOTIFIED"
        ]
    }
}
```

Reviewing finding details and history in Security Hub CSPM

In AWS Security Hub CSPM, a *finding* is an observable record of a security check or security-related detection. Security Hub CSPM generates a finding when it completes a security check of a control and when it ingests a finding from an integrated AWS service or third-party product. Each finding includes a history of changes and other details, such as a severity rating and information about the affected resources.

You can review the history and other details of individual findings on the Security Hub CSPM console or programmatically with the Security Hub CSPM API or the AWS CLI.

To help you streamline your analysis, the Security Hub CSPM console displays a finding panel when you choose a specific finding. The panel includes different menus and tabs for reviewing specific details of a finding.

Actions menu

From this menu, you can review the complete JSON of a finding or add notes. A finding can have only one note attached to it at a time. This menu also provides options to set the workflow status of a finding or send a finding to a custom action in Amazon EventBridge.

Investigate menu

From this menu, you can investigate a finding in Amazon Detective. Detective extracts entities, such as IP addresses and AWS users, from a finding and visualizes their activity. You can use the entity activity as a starting point to investigate the cause and impact of a finding.

Overview tab

This tab provides a summary of a finding. For example, you can determine when a finding was created and last updated, in which account it exists, and the source of the finding. For control findings, this tab also shows the name of the associated AWS Config rule and a link to remediation guidance in the Security Hub CSPM documentation.

In the **Resources** snapshot on the **Overview** tab, you can get a brief overview of the resources involved in a finding. For some resources, this includes an **Open resource** option, which links directly to an impacted resource on the relevant AWS service console. The **History** snapshot shows up to two changes made to the finding on the most recent date for which history is being tracked. For example, if you made one change yesterday and another one today, the snapshot shows today's change. To review earlier entries, switch to the **History** tab.

The **Compliance** row expands to show more details. For example, if a control includes parameters, you can review the parameter values that Security Hub CSPM currently uses when conducting security checks for the control.

Resources tab

This tab provides details about the resources involved in a finding. If you're signed in to the account that owns a resource, you can review the resource in the applicable AWS service console. If you're not the owner of a resource, this tab displays the AWS account ID for the owner.

The **Details** row shows resource-specific details in a finding. It shows the ResourceDetails section of the finding in JSON format.

The **Tags** row shows tag keys and values that are assigned to the resources involved in a finding. Resources that are supported by the GetResources operation of the AWS Resource Groups Tagging API can be tagged. Security Hub CSPM calls this operation by using a service-linked role when processing new or updated findings, and retrieves the resource tags if the AWS Security Finding Format (ASFF) `Resource.Id` field is populated with the ARN of a resource. Security Hub CSPM ignores invalid resource IDs. For more information about the inclusion of resource tags in findings, see Tags.

History tab

This tab tracks the history of a finding. Finding history is available for active and archived findings. It provides an immutable trail of changes made to a finding over time, including what ASFF field changed, when the change occurred, and by which user. Each page on the tab displays up to 20 changes. More recent changes are displayed first.

For active findings, finding history is available for up to 90 days. For archived findings, finding history is available for up to 30 days. Finding history includes changes that were made manually, or automatically by Security Hub CSPM automation rules. It doesn't include changes to top-level timestamp fields, such as the `CreatedAt` and `UpdatedAt` fields.

If you're signed in to a Security Hub CSPM administrator account, finding history is for the administrator account and all member accounts.

Threat tab

This tab includes data from the Action, Malware, and ProcessDetails objects of the ASFF, including the type of threat and whether a resource is the target or actor. These details typically apply to findings that originate in Amazon GuardDuty.

Vulnerabilities tab

This tab displays data from the Vulnerability object of the ASFF, including whether there are exploits or available fixes associated with a finding. These details typically apply to findings that originate in Amazon Inspector.

The rows on each tab include a copy or filter option. For example, if you open the panel for a finding that has a workflow status of **Notified**, you can choose the filter option next to the **Workflow status** row. If you choose **Show all findings with this value**, Security Hub CSPM filters the findings table and displays only findings with the same workflow status.

Reviewing finding details and history

Choose your preferred method, and follow the steps to review finding details in Security Hub CSPM.

If you enable cross-Region aggregation and sign in to the aggregation Region, finding data includes data from the aggregation Region and linked Regions. In other Regions, finding data is specific to that Region only. For more information about cross-Region aggregation, see the section called “Aggregating data across Regions”.

Security Hub CSPM console

Reviewing finding details and history

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. To display a finding list, do one of the following:
   - In the navigation pane, choose **Findings**. Add search filters as necessary to narrow the finding list.

- In the navigation pane, choose **Insights**. Choose an insight. Then, in the results list, choose an insight result.
- In the navigation pane, choose **Integrations**. Choose **See findings** for an integration.
- In the navigation pane, choose **Controls**.

3. Choose a finding. The finding panel displays the details of the finding.
4. In the finding panel, do any of the following:

- To review specific details for the finding, choose a tab.
- To take action on the finding, choose an option from the **Actions** menu.
- To investigate the finding in Amazon Detective, choose an **Investigate** option.

> **Note**
>
> If you integrate with AWS Organizations and you're signed in to a member account, the finding panel includes the account name. For member accounts that are invited manually, instead of through Organizations, the finding panel includes only the account ID.

Security Hub CSPM API

Use the GetFindings operation of the Security Hub CSPM API, or if you're using the AWS CLI, run the get-findings command. You can provide one or more values for the `Filters` parameter to narrow the findings to retrieve.

If the volume of results is too large, you can use the `MaxResults` parameter to limit the findings to a specified number and the `NextToken` parameter to paginate findings. Use the `SortCriteria` parameter to sort the findings by a specific field.

For example, the following AWS CLI command retrieves the findings that match the specified filter criteria, and sorts the results in descending order by the `LastObservedAt` field. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub get-findings \
--filters '{"GeneratorId":[{"Value": "aws-
foundational","Comparison":"PREFIX"}],"WorkflowStatus": [{"Value":
```

```
"NEW","Comparison":"EQUALS"}],"Confidence": [{"Gte": 85}]}' --sort-criteria
'{"Field": "LastObservedAt","SortOrder": "desc"}' --page-size 5 --max-items 100
```

To review finding history, use the GetFindingHistory operation. If you're using the AWS CLI, run the get-finding-history command. Identify the finding that you want to get history for with the `ProductArn` and `Id` fields. For information about these fields, see AwsSecurityFindingIdentifier. Each request can retrieve the history for only one finding.

For example, the following AWS CLI command retrieves the history for the specified finding. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub get-finding-history \
--region us-west-2 \
--finding-identifier Id="a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",ProductArn="arn:aws:securityhub:us-
west-2:123456789012:product/123456789012/default" \
--max-results 2 \
--start-time "2021-09-30T15:53:35.573Z" \
--end-time "2021-09-31T15:53:35.573Z"
```

PowerShell

Use the `Get-SHUBFinding` cmdlet. Optionally populate the `Filter` parameter to narrow the findings to retrieve.

For example, the following cmdlet retrieves the findings that match the specified filters.

```
Get-SHUBFinding -Filter @{AwsAccountId =
 [Amazon.SecurityHub.Model.StringFilter]@{Comparison = "EQUALS"; Value =
 "XXX"};ComplianceStatus = [Amazon.SecurityHub.Model.StringFilter]@{Comparison =
 "EQUALS"; Value = 'FAILED'}}
```

Note

If you filter findings by `CompanyName` or `ProductName`, Security Hub CSPM uses the values that are part of the `ProductFields` ASFF object. Security Hub CSPM doesn't use the top-level `CompanyName` and `ProductName` fields.

Filtering findings in Security Hub CSPM

AWS Security Hub CSPM generates its own findings from security checks and receives findings from integrated products. You can display a list of findings on the **Findings**, **Integrations**, and **Insights** pages of the Security Hub CSPM console. You can add filters to narrow a finding list so that the list is relevant to your organization or use case.

For information about filtering findings for a specific security control, see the section called "Filtering and sorting control findings". The information on this page applies to the **Findings**, **Insights**, and **Integrations** pages.

Default filters on finding lists

By default, finding lists on the Security Hub CSPM console are filtered based on the `RecordState` and `Workflow.Status` fields of the AWS Security Finding Format (ASFF). This is in addition to the filters for a specific insight or integration.

Record state indicates whether a finding is active or archived. By default, a finding list only shows active findings. A finding provider can archive a finding if it's no longer active or important. Security Hub CSPM also automatically archives control findings if the associated resource is deleted.

Workflow status indicates the status of an investigation into a finding. By default, a finding list only shows findings with a workflow status of `NEW` or `NOTIFIED`. You can update the workflow status of a finding.

Instructions for adding filters

You can filter a finding list by up to ten attributes. For each attribute, you can provide up to 20 filter values.

When filtering the finding list, Security Hub CSPM applies `AND` logic to the set of filters. A finding matches only if it matches all of the provided filters. For example, if you add GuardDuty as a filter for **Product name**, and `AwsS3Bucket` as a filter for **Resource type**, Security Hub CSPM displays findings that match both of these criteria.

Security Hub CSPM applies `OR` logic to filters that use the same attribute but different values. For example, if you add both GuardDuty and Amazon Inspector as filter values for **Product name**, Security Hub CSPM displays findings that were generated by either GuardDuty or Amazon Inspector.

To add filters to a findings list (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. To display a findings list, take one of the following actions from the navigation pane:
   - Choose **Findings**.
   - Choose **Insights**. Choose an insight. Then, on the results list, choose an insight result.
   - Choose **Integrations**. Choose **See findings** for an integration.
3. In the **Add filters** box, select one or more fileds to filter by.

   When you filter by **Company name** or **Product name**, the console uses the top-level `CompanyName` and `ProductName` fields of the AWS Security Finding Format (ASFF). The API uses the values that are nested under `ProductFields`.
4. Choose the filter match type.

   For a string filter, you can choose from the following options:

   - **is** – Find a value that exactly matches the filter value.
   - **starts with** – Find a value that starts with the filter value.
   - **is not** – Find a value that does not match the filter value.
   - **does not start with** – Find a value that does not start with the filter value.

   For the **Resource tags** field, you can filter based on specific keys or values.

   For a numeric filter, you can choose whether to provide a single number (**Simple**) or a range of numbers (**Range**).

   For a date or time filter, you can choose whether to provide a length of time from the current date and time (**Rolling window**) or a specific date range (**Fixed range**).

   Adding multiple filters has the following interactions:

   - **is** and **starts with** filters are joined by OR. A value matches if it contains any of the filter values. For example, if you specify **Severity label is CRITICAL** and **Severity label is HIGH**, the results include both critical and high severity findings.

- **is not** and **does not start with** filters are joined by AND. A value matches only if it does not contain any of those filter values. For example, if you specify **Severity label is not LOW** and **Severity label is not MEDIUM**, the results don't include low or medium severity findings.

If you have an **is** filter on a field, you can't have an **is not** or a **does not start with** filter on the same field.

5. Specify the filter value. For string filters, the filter value is case sensitive.
6. Choose **Apply**.

For an existing filter, you can change the filter match type or value. On a filtered finding list, choose the filter. In the **Edit filter** box, choose the new match type or value, and then choose **Apply**.

To remove a filter, choose the **x** icon. The list is updated automatically to reflect the change.

Grouping findings in Security Hub CSPM

You can group findings in AWS Security Hub CSPM based on the values of a selected attribute.

When you group the findings, the list of findings is replaced with a list of values for the selected attribute in the matching findings. For each value, the list displays the number of matching findings.

For example, if you group the findings by AWS account ID, you see a list of account identifiers, with the number of matching findings for each account.

Security Hub CSPM can display up to 100 values for a selected attribute. If there are more than 100 values, you only see the first 100.

When you choose an attribute value, Security Hub CSPM displays the list of matching findings for that value.

To group the findings in a findings list (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. To display a findings list, take one of the following actions from the navigation pane:

   - Choose **Findings**.

- Choose **Insights**. Choose an insight. Then, on the results list, choose an insight result.
- Choose **Integrations**. Choose **See findings** for an integration.

3. In the **Group by** drop down, choose the attribute to use for the grouping.

   To remove a grouping attribute, choose the **x** icon. When you remove the grouping attribute, the list changes from the list of attribute values to a list of findings.

Setting the workflow status of findings in Security Hub CSPM

Workflow status tracks the progress of your investigation into a finding. Workflow status is specific to an individual finding and doesn't affect generation of new findings. For example, if you change the workflow status of a finding to `SUPPRESSED` or `RESOLVED`, your change doesn't prevent Security Hub CSPM from generating a new finding for the same issue.

The workflow status of a finding can be one of the following values.

NEW

The initial state of a finding before you review it.

Findings that are ingested from integrated AWS services, such as AWS Config, have `NEW` as their initial status.

Security Hub CSPM also resets the workflow status from either `NOTIFIED` or `RESOLVED` to `NEW` in the following cases:

- `RecordState` changes from `ARCHIVED` to `ACTIVE`.
- `Compliance.Status` changes from `PASSED` to `FAILED`, `WARNING`, or `NOT_AVAILABLE`.

These changes imply that additional investigation is required.

NOTIFIED

Indicates that you notified the resource owner about the security issue. You can use this status when you are not the resource owner, and you need intervention from the resource owner in order to resolve a security issue.

If one of the following occurs, the workflow status is changed automatically from `NOTIFIED` to `NEW`:

- `RecordState` changes from `ARCHIVED` to `ACTIVE`.
- `Compliance.Status` changes from `PASSED` to `FAILED`, `WARNING`, or `NOT_AVAILABLE`.

SUPPRESSED

Indicates that you reviewed the finding and do not believe that any action is needed.

The workflow status of a `SUPPRESSED` finding does not change if `RecordState` changes from `ARCHIVED` to `ACTIVE`.

RESOLVED

The finding was reviewed and remediated and is now considered resolved.

The finding remains `RESOLVED` unless one of the following occurs:

- `RecordState` changes from `ARCHIVED` to `ACTIVE`.
- `Compliance.Status` changes from `PASSED` to `FAILED`, `WARNING`, or `NOT_AVAILABLE`.

In those cases, the workflow status is automatically reset to `NEW`.

For findings from controls, if `Compliance.Status` is `PASSED`, Security Hub CSPM automatically sets the workflow status to `RESOLVED`.

Setting the workflow status of findings

To change the workflow status of one or more findings, you can use the Security Hub CSPM console or the Security Hub CSPM API. If you change the workflow status of a finding, note that it can take several minutes for Security Hub CSPM to process your request and update the finding.

> **Tip**
>
> You can also change the workflow status of findings automatically by using automation rules. With automation rules, you configure Security Hub CSPM to automatically update the workflow status of findings based on criteria that you specify. For more information, see Understanding automation rules in Security Hub CSPM.

To change the workflow status of one or more findings, choose your preferred method and follow the steps.

Security Hub CSPM console

To change the workflow status of findings

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, do one of the following to display a table of findings:
   - Choose **Findings**.
   - Choose **Insights**. Then choose an insight. In the insight results, choose a result.
   - Choose **Integrations**. Then, in the section for the integration, choose **See findings**.
   - Choose **Security standards**. Then, in the section for the standard, choose **View results**. In the table of controls, choose a control to display findings for the control.
3. In the findings table, select the check box for each finding whose workflow status you want to change.
4. At the top of the page, choose **Workflow status**, and then choose the new workflow status for the selected findings.
5. In the **Set workflow status** dialog box, optionally enter a note that details the reason for changing the workflow status. Then choose **Set status**.

Security Hub CSPM API

Use the BatchUpdateFindings operation. Provide both the finding ID and the ARN of the product that generated the finding. You can get these details by using the GetFindings operation.

AWS CLI

Run the batch-update-findings command. Provide both the finding ID and the ARN of the product that generated the finding. You can get these details by running the get-findings command.

```
batch-update-findings --finding-identifiers
 Id="<findingID>",ProductArn="<productARN>" --workflow Status="<workflowStatus>"
```

Example

```
aws securityhub batch-update-findings --finding-identifiers
 Id="arn:aws:securityhub:us-west-1:123456789012:subscription/
pci-dss/v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",ProductArn="arn:aws:securityhub:us-west-1::product/aws/securityhub" --
workflow Status="RESOLVED"
```

Sending findings to a custom Security Hub CSPM action

You can create AWS Security Hub CSPM custom actions to automate Security Hub CSPM with Amazon EventBridge. For custom actions, the event type is **Security Hub Findings - Custom Action**. After you set up a custom action, you can send findings to it. For more information and detailed steps on creating custom actions, see the section called “Automated response and remediation”.

To send findings to a custom action (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. To display a finding list, do one of the following:
   - In the Security Hub CSPM navigation pane, choose **Findings**.
   - In the Security Hub CSPM navigation pane, choose **Insights**. Choose an insight. Then on the results list, choose an insight result.
   - In the Security Hub CSPM navigation pane, choose **Integrations**. Choose **See findings** for an integration.
   - In the Security Hub CSPM navigation pane, choose **Security standards**. Choose **View results** to display a list of controls. Then choose the control name.
3. In the finding list, select the check box for each finding to send to the custom action.

   You can send up to 20 findings at a time.
4. For **Actions**, choose the custom action.

AWS Security Finding Format (ASFF)

AWS Security Hub CSPM consumes and aggregates findings from integrated AWS services and third-party products. Security Hub CSPM processes these findings using a standard findings format

called the *AWS Security Finding Format (ASFF)*, which eliminates the need for time-consuming data conversion efforts.

This page provides a complete outline of the JSON for a finding in the AWS Security Finding Format (ASFF). The format derives from JSON Schema. Choose the name of a linked object to review an example of a finding for that object. Comparing your Security Hub CSPM findings with the resources and examples shown here can help you interpret your findings.

For descriptions of individual ASFF attributes, see the section called "Required top-level ASFF attributes" and the section called "Optional top-level ASFF attributes".

```
"Findings": [
    {
     "Action": {
      "ActionType": "string",
      "AwsApiCallAction": {
       "AffectedResources": {
        "string": "string"
       },
       "Api": "string",
       "CallerType": "string",
       "DomainDetails": {
        "Domain": "string"
       },
       "FirstSeen": "string",
       "LastSeen": "string",
       "RemoteIpDetails": {
        "City": {
         "CityName": "string"
        },
        "Country": {
         "CountryCode": "string",
         "CountryName": "string"
        },
        "IpAddressV4": "string",
        "Geolocation": {
         "Lat": number,
         "Lon": number
        },
        "Organization": {
         "Asn": number,
         "AsnOrg": "string",
         "Isp": "string",
```

```
          "Org": "string"
         }
        },
        "ServiceName": "string"
       },
       "DnsRequestAction": {
        "Blocked": boolean,
        "Domain": "string",
        "Protocol": "string"
       },
       "NetworkConnectionAction": {
        "Blocked": boolean,
        "ConnectionDirection": "string",
        "LocalPortDetails": {
         "Port": number,
         "PortName": "string"
        },
        "Protocol": "string",
        "RemoteIpDetails": {
         "City": {
          "CityName": "string"
         },
         "Country": {
          "CountryCode": "string",
          "CountryName": "string"
         },
         "IpAddressV4": "string",
         "Geolocation": {
          "Lat": number,
          "Lon": number
         },
         "Organization": {
          "Asn": number,
          "AsnOrg": "string",
          "Isp": "string",
          "Org": "string"
         }
        },
        "RemotePortDetails": {
         "Port": number,
         "PortName": "string"
        }
       },
       "PortProbeAction": {
```

```
        "Blocked": boolean,
        "PortProbeDetails": [{
         "LocalIpDetails": {
          "IpAddressV4": "string"
         },
         "LocalPortDetails": {
          "Port": number,
          "PortName": "string"
         },
         "RemoteIpDetails": {
          "City": {
           "CityName": "string"
          },
          "Country": {
           "CountryCode": "string",
           "CountryName": "string"
          },
          "GeoLocation": {
           "Lat": number,
           "Lon": number
          },
          "IpAddressV4": "string",
          "Organization": {
           "Asn": number,
           "AsnOrg": "string",
           "Isp": "string",
           "Org": "string"
          }
         }
        }]
       }
      },
      "AwsAccountId": "string",
      "AwsAccountName": "string",
      "CompanyName": "string",
      "Compliance": {
       "AssociatedStandards": [{
        "StandardsId": "string"
       }],
       "RelatedRequirements": ["string"],
       "SecurityControlId": "string",
       "SecurityControlParameters": [
        {
         "Name": "string",
```

```
        "Value": ["string"]
       }
      ],
       "Status": "string",
       "StatusReasons": [
        {
         "Description": "string",
         "ReasonCode": "string"
        }
       ]
      },
      "Confidence": number,
      "CreatedAt": "string",
      "Criticality": number,
      "Description": "string",
      "Detection": {
       "Sequence": {
        "Uid": "string",
        "Actors": [{
         "Id": "string",
         "Session": {
          "Uid": "string",
          "MfAStatus": "string",
          "CreatedTime": "string",
          "Issuer": "string"
         },
         "User": {
          "CredentialUid": "string",
          "Name": "string",
          "Type": "string",
          "Uid": "string",
          "Account": {
           "Uid": "string",
           "Name": "string"
          }
         }
        }],
        "Endpoints": [{
         "Id": "string",
         "Ip": "string",
         "Domain": "string",
         "Port": number,
         "Location": {
          "City": "string",
```

```
          "Country": "string",
          "Lat": number,
          "Lon": number
         },
         "AutonomousSystem": {
          "Name": "string",
          "Number": number
         },
         "Connection": {
          "Direction": "string"
         }
        }],
        "Signals": [{
         "Id": "string",
         "Title": "string",
         "ActorIds": ["string"],
         "Count": number,
         "FirstSeenAt": number,
         "SignalIndicators": [
          {
           "Key": "string",
           "Title": "string",
           "Values": ["string"]
          },
          {
           "Key": "string",
           "Title": "string",
           "Values": ["string"]
          }
         ],
         "LastSeenAt": number,
         "Name": "string",
         "ResourceIds": ["string"],
         "Type": "string"
        }],
        "SequenceIndicators": [
         {
          "Key": "string",
          "Title": "string",
          "Values": ["string"]
         },
         {
          "Key": "string",
          "Title": "string",
```

```
          "Values": ["string"]
        }
       ]
      }
     },
     "FindingProviderFields": {
      "Confidence": number,
      "Criticality": number,
      "RelatedFindings": [{
       "ProductArn": "string",
       "Id": "string"
      }],
      "Severity": {
       "Label": "string",
       "Normalized": number,
       "Original": "string"
      },
      "Types": ["string"]
     },
     "FirstObservedAt": "string",
     "GeneratorId": "string",
     "Id": "string",
     "LastObservedAt": "string",
     "Malware": [{
      "Name": "string",
      "Path": "string",
      "State": "string",
      "Type": "string"
     }],
     "Network": {
      "DestinationDomain": "string",
      "DestinationIpV4": "string",
      "DestinationIpV6": "string",
      "DestinationPort": number,
      "Direction": "string",
      "OpenPortRange": {
       "Begin": integer,
       "End": integer
      },
      "Protocol": "string",
      "SourceDomain": "string",
      "SourceIpV4": "string",
      "SourceIpV6": "string",
      "SourceMac": "string",
```

```
      "SourcePort": number
     },
     "NetworkPath": [{
      "ComponentId": "string",
      "ComponentType": "string",
      "Egress": {
       "Destination": {
        "Address": ["string"],
        "PortRanges": [{
         "Begin": integer,
         "End": integer
        }]
       },
       "Protocol": "string",
       "Source": {
        "Address": ["string"],
        "PortRanges": [{
         "Begin": integer,
         "End": integer
        }]
       }
      },
      "Ingress": {
       "Destination": {
        "Address": ["string"],
        "PortRanges": [{
         "Begin": integer,
         "End": integer
        }]
       },
       "Protocol": "string",
       "Source": {
        "Address": ["string"],
        "PortRanges": [{
         "Begin": integer,
         "End": integer
        }]
       }
      }
     }],
     "Note": {
      "Text": "string",
      "UpdatedAt": "string",
      "UpdatedBy": "string"
```

```
    },
    "PatchSummary": {
     "FailedCount": number,
     "Id": "string",
     "InstalledCount": number,
     "InstalledOtherCount": number,
     "InstalledPendingReboot": number,
     "InstalledRejectedCount": number,
     "MissingCount": number,
     "Operation": "string",
     "OperationEndTime": "string",
     "OperationStartTime": "string",
     "RebootOption": "string"
    },
    "Process": {
     "LaunchedAt": "string",
     "Name": "string",
     "ParentPid": number,
     "Path": "string",
     "Pid": number,
     "TerminatedAt": "string"
    },
    "ProductArn": "string",
    "ProductFields": {
     "string": "string"
    },
    "ProductName": "string",
    "RecordState": "string",
    "Region": "string",
    "RelatedFindings": [{
     "Id": "string",
     "ProductArn": "string"
    }],
    "Remediation": {
     "Recommendation": {
      "Text": "string",
      "Url": "string"
     }
    },
    "Resources": [{
     "ApplicationArn": "string",
     "ApplicationName": "string",
     "DataClassification": {
      "DetailedResultsLocation": "string",
```

```
        "Result": {
         "AdditionalOccurrences": boolean,
         "CustomDataIdentifiers": {
          "Detections": [{
           "Arn": "string",
           "Count": integer,
           "Name": "string",
           "Occurrences": {
            "Cells": [{
             "CellReference": "string",
             "Column": integer,
             "ColumnName": "string",
             "Row": integer
            }],
            "LineRanges": [{
             "End": integer,
             "Start": integer,
             "StartColumn": integer
            }],
            "OffsetRanges": [{
             "End": integer,
             "Start": integer,
             "StartColumn": integer
            }],
            "Pages": [{
             "LineRange": {
              "End": integer,
              "Start": integer,
              "StartColumn": integer
             },
             "OffsetRange": {
              "End": integer,
              "Start": integer,
              "StartColumn": integer
             },
             "PageNumber": integer
            }],
            "Records": [{
             "JsonPath": "string",
             "RecordIndex": integer
            }]
           }
          }],
          "TotalCount": integer
```

```
        },
        "MimeType": "string",
        "SensitiveData": [{
         "Category": "string",
         "Detections": [{
          "Count": integer,
          "Occurrences": {
           "Cells": [{
            "CellReference": "string",
            "Column": integer,
            "ColumnName": "string",
            "Row": integer
           }],
           "LineRanges": [{
            "End": integer,
            "Start": integer,
            "StartColumn": integer
           }],
           "OffsetRanges": [{
            "End": integer,
            "Start": integer,
            "StartColumn": integer
           }],
           "Pages": [{
            "LineRange": {
             "End": integer,
             "Start": integer,
             "StartColumn": integer
            },
            "OffsetRange": {
             "End": integer,
             "Start": integer,
             "StartColumn": integer
            },
            "PageNumber": integer
           }],
           "Records": [{
            "JsonPath": "string",
            "RecordIndex": integer
           }]
          },
          "Type": "string"
         }],
         "TotalCount": integer
```

```
        }],
        "SizeClassified": integer,
        "Status": {
         "Code": "string",
         "Reason": "string"
        }
       }
      },
      "Details": {
       "AwsAmazonMQBroker": {
        "AutoMinorVersionUpgrade": boolean,
        "BrokerArn": "string",
        "BrokerId": "string",
        "BrokerName": "string",
        "Configuration": {
         "Id": "string",
         "Revision": integer
        },
        "DeploymentMode": "string",
        "EncryptionOptions": {
         "UseAwsOwnedKey": boolean
        },
        "EngineType": "string",
        "EngineVersion": "string",
        "HostInstanceType": "string",
        "Logs": {
         "Audit": boolean,
         "AuditLogGroup": "string",
         "General": boolean,
         "GeneralLogGroup": "string"
        },
        "MaintenanceWindowStartTime": {
         "DayOfWeek": "string",
         "TimeOfDay": "string",
         "TimeZone": "string"
        },
        "PubliclyAccessible": boolean,
        "SecurityGroups": [
         "string"
        ],
        "StorageType": "string",
        "SubnetIds": [
         "string",
         "string"
```

```
        ],
        "Users": [{
         "Username": "string"
        }]
       },
       "AwsApiGatewayRestApi": {
        "ApiKeySource": "string",
        "BinaryMediaTypes": [" string"],
        "CreatedDate": "string",
        "Description": "string",
        "EndpointConfiguration": {
         "Types": ["string"]
        },
        "Id": "string",
        "MinimumCompressionSize": number,
        "Name": "string",
        "Version": "string"
       },
       "AwsApiGatewayStage": {
        "AccessLogSettings": {
         "DestinationArn": "string",
         "Format": "string"
        },
        "CacheClusterEnabled": boolean,
        "CacheClusterSize": "string",
        "CacheClusterStatus": "string",
        "CanarySettings": {
         "DeploymentId": "string",
         "PercentTraffic": number,
         "StageVariableOverrides": [{
          "string": "string"
         }],
         "UseStageCache": boolean
        },
        "ClientCertificateId": "string",
        "CreatedDate": "string",
        "DeploymentId": "string",
        "Description": "string",
        "DocumentationVersion": "string",
        "LastUpdatedDate": "string",
        "MethodSettings": [{
         "CacheDataEncrypted": boolean,
         "CachingEnabled": boolean,
         "CacheTtlInSeconds": number,
```

```
          "DataTraceEnabled": boolean,
          "HttpMethod": "string",
          "LoggingLevel": "string",
          "MetricsEnabled": boolean,
          "RequireAuthorizationForCacheControl": boolean,
          "ResourcePath": "string",
          "ThrottlingBurstLimit": number,
          "ThrottlingRateLimit": number,
          "UnauthorizedCacheControlHeaderStrategy": "string"
         }],
         "StageName": "string",
         "TracingEnabled": boolean,
         "Variables": {
          "string": "string"
         },
         "WebAclArn": "string"
        },
        "AwsApiGatewayV2Api": {
         "ApiEndpoint": "string",
         "ApiId": "string",
         "ApiKeySelectionExpression": "string",
         "CorsConfiguration": {
          "AllowCredentials": boolean,
          "AllowHeaders": ["string"],
          "AllowMethods": ["string"],
          "AllowOrigins": ["string"],
          "ExposeHeaders": ["string"],
          "MaxAge": number
         },
         "CreatedDate": "string",
         "Description": "string",
         "Name": "string",
         "ProtocolType": "string",
         "RouteSelectionExpression": "string",
         "Version": "string"
        },
        "AwsApiGatewayV2Stage": {
         "AccessLogSettings": {
          "DestinationArn": "string",
          "Format": "string"
         },
         "ApiGatewayManaged": boolean,
         "AutoDeploy": boolean,
         "ClientCertificateId": "string",
```

```
        "CreatedDate": "string",
        "DefaultRouteSettings": {
         "DataTraceEnabled": boolean,
         "DetailedMetricsEnabled": boolean,
         "LoggingLevel": "string",
         "ThrottlingBurstLimit": number,
         "ThrottlingRateLimit": number
        },
        "DeploymentId": "string",
        "Description": "string",
        "LastDeploymentStatusMessage": "string",
        "LastUpdatedDate": "string",
        "RouteSettings": {
         "DetailedMetricsEnabled": boolean,
         "LoggingLevel": "string",
         "DataTraceEnabled": boolean,
         "ThrottlingBurstLimit": number,
         "ThrottlingRateLimit": number
        },
        "StageName": "string",
        "StageVariables": [{
         "string": "string"
        }]
       },
       "AwsAppSyncGraphQLApi": {
        "AwsAppSyncGraphQlApi": {
         "AdditionalAuthenticationProviders": [
         {
          "AuthenticationType": "string",
          "LambdaAuthorizerConfig": {
           "AuthorizerResultTtlInSeconds": integer,
           "AuthorizerUri": "string"
          }
         },
         {
          "AuthenticationType": "string"
         }
         ],
         "ApiId": "string",
         "Arn": "string",
         "AuthenticationType": "string",
         "Id": "string",
         "LogConfig": {
          "CloudWatchLogsRoleArn": "string",
```

```
          "ExcludeVerboseContent": boolean,
          "FieldLogLevel": "string"
         },
         "Name": "string",
         "XrayEnabled": boolean
        }
       },
       "AwsAthenaWorkGroup": {
        "Description": "string",
        "Name": "string",
        "WorkgroupConfiguration": {
         "ResultConfiguration": {
          "EncryptionConfiguration": {
           "EncryptionOption": "string",
           "KmsKey": "string"
          }
         }
        },
        "State": "string"
       },
       "AwsAutoScalingAutoScalingGroup": {
        "AvailabilityZones": [{
         "Value": "string"
        }],
        "CreatedTime": "string",
        "HealthCheckGracePeriod": integer,
        "HealthCheckType": "string",
        "LaunchConfigurationName": "string",
        "LoadBalancerNames": ["string"],
        "LaunchTemplate": {
                       "LaunchTemplateId": "string",
                       "LaunchTemplateName": "string",
                       "Version": "string"
                   },
        "MixedInstancesPolicy": {
         "InstancesDistribution": {
          "OnDemandAllocationStrategy": "string",
          "OnDemandBaseCapacity": number,
          "OnDemandPercentageAboveBaseCapacity": number,
          "SpotAllocationStrategy": "string",
          "SpotInstancePools": number,
          "SpotMaxPrice": "string"
         },
         "LaunchTemplate": {
```

```
          "LaunchTemplateSpecification": {
           "LaunchTemplateId": "string",
           "LaunchTemplateName": "string",
           "Version": "string"
          },
          "CapacityRebalance": boolean,
          "Overrides": [{
           "InstanceType": "string",
           "WeightedCapacity": "string"
          }]
         }
        }
       },
       "AwsAutoScalingLaunchConfiguration": {
        "AssociatePublicIpAddress": boolean,
        "BlockDeviceMappings": [{
         "DeviceName": "string",
         "Ebs": {
          "DeleteOnTermination": boolean,
          "Encrypted": boolean,
          "Iops": number,
          "SnapshotId": "string",
          "VolumeSize": number,
          "VolumeType": "string"
         },
         "NoDevice": boolean,
         "VirtualName": "string"
        }],
        "ClassicLinkVpcId": "string",
        "ClassicLinkVpcSecurityGroups": ["string"],
        "CreatedTime": "string",
        "EbsOptimized": boolean,
        "IamInstanceProfile": "string"
       },
       "ImageId": "string",
       "InstanceMonitoring": {
        "Enabled": boolean
       },
       "InstanceType": "string",
       "KernelId": "string",
       "KeyName": "string",
       "LaunchConfigurationName": "string",
       "MetadataOptions": {
        "HttpEndPoint": "string",
```

```
        "HttpPutReponseHopLimit": number,
        "HttpTokens": "string"
       },
       "PlacementTenancy": "string",
       "RamdiskId": "string",
       "SecurityGroups": ["string"],
       "SpotPrice": "string",
       "UserData": "string"
      },
      "AwsBackupBackupPlan": {
       "BackupPlan": {
        "AdvancedBackupSettings": [{
         "BackupOptions": {
          "WindowsVSS":"string"
         },
         "ResourceType":"string"
        }],
        "BackupPlanName": "string",
        "BackupPlanRule": [{
         "CompletionWindowMinutes": integer,
         "CopyActions": [{
          "DestinationBackupVaultArn": "string",
          "Lifecycle": {
           "DeleteAfterDays": integer,
           "MoveToColdStorageAfterDays": integer
          }
         }],
         "Lifecycle": {
          "DeleteAfterDays": integer
         },
         "RuleName": "string",
         "ScheduleExpression": "string",
         "StartWindowMinutes": integer,
         "TargetBackupVault": "string"
        }]
       },
       "BackupPlanArn": "string",
       "BackupPlanId": "string",
       "VersionId": "string"
      },
      "AwsBackupBackupVault": {
       "AccessPolicy": {
        "Statement": [{
         "Action": ["string"],
```

```
          "Effect": "string",
          "Principal": {
           "AWS": "string"
          },
          "Resource": "string"
         }],
         "Version": "string"
        },
        "BackupVaultArn": "string",
        "BackupVaultName": "string",
        "EncryptionKeyArn": "string",
        "Notifications": {
         "BackupVaultEvents": ["string"],
         "SNSTopicArn": "string"
        }
       },
       "AwsBackupRecoveryPoint": {
        "BackupSizeInBytes": integer,
        "BackupVaultName": "string",
        "BackupVaultArn": "string",
        "CalculatedLifecycle": {
         "DeleteAt": "string",
         "MoveToColdStorageAt": "string"
        },
        "CompletionDate": "string",
        "CreatedBy": {
         "BackupPlanArn": "string",
         "BackupPlanId": "string",
         "BackupPlanVersion": "string",
         "BackupRuleId": "string"
        },
        "CreationDate": "string",
        "EncryptionKeyArn": "string",
        "IamRoleArn": "string",
        "IsEncrypted": boolean,
        "LastRestoreTime": "string",
        "Lifecycle": {
         "DeleteAfterDays": integer,
         "MoveToColdStorageAfterDays": integer
        },
        "RecoveryPointArn": "string",
        "ResourceArn": "string",
        "ResourceType": "string",
        "SourceBackupVaultArn": "string",
```

```
        "Status": "string",
        "StatusMessage": "string",
        "StorageClass": "string"
       },
       "AwsCertificateManagerCertificate": {
        "CertificateAuthorityArn": "string",
        "CreatedAt": "string",
        "DomainName": "string",
        "DomainValidationOptions": [{
         "DomainName": "string",
         "ResourceRecord": {
          "Name": "string",
          "Type": "string",
          "Value": "string"
         },
         "ValidationDomain": "string",
         "ValidationEmails": ["string"],
         "ValidationMethod": "string",
         "ValidationStatus": "string"
        }],
        "ExtendedKeyUsages": [{
         "Name": "string",
         "OId": "string"
        }],
        "FailureReason": "string",
        "ImportedAt": "string",
        "InUseBy": ["string"],
        "IssuedAt": "string",
        "Issuer": "string",
        "KeyAlgorithm": "string",
        "KeyUsages": [{
         "Name": "string"
        }],
        "NotAfter": "string",
        "NotBefore": "string",
        "Options": {
         "CertificateTransparencyLoggingPreference": "string"
        },
        "RenewalEligibility": "string",
        "RenewalSummary": {
         "DomainValidationOptions": [{
          "DomainName": "string",
          "ResourceRecord": {
           "Name": "string",
```

```
          "Type": "string",
          "Value": "string"
         },
         "ValidationDomain": "string",
         "ValidationEmails": ["string"],
         "ValidationMethod": "string",
         "ValidationStatus": "string"
        }],
        "RenewalStatus": "string",
        "RenewalStatusReason": "string",
        "UpdatedAt": "string"
       },
       "Serial": "string",
       "SignatureAlgorithm": "string",
       "Status": "string",
       "Subject": "string",
       "SubjectAlternativeNames": ["string"],
       "Type": "string"
      },
      "AwsCloudFormationStack": {
       "Capabilities": ["string"],
       "CreationTime": "string",
       "Description": "string",
       "DisableRollback": boolean,
       "DriftInformation": {
        "StackDriftStatus": "string"
       },
       "EnableTerminationProtection": boolean,
       "LastUpdatedTime": "string",
       "NotificationArns": ["string"],
       "Outputs": [{
        "Description": "string",
        "OutputKey": "string",
        "OutputValue": "string"
       }],
       "RoleArn": "string",
       "StackId": "string",
       "StackName": "string",
       "StackStatus": "string",
       "StackStatusReason": "string",
       "TimeoutInMinutes": number
      },
      "AwsCloudFrontDistribution": {
       "CacheBehaviors": {
```

```
        "Items": [{
         "ViewerProtocolPolicy": "string"
        }]
       },
       "DefaultCacheBehavior": {
        "ViewerProtocolPolicy": "string"
       },
       "DefaultRootObject": "string",
       "DomainName": "string",
       "Etag": "string",
       "LastModifiedTime": "string",
       "Logging": {
        "Bucket": "string",
        "Enabled": boolean,
        "IncludeCookies": boolean,
        "Prefix": "string"
       },
       "OriginGroups": {
        "Items": [{
         "FailoverCriteria": {
          "StatusCodes": {
           "Items": [number],
           "Quantity": number
          }
         }
        }]
       },
       "Origins": {
        "Items": [{
         "CustomOriginConfig": {
          "HttpPort": number,
          "HttpsPort": number,
          "OriginKeepaliveTimeout": number,
          "OriginProtocolPolicy": "string",
          "OriginReadTimeout": number,
          "OriginSslProtocols": {
           "Items": ["string"],
           "Quantity": number
          }
         },
         "DomainName": "string",
         "Id": "string",
         "OriginPath": "string",
         "S3OriginConfig": {
```

```
          "OriginAccessIdentity": "string"
         }
        }]
       },
       "Status": "string",
       "ViewerCertificate": {
        "AcmCertificateArn": "string",
        "Certificate": "string",
        "CertificateSource": "string",
        "CloudFrontDefaultCertificate": boolean,
        "IamCertificateId": "string",
        "MinimumProtocolVersion": "string",
        "SslSupportMethod": "string"
       },
       "WebAclId": "string"
      },
      "AwsCloudTrailTrail": {
       "CloudWatchLogsLogGroupArn": "string",
       "CloudWatchLogsRoleArn": "string",
       "HasCustomEventSelectors": boolean,
       "HomeRegion": "string",
       "IncludeGlobalServiceEvents": boolean,
       "IsMultiRegionTrail": boolean,
       "IsOrganizationTrail": boolean,
       "KmsKeyId": "string",
       "LogFileValidationEnabled": boolean,
       "Name": "string",
       "S3BucketName": "string",
       "S3KeyPrefix": "string",
       "SnsTopicArn": "string",
       "SnsTopicName": "string",
       "TrailArn": "string"
      },
      "AwsCloudWatchAlarm": {
       "ActionsEnabled": boolean,
       "AlarmActions": ["string"],
       "AlarmArn": "string",
       "AlarmConfigurationUpdatedTimestamp": "string",
       "AlarmDescription": "string",
       "AlarmName": "string",
       "ComparisonOperator": "string",
       "DatapointsToAlarm": number,
       "Dimensions": [{
        "Name": "string",
```

```
        "Value": "string"
       }],
       "EvaluateLowSampleCountPercentile": "string",
       "EvaluationPeriods": number,
       "ExtendedStatistic": "string",
       "InsufficientDataActions": ["string"],
       "MetricName": "string",
       "Namespace": "string",
       "OkActions": ["string"],
       "Period": number,
       "Statistic": "string",
       "Threshold": number,
       "ThresholdMetricId": "string",
       "TreatMissingData": "string",
       "Unit": "string"
      },
      "AwsCodeBuildProject": {
       "Artifacts": [{
        "ArtifactIdentifier": "string",
        "EncryptionDisabled": boolean,
        "Location": "string",
        "Name": "string",
        "NamespaceType": "string",
        "OverrideArtifactName": boolean,
        "Packaging": "string",
        "Path": "string",
        "Type": "string"
       }],
       "SecondaryArtifacts": [{
                   "ArtifactIdentifier": "string",
                   "Type": "string",
                   "Location": "string",
                   "Name": "string",
                   "NamespaceType": "string",
                   "Packaging": "string",
                   "Path": "string",
                   "EncryptionDisabled": boolean,
                   "OverrideArtifactName": boolean
               }],
       "EncryptionKey": "string",
       "Certificate": "string",
       "Environment": {
        "Certificate": "string",
        "EnvironmentVariables": [{
```

```
          "Name": "string",
          "Type": "string",
          "Value": "string"
         }],
         "ImagePullCredentialsType": "string",
         "PrivilegedMode": boolean,
         "RegistryCredential": {
          "Credential": "string",
          "CredentialProvider": "string"
         },
         "Type": "string"
        },
        "LogsConfig": {
         "CloudWatchLogs": {
          "GroupName": "string",
          "Status": "string",
          "StreamName": "string"
         },
         "S3Logs": {
          "EncryptionDisabled": boolean,
          "Location": "string",
          "Status": "string"
         }
        },
        "Name": "string",
        "ServiceRole": "string",
        "Source": {
         "Type": "string",
         "Location": "string",
         "GitCloneDepth": integer
        },
        "VpcConfig": {
         "VpcId": "string",
         "Subnets": ["string"],
         "SecurityGroupIds": ["string"]
        }
       },
       "AwsDmsEndpoint": {
        "CertificateArn": "string",
        "DatabaseName": "string",
        "EndpointArn": "string",
        "EndpointIdentifier": "string",
        "EndpointType": "string",
        "EngineName": "string",
```

```
        "KmsKeyId": "string",
        "Port": integer,
        "ServerName": "string",
        "SslMode": "string",
        "Username": "string"
      },
      "AwsDmsReplicationInstance": {
        "AllocatedStorage": integer,
        "AutoMinorVersionUpgrade": boolean,
        "AvailabilityZone": "string",
        "EngineVersion": "string",
        "KmsKeyId": "string",
        "MultiAZ": boolean,
        "PreferredMaintenanceWindow": "string",
        "PubliclyAccessible": boolean,
        "ReplicationInstanceClass": "string",
        "ReplicationInstanceIdentifier": "string",
        "ReplicationSubnetGroup": {
            "ReplicationSubnetGroupIdentifier": "string"
        },
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "string"
            }
        ]
      },
      "AwsDmsReplicationTask": {
        "CdcStartPosition": "string",
        "Id": "string",
        "MigrationType": "string",
        "ReplicationInstanceArn": "string",
        "ReplicationTaskIdentifier": "string",
        "ReplicationTaskSettings": {
          "string": "string"
        },
        "SourceEndpointArn": "string",
        "TableMappings": {
          "string": "string"
        },
        "TargetEndpointArn": "string"
      },
      "AwsDynamoDbTable": {
        "AttributeDefinitions": [{
          "AttributeName": "string",
```

```
        "AttributeType": "string"
       }],
       "BillingModeSummary": {
        "BillingMode": "string",
        "LastUpdateToPayPerRequestDateTime": "string"
       },
       "CreationDateTime": "string",
       "DeletionProtectionEnabled": boolean,
       "GlobalSecondaryIndexes": [{
        "Backfilling": boolean,
        "IndexArn": "string",
        "IndexName": "string",
        "IndexSizeBytes": number,
        "IndexStatus": "string",
        "ItemCount": number,
        "KeySchema": [{
         "AttributeName": "string",
         "KeyType": "string"
        }],
        "Projection": {
         "NonKeyAttributes": ["string"],
         "ProjectionType": "string"
        },
        "ProvisionedThroughput": {
         "LastDecreaseDateTime": "string",
         "LastIncreaseDateTime": "string",
         "NumberOfDecreasesToday": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
        }
       }],
       "GlobalTableVersion": "string",
       "ItemCount": number,
       "KeySchema": [{
        "AttributeName": "string",
        "KeyType": "string"
       }],
       "LatestStreamArn": "string",
       "LatestStreamLabel": "string",
       "LocalSecondaryIndexes": [{
        "IndexArn": "string",
        "IndexName": "string",
        "KeySchema": [{
         "AttributeName": "string",
```

```
          "KeyType": "string"
         }],
         "Projection": {
          "NonKeyAttributes": ["string"],
          "ProjectionType": "string"
         }
        }],
        "ProvisionedThroughput": {
         "LastDecreaseDateTime": "string",
         "LastIncreaseDateTime": "string",
         "NumberOfDecreasesToday": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
        },
        "Replicas": [{
         "GlobalSecondaryIndexes": [{
          "IndexName": "string",
          "ProvisionedThroughputOverride": {
           "ReadCapacityUnits": number
          }
         }],
         "KmsMasterKeyId": "string",
         "ProvisionedThroughputOverride": {
          "ReadCapacityUnits": number
         },
         "RegionName": "string",
         "ReplicaStatus": "string",
         "ReplicaStatusDescription": "string"
        }],
        "RestoreSummary": {
         "RestoreDateTime": "string",
         "RestoreInProgress": boolean,
         "SourceBackupArn": "string",
         "SourceTableArn": "string"
        },
        "SseDescription": {
         "InaccessibleEncryptionDateTime": "string",
         "KmsMasterKeyArn": "string",
         "SseType": "string",
         "Status": "string"
        },
        "StreamSpecification": {
         "StreamEnabled": boolean,
         "StreamViewType": "string"
```

```
        },
        "TableId": "string",
        "TableName": "string",
        "TableSizeBytes": number,
        "TableStatus": "string"
      },
      "AwsEc2ClientVpnEndpoint": {
        "AuthenticationOptions": [
         {
          "MutualAuthentication": {
           "ClientRootCertificateChainArn": "string"
          },
          "Type": "string"
         }
        ],
        "ClientCidrBlock": "string",
        "ClientConnectOptions": {
         "Enabled": boolean
        },
        "ClientLoginBannerOptions": {
         "Enabled": boolean
        },
        "ClientVpnEndpointId": "string",
        "ConnectionLogOptions": {
         "Enabled": boolean
        },
        "Description": "string",
        "DnsServer": ["string"],
        "ServerCertificateArn": "string",
        "SecurityGroupIdSet": [
         "string"
        ],
        "SelfServicePortalUrl": "string",
        "SessionTimeoutHours": "integer",
        "SplitTunnel": boolean,
        "TransportProtocol": "string",
        "VpcId": "string",
        "VpnPort": integer
      },
      "AwsEc2Eip": {
        "AllocationId": "string",
        "AssociationId": "string",
        "Domain": "string",
        "InstanceId": "string",
```

```
        "NetworkBorderGroup": "string",
        "NetworkInterfaceId": "string",
        "NetworkInterfaceOwnerId": "string",
        "PrivateIpAddress": "string",
        "PublicIp": "string",
        "PublicIpv4Pool": "string"
       },
       "AwsEc2Instance": {
        "IamInstanceProfileArn": "string",
        "ImageId": "string",
        "IpV4Addresses": ["string"],
        "IpV6Addresses": ["string"],
        "KeyName": "string",
        "LaunchedAt": "string",
        "MetadataOptions": {
         "HttpEndpoint": "string",
         "HttpProtocolIpv6": "string",
         "HttpPutResponseHopLimit": number,
         "HttpTokens": "string",
         "InstanceMetadataTags": "string"
        },
        "Monitoring": {
         "State": "string"
        },
        "NetworkInterfaces": [{
         "NetworkInterfaceId": "string"
        }],
        "SubnetId": "string",
        "Type": "string",
        "VirtualizationType": "string",
        "VpcId": "string"
       },
       "AwsEc2LaunchTemplate": {
        "DefaultVersionNumber": "string",
        "ElasticGpuSpecifications": ["string"],
        "ElasticInferenceAccelerators": ["string"],
        "Id": "string",
        "ImageId": "string",
        "LatestVersionNumber": "string",
        "LaunchTemplateData": {
         "BlockDeviceMappings": [{
          "DeviceName": "string",
          "Ebs": {
           "DeleteonTermination": boolean,
```

```
          "Encrypted": boolean,
          "SnapshotId": "string",
          "VolumeSize": number,
          "VolumeType": "string"
         }
        }],
        "MetadataOptions": {
         "HttpTokens": "string",
         "HttpPutResponseHopLimit" : number
        },
        "Monitoring": {
         "Enabled": boolean
        },
        "NetworkInterfaces": [{
         "AssociatePublicIpAddress" : boolean
        }]
       },
       "LaunchTemplateName": "string",
       "LicenseSpecifications": ["string"],
       "SecurityGroupIds": ["string"],
       "SecurityGroups": ["string"],
       "TagSpecifications": ["string"]
      },
      "AwsEc2NetworkAcl": {
       "Associations": [{
        "NetworkAclAssociationId": "string",
        "NetworkAclId": "string",
        "SubnetId": "string"
       }],
       "Entries": [{
        "CidrBlock": "string",
        "Egress": boolean,
        "IcmpTypeCode": {
         "Code": number,
         "Type": number
        },
        "Ipv6CidrBlock": "string",
        "PortRange": {
         "From": number,
         "To": number
        },
        "Protocol": "string",
        "RuleAction": "string",
        "RuleNumber": number
```

```
        }],
        "IsDefault": boolean,
        "NetworkAclId": "string",
        "OwnerId": "string",
        "VpcId": "string"
       },
       "AwsEc2NetworkInterface": {
        "Attachment": {
         "AttachmentId": "string",
         "AttachTime": "string",
         "DeleteOnTermination": boolean,
         "DeviceIndex": number,
         "InstanceId": "string",
         "InstanceOwnerId": "string",
         "Status": "string"
        },
        "Ipv6Addresses": [{
         "Ipv6Address": "string"
        }],
        "NetworkInterfaceId": "string",
        "PrivateIpAddresses": [{
         "PrivateDnsName": "string",
         "PrivateIpAddress": "string"
        }],
        "PublicDnsName": "string",
        "PublicIp": "string",
        "SecurityGroups": [{
         "GroupId": "string",
         "GroupName": "string"
        }],
        "SourceDestCheck": boolean
       },
       "AwsEc2RouteTable": {
        "AssociationSet": [{
         "AssociationState": {
          "State": "string"
         },
         "Main": boolean,
         "RouteTableAssociationId": "string",
         "RouteTableId": "string"
        }],
        "PropogatingVgwSet": [],
        "RouteTableId": "string",
        "RouteSet": [
```

```
        {
         "DestinationCidrBlock": "string",
         "GatewayId": "string",
         "Origin": "string",
         "State": "string"
        },
        {
         "DestinationCidrBlock": "string",
         "GatewayId": "string",
         "Origin": "string",
         "State": "string"
        }
       ],
       "VpcId": "string"
      },
      "AwsEc2SecurityGroup": {
       "GroupId": "string",
       "GroupName": "string",
       "IpPermissions": [{
        "FromPort": number,
        "IpProtocol": "string",
        "IpRanges": [{
         "CidrIp": "string"
        }],
        "Ipv6Ranges": [{
         "CidrIpv6": "string"
        }],
        "PrefixListIds": [{
         "PrefixListId": "string"
        }],
        "ToPort": number,
        "UserIdGroupPairs": [{
         "GroupId": "string",
         "GroupName": "string",
         "PeeringStatus": "string",
         "UserId": "string",
         "VpcId": "string",
         "VpcPeeringConnectionId": "string"
        }]
       }],
       "IpPermissionsEgress": [{
        "FromPort": number,
        "IpProtocol": "string",
        "IpRanges": [{
```

```
          "CidrIp": "string"
         }],
         "Ipv6Ranges": [{
          "CidrIpv6": "string"
         }],
         "PrefixListIds": [{
          "PrefixListId": "string"
         }],
         "ToPort": number,
         "UserIdGroupPairs": [{
          "GroupId": "string",
          "GroupName": "string",
          "PeeringStatus": "string",
          "UserId": "string",
          "VpcId": "string",
          "VpcPeeringConnectionId": "string"
         }]
        }],
        "OwnerId": "string",
        "VpcId": "string"
       },
       "AwsEc2Subnet": {
        "AssignIpv6AddressOnCreation": boolean,
        "AvailabilityZone": "string",
        "AvailabilityZoneId": "string",
        "AvailableIpAddressCount": number,
        "CidrBlock": "string",
        "DefaultForAz": boolean,
        "Ipv6CidrBlockAssociationSet": [{
         "AssociationId": "string",
         "Ipv6CidrBlock": "string",
         "CidrBlockState": "string"
        }],
        "MapPublicIpOnLaunch": boolean,
        "OwnerId": "string",
        "State": "string",
        "SubnetArn": "string",
        "SubnetId": "string",
        "VpcId": "string"
       },
       "AwsEc2TransitGateway": {
        "AmazonSideAsn": number,
        "AssociationDefaultRouteTableId": "string",
        "AutoAcceptSharedAttachments": "string",
```

```
        "DefaultRouteTableAssociation": "string",
        "DefaultRouteTablePropagation": "string",
        "Description": "string",
        "DnsSupport": "string",
        "Id": "string",
        "MulticastSupport": "string",
        "PropagationDefaultRouteTableId": "string",
        "TransitGatewayCidrBlocks": ["string"],
        "VpnEcmpSupport": "string"
      },
      "AwsEc2Volume": {
        "Attachments": [{
          "AttachTime": "string",
          "DeleteOnTermination": boolean,
          "InstanceId": "string",
          "Status": "string"
        }],
        "CreateTime": "string",
        "DeviceName": "string",
        "Encrypted": boolean,
        "KmsKeyId": "string",
        "Size": number,
        "SnapshotId": "string",
        "Status": "string",
        "VolumeId": "string",
        "VolumeScanStatus": "string",
        "VolumeType": "string"
      },
      "AwsEc2Vpc": {
        "CidrBlockAssociationSet": [{
          "AssociationId": "string",
          "CidrBlock": "string",
          "CidrBlockState": "string"
        }],
        "DhcpOptionsId": "string",
        "Ipv6CidrBlockAssociationSet": [{
          "AssociationId": "string",
          "CidrBlockState": "string",
          "Ipv6CidrBlock": "string"
        }],
        "State": "string"
      },
      "AwsEc2VpcEndpointService": {
        "AcceptanceRequired": boolean,
```

```
        "AvailabilityZones": ["string"],
        "BaseEndpointDnsNames": ["string"],
        "ManagesVpcEndpoints": boolean,
        "GatewayLoadBalancerArns": ["string"],
        "NetworkLoadBalancerArns": ["string"],
        "PrivateDnsName": "string",
        "ServiceId": "string",
        "ServiceName": "string",
        "ServiceState": "string",
        "ServiceType": [{
         "ServiceType": "string"
        }]
       },
       "AwsEc2VpcPeeringConnection": {
        "AccepterVpcInfo": {
         "CidrBlock": "string",
         "CidrBlockSet": [{
          "CidrBlock": "string"
         }],
         "Ipv6CidrBlockSet": [{
          "Ipv6CidrBlock": "string"
         }],
         "OwnerId": "string",
         "PeeringOptions": {
          "AllowDnsResolutionFromRemoteVpc": boolean,
          "AllowEgressFromLocalClassicLinkToRemoteVpc": boolean,
          "AllowEgressFromLocalVpcToRemoteClassicLink": boolean
         },
         "Region": "string",
         "VpcId": "string"
        },
        "ExpirationTime": "string",
        "RequesterVpcInfo": {
         "CidrBlock": "string",
         "CidrBlockSet": [{
          "CidrBlock": "string"
         }],
         "Ipv6CidrBlockSet": [{
          "Ipv6CidrBlock": "string"
         }],
         "OwnerId": "string",
         "PeeringOptions": {
          "AllowDnsResolutionFromRemoteVpc": boolean,
          "AllowEgressFromLocalClassicLinkToRemoteVpc": boolean,
```

```
          "AllowEgressFromLocalVpcToRemoteClassicLink": boolean
         },
         "Region": "string",
         "VpcId": "string"
        },
        "Status": {
         "Code": "string",
         "Message": "string"
        },
        "VpcPeeringConnectionId": "string"
       },
       "AwsEcrContainerImage": {
        "Architecture": "string",
        "ImageDigest": "string",
        "ImagePublishedAt": "string",
        "ImageTags": ["string"],
        "RegistryId": "string",
        "RepositoryName": "string"
       },
       "AwsEcrRepository": {
        "Arn": "string",
        "ImageScanningConfiguration": {
         "ScanOnPush": boolean
        },
        "ImageTagMutability": "string",
        "LifecyclePolicy": {
         "LifecyclePolicyText": "string",
         "RegistryId": "string"
        },
        "RepositoryName": "string",
        "RepositoryPolicyText": "string"
       },
       "AwsEcsCluster": {
        "ActiveServicesCount": number,
        "CapacityProviders": ["string"],
        "ClusterArn": "string",
        "ClusterName": "string",
        "ClusterSettings": [{
         "Name": "string",
         "Value": "string"
        }],
        "Configuration": {
         "ExecuteCommandConfiguration": {
          "KmsKeyId": "string",
```

```
          "LogConfiguration": {
           "CloudWatchEncryptionEnabled": boolean,
           "CloudWatchLogGroupName": "string",
           "S3BucketName": "string",
           "S3EncryptionEnabled": boolean,
           "S3KeyPrefix": "string"
          },
          "Logging": "string"
         }
        },
        "DefaultCapacityProviderStrategy": [{
         "Base": number,
         "CapacityProvider": "string",
         "Weight": number
        }],
        "RegisteredContainerInstancesCount": number,
        "RunningTasksCount": number,
        "Status": "string"
       },
       "AwsEcsContainer": {
        "Image": "string",
        "MountPoints": [{
         "ContainerPath": "string",
         "SourceVolume": "string"
        }],
        "Name": "string",
        "Privileged": boolean
       },
       "AwsEcsService": {
        "CapacityProviderStrategy": [{
         "Base": number,
         "CapacityProvider": "string",
         "Weight": number
        }],
        "Cluster": "string",
        "DeploymentConfiguration": {
         "DeploymentCircuitBreaker": {
          "Enable": boolean,
          "Rollback": boolean
         },
         "MaximumPercent": number,
         "MinimumHealthyPercent": number
        },
        "DeploymentController": {
```

```
        "Type": "string"
       },
       "DesiredCount": number,
       "EnableEcsManagedTags": boolean,
       "EnableExecuteCommand": boolean,
       "HealthCheckGracePeriodSeconds": number,
       "LaunchType": "string",
       "LoadBalancers": [{
        "ContainerName": "string",
        "ContainerPort": number,
        "LoadBalancerName": "string",
        "TargetGroupArn": "string"
       }],
       "Name": "string",
       "NetworkConfiguration": {
        "AwsVpcConfiguration": {
         "AssignPublicIp": "string",
         "SecurityGroups": ["string"],
         "Subnets": ["string"]
        }
       },
       "PlacementConstraints": [{
        "Expression": "string",
        "Type": "string"
       }],
       "PlacementStrategies": [{
        "Field": "string",
        "Type": "string"
       }],
       "PlatformVersion": "string",
       "PropagateTags": "string",
       "Role": "string",
       "SchedulingStrategy": "string",
       "ServiceArn": "string",
       "ServiceName": "string",
       "ServiceRegistries": [{
        "ContainerName": "string",
        "ContainerPort": number,
        "Port": number,
        "RegistryArn": "string"
       }],
       "TaskDefinition": "string"
      },
      "AwsEcsTask": {
```

```
        "CreatedAt": "string",
        "ClusterArn": "string",
        "Group": "string",
        "StartedAt": "string",
        "StartedBy": "string",
        "TaskDefinitionArn": "string",
        "Version": number,
        "Volumes": [{
         "Name": "string",
         "Host": {
          "SourcePath": "string"
         }
        }],
        "Containers": [{
         "Image": "string",
         "MountPoints": [{
          "ContainerPath": "string",
          "SourceVolume": "string"
         }],
         "Name": "string",
         "Privileged": boolean
        }]
       },
       "AwsEcsTaskDefinition": {
        "ContainerDefinitions": [{
         "Command": ["string"],
         "Cpu": number,
         "DependsOn": [{
          "Condition": "string",
          "ContainerName": "string"
         }],
         "DisableNetworking": boolean,
         "DnsSearchDomains": ["string"],
         "DnsServers": ["string"],
         "DockerLabels": {
          "string": "string"
         },
         "DockerSecurityOptions": ["string"],
         "EntryPoint": ["string"],
         "Environment": [{
          "Name": "string",
          "Value": "string"
         }],
         "EnvironmentFiles": [{
```

```
          "Type": "string",
          "Value": "string"
        }],
        "Essential": boolean,
        "ExtraHosts": [{
          "Hostname": "string",
          "IpAddress": "string"
        }],
        "FirelensConfiguration": {
          "Options": {
            "string": "string"
          },
          "Type": "string"
        },
        "HealthCheck": {
          "Command": ["string"],
          "Interval": number,
          "Retries": number,
          "StartPeriod": number,
          "Timeout": number
        },
        "Hostname": "string",
        "Image": "string",
        "Interactive": boolean,
        "Links": ["string"],
        "LinuxParameters": {
          "Capabilities": {
            "Add": ["string"],
            "Drop": ["string"]
          },
          "Devices": [{
            "ContainerPath": "string",
            "HostPath": "string",
            "Permissions": ["string"]
          }],
          "InitProcessEnabled": boolean,
          "MaxSwap": number,
          "SharedMemorySize": number,
          "Swappiness": number,
          "Tmpfs": [{
            "ContainerPath": "string",
            "MountOptions": ["string"],
            "Size": number
          }]
```

```
        },
        "LogConfiguration": {
         "LogDriver": "string",
         "Options": {
          "string": "string"
         },
         "SecretOptions": [{
          "Name": "string",
          "ValueFrom": "string"
         }]
        },
        "Memory": number,
        "MemoryReservation": number,
        "MountPoints": [{
         "ContainerPath": "string",
         "ReadOnly": boolean,
         "SourceVolume": "string"
        }],
        "Name": "string",
        "PortMappings": [{
         "ContainerPort": number,
         "HostPort": number,
         "Protocol": "string"
        }],
        "Privileged": boolean,
        "PseudoTerminal": boolean,
        "ReadonlyRootFilesystem": boolean,
        "RepositoryCredentials": {
         "CredentialsParameter": "string"
        },
        "ResourceRequirements": [{
         "Type": "string",
         "Value": "string"
        }],
        "Secrets": [{
         "Name": "string",
         "ValueFrom": "string"
        }],
        "StartTimeout": number,
        "StopTimeout": number,
        "SystemControls": [{
         "Namespace": "string",
         "Value": "string"
        }],
```

```
        "Ulimits": [{
         "HardLimit": number,
         "Name": "string",
         "SoftLimit": number
        }],
        "User": "string",
        "VolumesFrom": [{
         "ReadOnly": boolean,
         "SourceContainer": "string"
        }],
        "WorkingDirectory": "string"
       }],
       "Cpu": "string",
       "ExecutionRoleArn": "string",
       "Family": "string",
       "InferenceAccelerators": [{
        "DeviceName": "string",
        "DeviceType": "string"
       }],
       "IpcMode": "string",
       "Memory": "string",
       "NetworkMode": "string",
       "PidMode": "string",
       "PlacementConstraints": [{
        "Expression": "string",
        "Type": "string"
       }],
       "ProxyConfiguration": {
        "ContainerName": "string",
        "ProxyConfigurationProperties": [{
         "Name": "string",
         "Value": "string"
        }],
        "Type": "string"
       },
       "RequiresCompatibilities": ["string"],
       "Status": "string",
       "TaskRoleArn": "string",
       "Volumes": [{
        "DockerVolumeConfiguration": {
         "Autoprovision": boolean,
         "Driver": "string",
         "DriverOpts": {
          "string": "string"
```

```
          },
          "Labels": {
           "string": "string"
          },
          "Scope": "string"
         },
         "EfsVolumeConfiguration": {
          "AuthorizationConfig": {
           "AccessPointId": "string",
           "Iam": "string"
          },
          "FilesystemId": "string",
          "RootDirectory": "string",
          "TransitEncryption": "string",
          "TransitEncryptionPort": number
         },
         "Host": {
          "SourcePath": "string"
         },
         "Name": "string"
        }]
       },
       "AwsEfsAccessPoint": {
        "AccessPointId": "string",
        "Arn": "string",
        "ClientToken": "string",
        "FileSystemId": "string",
        "PosixUser": {
         "Gid": "string",
         "SecondaryGids": ["string"],
         "Uid": "string"
        },
        "RootDirectory": {
         "CreationInfo": {
          "OwnerGid": "string",
          "OwnerUid": "string",
          "Permissions": "string"
         },
         "Path": "string"
        }
       },
       "AwsEksCluster": {
        "Arn": "string",
        "CertificateAuthorityData": "string",
```

```
        "ClusterStatus": "string",
        "Endpoint": "string",
        "Logging": {
         "ClusterLogging": [{
          "Enabled": boolean,
          "Types": ["string"]
         }]
        },
        "Name": "string",
        "ResourcesVpcConfig": {
         "EndpointPublicAccess": boolean,
         "SecurityGroupIds": ["string"],
         "SubnetIds": ["string"]
        },
        "RoleArn": "string",
        "Version": "string"
       },
       "AwsElasticBeanstalkEnvironment": {
        "ApplicationName": "string",
        "Cname": "string",
        "DateCreated": "string",
        "DateUpdated": "string",
        "Description": "string",
        "EndpointUrl": "string",
        "EnvironmentArn": "string",
        "EnvironmentId": "string",
        "EnvironmentLinks": [{
         "EnvironmentName": "string",
         "LinkName": "string"
        }],
        "EnvironmentName": "string",
        "OptionSettings": [{
         "Namespace": "string",
         "OptionName": "string",
         "ResourceName": "string",
         "Value": "string"
        }],
        "PlatformArn": "string",
        "SolutionStackName": "string",
        "Status": "string",
        "Tier": {
         "Name": "string",
         "Type": "string",
         "Version": "string"
```

```
        },
        "VersionLabel": "string"
       },
       "AwsElasticSearchDomain": {
        "AccessPolicies": "string",
        "DomainStatus": {
         "DomainId": "string",
         "DomainName": "string",
         "Endpoint": "string",
         "Endpoints": {
          "string": "string"
         }
        },
        "DomainEndpointOptions": {
         "EnforceHTTPS": boolean,
         "TLSSecurityPolicy": "string"
        },
        "ElasticsearchClusterConfig": {
         "DedicatedMasterCount": number,
         "DedicatedMasterEnabled": boolean,
         "DedicatedMasterType": "string",
         "InstanceCount": number,
         "InstanceType": "string",
         "ZoneAwarenessConfig": {
          "AvailabilityZoneCount": number
         },
         "ZoneAwarenessEnabled": boolean
        },
        "ElasticsearchVersion": "string",
        "EncryptionAtRestOptions": {
         "Enabled": boolean,
         "KmsKeyId": "string"
        },
        "LogPublishingOptions": {
         "AuditLogs": {
          "CloudWatchLogsLogGroupArn": "string",
          "Enabled": boolean
         },
         "IndexSlowLogs": {
          "CloudWatchLogsLogGroupArn": "string",
          "Enabled": boolean
         },
         "SearchSlowLogs": {
          "CloudWatchLogsLogGroupArn": "string",
```

```
          "Enabled": boolean
         }
        },
        "NodeToNodeEncryptionOptions": {
         "Enabled": boolean
        },
        "ServiceSoftwareOptions": {
         "AutomatedUpdateDate": "string",
         "Cancellable": boolean,
         "CurrentVersion": "string",
         "Description": "string",
         "NewVersion": "string",
         "UpdateAvailable": boolean,
         "UpdateStatus": "string"
        },
        "VPCOptions": {
         "AvailabilityZones": [
          "string"
         ],
         "SecurityGroupIds": [
          "string"
         ],
         "SubnetIds": [
          "string"
         ],
         "VPCId": "string"
        }
       },
       "AwsElbLoadBalancer": {
        "AvailabilityZones": ["string"],
        "BackendServerDescriptions": [{
         "InstancePort": number,
         "PolicyNames": ["string"]
        }],
        "CanonicalHostedZoneName": "string",
        "CanonicalHostedZoneNameID": "string",
        "CreatedTime": "string",
        "DnsName": "string",
        "HealthCheck": {
         "HealthyThreshold": number,
         "Interval": number,
         "Target": "string",
         "Timeout": number,
         "UnhealthyThreshold": number
```

```
        },
        "Instances": [{
         "InstanceId": "string"
        }],
        "ListenerDescriptions": [{
         "Listener": {
          "InstancePort": number,
          "InstanceProtocol": "string",
          "LoadBalancerPort": number,
          "Protocol": "string",
          "SslCertificateId": "string"
         },
         "PolicyNames": ["string"]
        }],
        "LoadBalancerAttributes": {
         "AccessLog": {
          "EmitInterval": number,
          "Enabled": boolean,
          "S3BucketName": "string",
          "S3BucketPrefix": "string"
         },
         "ConnectionDraining": {
          "Enabled": boolean,
          "Timeout": number
         },
         "ConnectionSettings": {
          "IdleTimeout": number
         },
         "CrossZoneLoadBalancing": {
          "Enabled": boolean
         },
         "AdditionalAttributes": [{
                        "Key": "string",
                        "Value": "string"
                    }]
        },
        "LoadBalancerName": "string",
        "Policies": {
         "AppCookieStickinessPolicies": [{
          "CookieName": "string",
          "PolicyName": "string"
         }],
         "LbCookieStickinessPolicies": [{
          "CookieExpirationPeriod": number,
```

```
          "PolicyName": "string"
         }],
         "OtherPolicies": ["string"]
        },
        "Scheme": "string",
        "SecurityGroups": ["string"],
        "SourceSecurityGroup": {
         "GroupName": "string",
         "OwnerAlias": "string"
        },
        "Subnets": ["string"],
        "VpcId": "string"
       },
       "AwsElbv2LoadBalancer": {
        "AvailabilityZones": {
         "SubnetId": "string",
         "ZoneName": "string"
        },
        "CanonicalHostedZoneId": "string",
        "CreatedTime": "string",
        "DNSName": "string",
        "IpAddressType": "string",
        "LoadBalancerAttributes": [{
         "Key": "string",
         "Value": "string"
        }],
        "Scheme": "string",
        "SecurityGroups": ["string"],
        "State": {
         "Code": "string",
         "Reason": "string"
        },
        "Type": "string",
        "VpcId": "string"
       },
       "AwsEventSchemasRegistry": {
        "Description": "string",
        "RegistryArn": "string",
        "RegistryName": "string"
       },
       "AwsEventsEndpoint": {
        "Arn": "string",
        "Description": "string",
        "EndpointId": "string",
```

```
        "EndpointUrl": "string",
        "EventBuses": [
            {
                "EventBusArn": "string"
            },
            {
                "EventBusArn": "string"
            }
        ],
        "Name": "string",
        "ReplicationConfig": {
            "State": "string"
        },
        "RoleArn": "string",
        "RoutingConfig": {
            "FailoverConfig": {
                "Primary": {
                    "HealthCheck": "string"
                },
                "Secondary": {
                    "Route": "string"
                }
            }
        },
        "State": "string"
      },
      "AwsEventsEventBus": {
        "Arn": "string",
        "Name": "string",
        "Policy": "string"
      },
      "AwsGuardDutyDetector": {
        "FindingPublishingFrequency": "string",
        "ServiceRole": "string",
        "Status": "string",
        "DataSources": {
          "CloudTrail": {
            "Status": "string"
          },
          "DnsLogs": {
            "Status": "string"
          },
          "FlowLogs": {
            "Status": "string"
```

```
        },
        "S3Logs": {
         "Status": "string"
        },
        "Kubernetes": {
         "AuditLogs": {
          "Status": "string"
         }
        },
        "MalwareProtection": {
         "ScanEc2InstanceWithFindings": {
          "EbsVolumes": {
           "Status": "string"
          }
         },
         "ServiceRole": "string"
        }
       }
      },
      "AwsIamAccessKey": {
       "AccessKeyId": "string",
       "AccountId": "string",
       "CreatedAt": "string",
       "PrincipalId": "string",
       "PrincipalName": "string",
       "PrincipalType": "string",
       "SessionContext": {
        "Attributes": {
         "CreationDate": "string",
         "MfaAuthenticated": boolean
        },
        "SessionIssuer": {
         "AccountId": "string",
         "Arn": "string",
         "PrincipalId": "string",
         "Type": "string",
         "UserName": "string"
        }
       },
       "Status": "string"
      },
      "AwsIamGroup": {
       "AttachedManagedPolicies": [{
        "PolicyArn": "string",
```

```
        "PolicyName": "string"
       }],
       "CreateDate": "string",
       "GroupId": "string",
       "GroupName": "string",
       "GroupPolicyList": [{
        "PolicyName": "string"
       }],
       "Path": "string"
      },
      "AwsIamPolicy": {
       "AttachmentCount": number,
       "CreateDate": "string",
       "DefaultVersionId": "string",
       "Description": "string",
       "IsAttachable": boolean,
       "Path": "string",
       "PermissionsBoundaryUsageCount": number,
       "PolicyId": "string",
       "PolicyName": "string",
       "PolicyVersionList": [{
        "CreateDate": "string",
        "IsDefaultVersion": boolean,
        "VersionId": "string"
       }],
       "UpdateDate": "string"
      },
      "AwsIamRole": {
       "AssumeRolePolicyDocument": "string",
       "AttachedManagedPolicies": [{
        "PolicyArn": "string",
        "PolicyName": "string"
       }],
       "CreateDate": "string",
       "InstanceProfileList": [{
        "Arn": "string",
        "CreateDate": "string",
        "InstanceProfileId": "string",
        "InstanceProfileName": "string",
        "Path": "string",
        "Roles": [{
         "Arn": "string",
         "AssumeRolePolicyDocument": "string",
         "CreateDate": "string",
```

```
          "Path": "string",
          "RoleId": "string",
          "RoleName": "string"
         }]
        }],
        "MaxSessionDuration": number,
        "Path": "string",
        "PermissionsBoundary": {
         "PermissionsBoundaryArn": "string",
         "PermissionsBoundaryType": "string"
        },
        "RoleId": "string",
        "RoleName": "string",
        "RolePolicyList": [{
         "PolicyName": "string"
        }]
       },
       "AwsIamUser": {
        "AttachedManagedPolicies": [{
         "PolicyArn": "string",
         "PolicyName": "string"
        }],
        "CreateDate": "string",
        "GroupList": ["string"],
        "Path": "string",
        "PermissionsBoundary": {
         "PermissionsBoundaryArn": "string",
         "PermissionsBoundaryType": "string"
        },
        "UserId": "string",
        "UserName": "string",
        "UserPolicyList": [{
         "PolicyName": "string"
        }]
       },
       "AwsKinesisStream": {
        "Arn": "string",
        "Name": "string",
        "RetentionPeriodHours": number,
        "ShardCount": number,
        "StreamEncryption": {
         "EncryptionType": "string",
         "KeyId": "string"
        }
```

```
      },
      "AwsKmsKey": {
       "AWSAccountId": "string",
       "CreationDate": "string",
       "Description": "string",
       "KeyId": "string",
       "KeyManager": "string",
       "KeyRotationStatus": boolean,
       "KeyState": "string",
       "Origin": "string"
      },
      "AwsLambdaFunction": {
       "Architectures": [
        "string"
       ],
       "Code": {
        "S3Bucket": "string",
        "S3Key": "string",
        "S3ObjectVersion": "string",
        "ZipFile": "string"
       },
       "CodeSha256": "string",
       "DeadLetterConfig": {
        "TargetArn": "string"
       },
       "Environment": {
        "Variables": {
         "Stage": "string"
        },
        "Error": {
         "ErrorCode": "string",
         "Message": "string"
        }
       },
       "FunctionName": "string",
       "Handler": "string",
       "KmsKeyArn": "string",
       "LastModified": "string",
       "Layers": {
        "Arn": "string",
        "CodeSize": number
       },
       "PackageType": "string",
       "RevisionId": "string",
```

```
        "Role": "string",
        "Runtime": "string",
        "Timeout": integer,
        "TracingConfig": {
         "Mode": "string"
        },
        "Version": "string",
        "VpcConfig": {
         "SecurityGroupIds": ["string"],
         "SubnetIds": ["string"]
        },
        "MasterArn": "string",
        "MemorySize": number
       },
       "AwsLambdaLayerVersion": {
        "CompatibleRuntimes": [
         "string"
        ],
        "CreatedDate": "string",
        "Version": number
       },
       "AwsMskCluster": {
        "ClusterInfo": {
         "ClientAuthentication": {
          "Sasl": {
           "Scram": {
            "Enabled": boolean
           },
           "Iam": {
            "Enabled": boolean
           }
          },
          "Tls": {
           "CertificateAuthorityArnList": [],
           "Enabled": boolean
          },
          "Unauthenticated": {
           "Enabled": boolean
          }
         },
         "ClusterName": "string",
         "CurrentVersion": "string",
         "EncryptionInfo": {
          "EncryptionAtRest": {
```

```
          "DataVolumeKMSKeyId": "string"
         },
         "EncryptionInTransit": {
          "ClientBroker": "string",
          "InCluster": boolean
         }
        },
        "EnhancedMonitoring": "string",
        "NumberOfBrokerNodes": integer
       }
      },
      "AwsNetworkFirewallFirewall": {
       "DeleteProtection": boolean,
       "Description": "string",
       "FirewallArn": "string",
       "FirewallId": "string",
       "FirewallName": "string",
       "FirewallPolicyArn": "string",
       "FirewallPolicyChangeProtection": boolean,
       "SubnetChangeProtection": boolean,
       "SubnetMappings": [{
        "SubnetId": "string"
       }],
       "VpcId": "string"
      },
      "AwsNetworkFirewallFirewallPolicy": {
       "Description": "string",
       "FirewallPolicy": {
        "StatefulRuleGroupReferences": [{
         "ResourceArn": "string"
        }],
        "StatelessCustomActions": [{
         "ActionDefinition": {
          "PublishMetricAction": {
           "Dimensions": [{
            "Value": "string"
           }]
          }
         },
         "ActionName": "string"
        }],
        "StatelessDefaultActions": ["string"],
        "StatelessFragmentDefaultActions": ["string"],
        "StatelessRuleGroupReferences": [{
```

```
          "Priority": number,
          "ResourceArn": "string"
         }]
        },
        "FirewallPolicyArn": "string",
        "FirewallPolicyId": "string",
        "FirewallPolicyName": "string"
       },
       "AwsNetworkFirewallRuleGroup": {
        "Capacity": number,
        "Description": "string",
        "RuleGroup": {
         "RulesSource": {
          "RulesSourceList": {
           "GeneratedRulesType": "string",
           "Targets": ["string"],
           "TargetTypes": ["string"]
          },
          "RulesString": "string",
          "StatefulRules": [{
           "Action": "string",
           "Header": {
            "Destination": "string",
            "DestinationPort": "string",
            "Direction": "string",
            "Protocol": "string",
            "Source": "string",
            "SourcePort": "string"
           },
           "RuleOptions": [{
            "Keyword": "string",
            "Settings": ["string"]
           }]
          }],
          "StatelessRulesAndCustomActions": {
           "CustomActions": [{
            "ActionDefinition": {
             "PublishMetricAction": {
              "Dimensions": [{
               "Value": "string"
              }]
             }
            },
            "ActionName": "string"
```

```
          }],
          "StatelessRules": [{
           "Priority": number,
           "RuleDefinition": {
            "Actions": ["string"],
            "MatchAttributes": {
             "DestinationPorts": [{
              "FromPort": number,
              "ToPort": number
             }],
             "Destinations": [{
              "AddressDefinition": "string"
             }],
             "Protocols": [number],
             "SourcePorts": [{
              "FromPort": number,
              "ToPort": number
             }],
             "Sources": [{
              "AddressDefinition": "string"
             }],
             "TcpFlags": [{
              "Flags": ["string"],
              "Masks": ["string"]
             }]
            }
           }
          }]
         }
        },
        "RuleVariables": {
         "IpSets": {
          "Definition": ["string"]
         },
         "PortSets": {
          "Definition": ["string"]
         }
        }
       },
       "RuleGroupArn": "string",
       "RuleGroupId": "string",
       "RuleGroupName": "string",
       "Type": "string"
      },
```

```
      "AwsOpenSearchServiceDomain": {
       "AccessPolicies": "string",
       "AdvancedSecurityOptions": {
        "Enabled": boolean,
        "InternalUserDatabaseEnabled": boolean,
        "MasterUserOptions": {
         "MasterUserArn": "string",
         "MasterUserName": "string",
         "MasterUserPassword": "string"
        }
       },
       "Arn": "string",
       "ClusterConfig": {
        "DedicatedMasterCount": number,
        "DedicatedMasterEnabled": boolean,
        "DedicatedMasterType": "string",
        "InstanceCount": number,
        "InstanceType": "string",
        "WarmCount": number,
        "WarmEnabled": boolean,
        "WarmType": "string",
        "ZoneAwarenessConfig": {
         "AvailabilityZoneCount": number
        },
        "ZoneAwarenessEnabled": boolean
       },
       "DomainEndpoint": "string",
       "DomainEndpointOptions": {
        "CustomEndpoint": "string",
        "CustomEndpointCertificateArn": "string",
        "CustomEndpointEnabled": boolean,
        "EnforceHTTPS": boolean,
        "TLSSecurityPolicy": "string"
       },
       "DomainEndpoints": {
        "string": "string"
       },
       "DomainName": "string",
       "EncryptionAtRestOptions": {
        "Enabled": boolean,
        "KmsKeyId": "string"
       },
       "EngineVersion": "string",
       "Id": "string",
```

```
        "LogPublishingOptions": {
         "AuditLogs": {
          "CloudWatchLogsLogGroupArn": "string",
          "Enabled": boolean
         },
         "IndexSlowLogs": {
          "CloudWatchLogsLogGroupArn": "string",
          "Enabled": boolean
         },
         "SearchSlowLogs": {
          "CloudWatchLogsLogGroupArn": "string",
          "Enabled": boolean
         }
        },
        "NodeToNodeEncryptionOptions": {
         "Enabled": boolean
        },
        "ServiceSoftwareOptions": {
         "AutomatedUpdateDate": "string",
         "Cancellable": boolean,
         "CurrentVersion": "string",
         "Description": "string",
         "NewVersion": "string",
         "OptionalDeployment": boolean,
         "UpdateAvailable": boolean,
         "UpdateStatus": "string"
        },
        "VpcOptions": {
         "SecurityGroupIds": ["string"],
         "SubnetIds": ["string"]
        }
       },
       "AwsRdsDbCluster": {
        "ActivityStreamStatus": "string",
        "AllocatedStorage": number,
        "AssociatedRoles": [{
         "RoleArn": "string",
         "Status": "string"
        }],
        "AutoMinorVersionUpgrade": boolean,
        "AvailabilityZones": ["string"],
        "BackupRetentionPeriod": integer,
        "ClusterCreateTime": "string",
        "CopyTagsToSnapshot": boolean,
```

```
        "CrossAccountClone": boolean,
        "CustomEndpoints": ["string"],
        "DatabaseName": "string",
        "DbClusterIdentifier": "string",
        "DbClusterMembers": [{
         "DbClusterParameterGroupStatus": "string",
         "DbInstanceIdentifier": "string",
         "IsClusterWriter": boolean,
         "PromotionTier": integer
        }],
        "DbClusterOptionGroupMemberships": [{
         "DbClusterOptionGroupName": "string",
         "Status": "string"
        }],
        "DbClusterParameterGroup": "string",
        "DbClusterResourceId": "string",
        "DbSubnetGroup": "string",
        "DeletionProtection": boolean,
        "DomainMemberships": [{
         "Domain": "string",
         "Fqdn": "string",
         "IamRoleName": "string",
         "Status": "string"
        }],
        "EnabledCloudwatchLogsExports": ["string"],
        "Endpoint": "string",
        "Engine": "string",
        "EngineMode": "string",
        "EngineVersion": "string",
        "HostedZoneId": "string",
        "HttpEndpointEnabled": boolean,
        "IamDatabaseAuthenticationEnabled": boolean,
        "KmsKeyId": "string",
        "MasterUsername": "string",
        "MultiAz": boolean,
        "Port": integer,
        "PreferredBackupWindow": "string",
        "PreferredMaintenanceWindow": "string",
        "ReaderEndpoint": "string",
        "ReadReplicaIdentifiers": ["string"],
        "Status": "string",
        "StorageEncrypted": boolean,
        "VpcSecurityGroups": [{
         "Status": "string",
```

```
        "VpcSecurityGroupId": "string"
       }]
      },
      "AwsRdsDbClusterSnapshot": {
       "AllocatedStorage": integer,
       "AvailabilityZones": ["string"],
       "ClusterCreateTime": "string",
       "DbClusterIdentifier": "string",
       "DbClusterSnapshotAttributes": [{
        "AttributeName": "string",
        "AttributeValues": ["string"]
       }],
       "DbClusterSnapshotIdentifier": "string",
       "Engine": "string",
       "EngineVersion": "string",
       "IamDatabaseAuthenticationEnabled": boolean,
       "KmsKeyId": "string",
       "LicenseModel": "string",
       "MasterUsername": "string",
       "PercentProgress": integer,
       "Port": integer,
       "SnapshotCreateTime": "string",
       "SnapshotType": "string",
       "Status": "string",
       "StorageEncrypted": boolean,
       "VpcId": "string"
      },
      "AwsRdsDbInstance": {
       "AllocatedStorage": number,
       "AssociatedRoles": [{
        "RoleArn": "string",
        "FeatureName": "string",
        "Status": "string"
       }],
       "AutoMinorVersionUpgrade": boolean,
       "AvailabilityZone": "string",
       "BackupRetentionPeriod": number,
       "CACertificateIdentifier": "string",
       "CharacterSetName": "string",
       "CopyTagsToSnapshot": boolean,
       "DBClusterIdentifier": "string",
       "DBInstanceClass": "string",
       "DBInstanceIdentifier": "string",
       "DbInstancePort": number,
```

```
        "DbInstanceStatus": "string",
        "DbiResourceId": "string",
        "DBName": "string",
        "DbParameterGroups": [{
         "DbParameterGroupName": "string",
         "ParameterApplyStatus": "string"
        }],
        "DbSecurityGroups": ["string"],
        "DbSubnetGroup": {
         "DbSubnetGroupArn": "string",
         "DbSubnetGroupDescription": "string",
         "DbSubnetGroupName": "string",
         "SubnetGroupStatus": "string",
         "Subnets": [{
          "SubnetAvailabilityZone": {
           "Name": "string"
          },
          "SubnetIdentifier": "string",
          "SubnetStatus": "string"
         }],
         "VpcId": "string"
        },
        "DeletionProtection": boolean,
        "Endpoint": {
         "Address": "string",
         "Port": number,
         "HostedZoneId": "string"
        },
        "DomainMemberships": [{
         "Domain": "string",
         "Fqdn": "string",
         "IamRoleName": "string",
         "Status": "string"
        }],
        "EnabledCloudwatchLogsExports": ["string"],
        "Engine": "string",
        "EngineVersion": "string",
        "EnhancedMonitoringResourceArn": "string",
        "IAMDatabaseAuthenticationEnabled": boolean,
        "InstanceCreateTime": "string",
        "Iops": number,
        "KmsKeyId": "string",
        "LatestRestorableTime": "string",
        "LicenseModel": "string",
```

```
        "ListenerEndpoint": {
         "Address": "string",
         "HostedZoneId": "string",
         "Port": number
        },
        "MasterUsername": "admin",
        "MaxAllocatedStorage": number,
        "MonitoringInterval": number,
        "MonitoringRoleArn": "string",
        "MultiAz": boolean,
        "OptionGroupMemberships": [{
         "OptionGroupName": "string",
         "Status": "string"
        }],
        "PendingModifiedValues": {
         "AllocatedStorage": number,
         "BackupRetentionPeriod": number,
         "CaCertificateIdentifier": "string",
         "DbInstanceClass": "string",
         "DbInstanceIdentifier": "string",
         "DbSubnetGroupName": "string",
         "EngineVersion": "string",
         "Iops": number,
         "LicenseModel": "string",
         "MasterUserPassword": "string",
         "MultiAZ": boolean,
         "PendingCloudWatchLogsExports": {
          "LogTypesToDisable": ["string"],
          "LogTypesToEnable": ["string"]
         },
         "Port": number,
         "ProcessorFeatures": [{
          "Name": "string",
          "Value": "string"
         }],
         "StorageType": "string"
        },
        "PerformanceInsightsEnabled": boolean,
        "PerformanceInsightsKmsKeyId": "string",
        "PerformanceInsightsRetentionPeriod": number,
        "PreferredBackupWindow": "string",
        "PreferredMaintenanceWindow": "string",
        "ProcessorFeatures": [{
         "Name": "string",
```

```
        "Value": "string"
       }],
       "PromotionTier": number,
       "PubliclyAccessible": boolean,
       "ReadReplicaDBClusterIdentifiers": ["string"],
       "ReadReplicaDBInstanceIdentifiers": ["string"],
       "ReadReplicaSourceDBInstanceIdentifier": "string",
       "SecondaryAvailabilityZone": "string",
       "StatusInfos": [{
        "Message": "string",
        "Normal": boolean,
        "Status": "string",
        "StatusType": "string"
       }],
       "StorageEncrypted": boolean,
       "TdeCredentialArn": "string",
       "Timezone": "string",
       "VpcSecurityGroups": [{
        "VpcSecurityGroupId": "string",
        "Status": "string"
       }]
      },
      "AwsRdsDbSecurityGroup": {
       "DbSecurityGroupArn": "string",
       "DbSecurityGroupDescription": "string",
       "DbSecurityGroupName": "string",
       "Ec2SecurityGroups": [{
        "Ec2SecurityGroupuId": "string",
        "Ec2SecurityGroupName": "string",
        "Ec2SecurityGroupOwnerId": "string",
        "Status": "string"
       }],
       "IpRanges": [{
        "CidrIp": "string",
        "Status": "string"
       }],
       "OwnerId": "string",
       "VpcId": "string"
      },
      "AwsRdsDbSnapshot": {
       "AllocatedStorage": integer,
       "AvailabilityZone": "string",
       "DbInstanceIdentifier": "string",
       "DbiResourceId": "string",
```

```
        "DbSnapshotIdentifier": "string",
        "Encrypted": boolean,
        "Engine": "string",
        "EngineVersion": "string",
        "IamDatabaseAuthenticationEnabled": boolean,
        "InstanceCreateTime": "string",
        "Iops": number,
        "KmsKeyId": "string",
        "LicenseModel": "string",
        "MasterUsername": "string",
        "OptionGroupName": "string",
        "PercentProgress": integer,
        "Port": integer,
        "ProcessorFeatures": [],
        "SnapshotCreateTime": "string",
        "SnapshotType": "string",
        "SourceDbSnapshotIdentifier": "string",
        "SourceRegion": "string",
        "Status": "string",
        "StorageType": "string",
        "TdeCredentialArn": "string",
        "Timezone": "string",
        "VpcId": "string"
      },
      "AwsRdsEventSubscription": {
        "CustomerAwsId": "string",
        "CustSubscriptionId": "string",
        "Enabled": boolean,
        "EventCategoriesList": ["string"],
        "EventSubscriptionArn": "string",
        "SnsTopicArn": "string",
        "SourceIdsList": ["string"],
        "SourceType": "string",
        "Status": "string",
        "SubscriptionCreationTime": "string"
      },
      "AwsRedshiftCluster": {
        "AllowVersionUpgrade": boolean,
        "AutomatedSnapshotRetentionPeriod": number,
        "AvailabilityZone": "string",
        "ClusterAvailabilityStatus": "string",
        "ClusterCreateTime": "string",
        "ClusterIdentifier": "string",
        "ClusterNodes": [{
```

```
        "NodeRole": "string",
        "PrivateIPAddress": "string",
        "PublicIPAddress": "string"
       }],
       "ClusterParameterGroups": [{
        "ClusterParameterStatusList": [{
         "ParameterApplyErrorDescription": "string",
         "ParameterApplyStatus": "string",
         "ParameterName": "string"
        }],
        "ParameterApplyStatus": "string",
        "ParameterGroupName": "string"
       }],
       "ClusterPublicKey": "string",
       "ClusterRevisionNumber": "string",
       "ClusterSecurityGroups": [{
        "ClusterSecurityGroupName": "string",
        "Status": "string"
       }],
       "ClusterSnapshotCopyStatus": {
        "DestinationRegion": "string",
        "ManualSnapshotRetentionPeriod": number,
        "RetentionPeriod": number,
        "SnapshotCopyGrantName": "string"
       },
       "ClusterStatus": "string",
       "ClusterSubnetGroupName": "string",
       "ClusterVersion": "string",
       "DBName": "string",
       "DeferredMaintenanceWindows": [{
        "DeferMaintenanceEndTime": "string",
        "DeferMaintenanceIdentifier": "string",
        "DeferMaintenanceStartTime": "string"
       }],
       "ElasticIpStatus": {
        "ElasticIp": "string",
        "Status": "string"
       },
       "ElasticResizeNumberOfNodeOptions": "string",
       "Encrypted": boolean,
       "Endpoint": {
        "Address": "string",
        "Port": number
       },
```

```
        "EnhancedVpcRouting": boolean,
        "ExpectedNextSnapshotScheduleTime": "string",
        "ExpectedNextSnapshotScheduleTimeStatus": "string",
        "HsmStatus": {
         "HsmClientCertificateIdentifier": "string",
         "HsmConfigurationIdentifier": "string",
         "Status": "string"
        },
        "IamRoles": [{
         "ApplyStatus": "string",
         "IamRoleArn": "string"
        }],
        "KmsKeyId": "string",
        "LoggingStatus":{
                    "BucketName": "string",
                    "LastFailureMessage": "string",
                    "LastFailureTime": "string",
                    "LastSuccessfulDeliveryTime": "string",
                    "LoggingEnabled": boolean,
                    "S3KeyPrefix": "string"
                },
        "MaintenanceTrackName": "string",
        "ManualSnapshotRetentionPeriod": number,
        "MasterUsername": "string",
        "NextMaintenanceWindowStartTime": "string",
        "NodeType": "string",
        "NumberOfNodes": number,
        "PendingActions": ["string"],
        "PendingModifiedValues": {
         "AutomatedSnapshotRetentionPeriod": number,
         "ClusterIdentifier": "string",
         "ClusterType": "string",
         "ClusterVersion": "string",
         "EncryptionType": "string",
         "EnhancedVpcRouting": boolean,
         "MaintenanceTrackName": "string",
         "MasterUserPassword": "string",
         "NodeType": "string",
         "NumberOfNodes": number,
         "PubliclyAccessible": "string"
        },
        "PreferredMaintenanceWindow": "string",
        "PubliclyAccessible": boolean,
        "ResizeInfo": {
```

```
        "AllowCancelResize": boolean,
        "ResizeType": "string"
       },
       "RestoreStatus": {
        "CurrentRestoreRateInMegaBytesPerSecond": number,
        "ElapsedTimeInSeconds": number,
        "EstimatedTimeToCompletionInSeconds": number,
        "ProgressInMegaBytes": number,
        "SnapshotSizeInMegaBytes": number,
        "Status": "string"
       },
       "SnapshotScheduleIdentifier": "string",
       "SnapshotScheduleState": "string",
       "VpcId": "string",
       "VpcSecurityGroups": [{
        "Status": "string",
        "VpcSecurityGroupId": "string"
       }]
      },
      "AwsRoute53HostedZone": {
       "HostedZone": {
        "Id": "string",
        "Name": "string",
        "Config": {
         "Comment": "string"
        }
       },
       "NameServers": ["string"],
       "QueryLoggingConfig": {
        "CloudWatchLogsLogGroupArn": {
         "CloudWatchLogsLogGroupArn": "string",
         "Id": "string",
         "HostedZoneId": "string"
        }
       },
       "Vpcs": [
        {
         "Id": "string",
         "Region": "string"
        }
       ]
      },
      "AwsS3AccessPoint": {
       "AccessPointArn": "string",
```

```
        "Alias": "string",
        "Bucket": "string",
        "BucketAccountId": "string",
        "Name": "string",
        "NetworkOrigin": "string",
        "PublicAccessBlockConfiguration": {
         "BlockPublicAcls": boolean,
         "BlockPublicPolicy": boolean,
         "IgnorePublicAcls": boolean,
         "RestrictPublicBuckets": boolean
        },
        "VpcConfiguration": {
         "VpcId": "string"
        }
       },
       "AwsS3AccountPublicAccessBlock": {
        "BlockPublicAcls": boolean,
        "BlockPublicPolicy": boolean,
        "IgnorePublicAcls": boolean,
        "RestrictPublicBuckets": boolean
       },
       "AwsS3Bucket": {
        "AccessControlList": "string",
        "BucketLifecycleConfiguration": {
         "Rules": [{
          "AbortIncompleteMultipartUpload": {
           "DaysAfterInitiation": number
          },
          "ExpirationDate": "string",
          "ExpirationInDays": number,
          "ExpiredObjectDeleteMarker": boolean,
          "Filter": {
           "Predicate": {
            "Operands": [{
              "Prefix": "string",
              "Type": "string"
             },
             {
              "Tag": {
               "Key": "string",
               "Value": "string"
              },
              "Type": "string"
             }
```

```
          ],
          "Type": "string"
         }
        },
        "Id": "string",
        "NoncurrentVersionExpirationInDays": number,
        "NoncurrentVersionTransitions": [{
         "Days": number,
         "StorageClass": "string"
        }],
        "Prefix": "string",
        "Status": "string",
        "Transitions": [{
         "Date": "string",
         "Days": number,
         "StorageClass": "string"
        }]
       }]
      },
      "BucketLoggingConfiguration": {
       "DestinationBucketName": "string",
       "LogFilePrefix": "string"
      },
      "BucketName": "string",
      "BucketNotificationConfiguration": {
       "Configurations": [{
        "Destination": "string",
        "Events": ["string"],
        "Filter": {
         "S3KeyFilter": {
          "FilterRules": [{
           "Name": "string",
           "Value": "string"
          }]
         }
        },
        "Type": "string"
       }]
      },
      "BucketVersioningConfiguration": {
       "IsMfaDeleteEnabled": boolean,
       "Status": "string"
      },
      "BucketWebsiteConfiguration": {
```

```
        "ErrorDocument": "string",
        "IndexDocumentSuffix": "string",
        "RedirectAllRequestsTo": {
         "HostName": "string",
         "Protocol": "string"
        },
        "RoutingRules": [{
         "Condition": {
          "HttpErrorCodeReturnedEquals": "string",
          "KeyPrefixEquals": "string"
         },
         "Redirect": {
          "HostName": "string",
          "HttpRedirectCode": "string",
          "Protocol": "string",
          "ReplaceKeyPrefixWith": "string",
          "ReplaceKeyWith": "string"
         }
        }]
       },
       "CreatedAt": "string",
       "ObjectLockConfiguration": {
        "ObjectLockEnabled": "string",
        "Rule": {
         "DefaultRetention": {
          "Days": integer,
          "Mode": "string",
          "Years": integer
         }
        }
       },
       "OwnerAccountId": "string",
       "OwnerId": "string",
       "OwnerName": "string",
       "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": boolean,
        "BlockPublicPolicy": boolean,
        "IgnorePublicAcls": boolean,
        "RestrictPublicBuckets": boolean
       },
       "ServerSideEncryptionConfiguration": {
        "Rules": [{
         "ApplyServerSideEncryptionByDefault": {
          "KMSMasterKeyID": "string",
```

```
          "SSEAlgorithm": "string"
         }
        }]
       }
      },
      "AwsS3Object": {
       "ContentType": "string",
       "ETag": "string",
       "LastModified": "string",
       "ServerSideEncryption": "string",
       "SSEKMSKeyId": "string",
       "VersionId": "string"
      },
      "AwsSagemakerNotebookInstance": {
       "DirectInternetAccess": "string",
       "InstanceMetadataServiceConfiguration": {
        "MinimumInstanceMetadataServiceVersion": "string"
       },
       "InstanceType": "string",
       "LastModifiedTime": "string",
       "NetworkInterfaceId": "string",
       "NotebookInstanceArn": "string",
       "NotebookInstanceName": "string",
       "NotebookInstanceStatus": "string",
       "PlatformIdentifier": "string",
       "RoleArn": "string",
       "RootAccess": "string",
       "SecurityGroups": ["string"],
       "SubnetId": "string",
       "Url": "string",
       "VolumeSizeInGB": number
      },
      "AwsSecretsManagerSecret": {
       "Deleted": boolean,
       "Description": "string",
       "KmsKeyId": "string",
       "Name": "string",
       "RotationEnabled": boolean,
       "RotationLambdaArn": "string",
       "RotationOccurredWithinFrequency": boolean,
       "RotationRules": {
        "AutomaticallyAfterDays": integer
       }
      },
```

```
      "AwsSnsTopic": {
       "ApplicationSuccessFeedbackRoleArn": "string",
       "FirehoseFailureFeedbackRoleArn": "string",
       "FirehoseSuccessFeedbackRoleArn": "string",
       "HttpFailureFeedbackRoleArn": "string",
       "HttpSuccessFeedbackRoleArn": "string",
       "KmsMasterKeyId": "string",
       "Owner": "string",
       "SqsFailureFeedbackRoleArn": "string",
       "SqsSuccessFeedbackRoleArn": "string",
       "Subscription": {
        "Endpoint": "string",
        "Protocol": "string"
       },
       "TopicName": "string"
      },
      "AwsSqsQueue": {
       "DeadLetterTargetArn": "string",
       "KmsDataKeyReusePeriodSeconds": number,
       "KmsMasterKeyId": "string",
       "QueueName": "string"
      },
      "AwsSsmPatchCompliance": {
       "Patch": {
        "ComplianceSummary": {
         "ComplianceType": "string",
         "CompliantCriticalCount": integer,
         "CompliantHighCount": integer,
         "CompliantInformationalCount": integer,
         "CompliantLowCount": integer,
         "CompliantMediumCount": integer,
         "CompliantUnspecifiedCount": integer,
         "ExecutionType": "string",
         "NonCompliantCriticalCount": integer,
         "NonCompliantHighCount": integer,
         "NonCompliantInformationalCount": integer,
         "NonCompliantLowCount": integer,
         "NonCompliantMediumCount": integer,
         "NonCompliantUnspecifiedCount": integer,
         "OverallSeverity": "string",
         "PatchBaselineId": "string",
         "PatchGroup": "string",
         "Status": "string"
        }
```

```
        }
       },
       "AwsStepFunctionStateMachine": {
        "StateMachineArn": "string",
        "Name": "string",
        "Status": "string",
        "RoleArn": "string",
        "Type": "string",
        "LoggingConfiguration": {
         "Level": "string",
         "IncludeExecutionData": boolean
        },
        "TracingConfiguration": {
         "Enabled": boolean
        }
       },
       "AwsWafRateBasedRule": {
        "MatchPredicates": [{
         "DataId": "string",
         "Negated": boolean,
         "Type": "string"
        }],
        "MetricName": "string",
        "Name": "string",
        "RateKey": "string",
        "RateLimit": number,
        "RuleId": "string"
       },
       "AwsWafRegionalRateBasedRule": {
        "MatchPredicates": [{
         "DataId": "string",
         "Negated": boolean,
         "Type": "string"
        }],
        "MetricName": "string",
        "Name": "string",
        "RateKey": "string",
        "RateLimit": number,
        "RuleId": "string"
       },
       "AwsWafRegionalRule": {
        "MetricName": "string",
        "Name": "string",
        "RuleId": "string",
```

```
        "PredicateList": [{
            "DataId": "string",
            "Negated": boolean,
            "Type": "string"
        }]
      },
      "AwsWafRegionalRuleGroup": {
        "MetricName": "string",
        "Name": "string",
        "RuleGroupId": "string",
        "Rules": [{
          "Action": {
            "Type": "string"
          },
          "Priority": number,
          "RuleId": "string",
          "Type": "string"
        }]
      },
      "AwsWafRegionalWebAcl": {
        "DefaultAction": "string",
        "MetricName" : "string",
        "Name": "string",
        "RulesList" : [{
          "Action": {
            "Type": "string"
          },
          "Priority": number,
          "RuleId": "string",
          "Type": "string",
          "ExcludedRules": [{
            "ExclusionType": "string",
            "RuleId": "string"
          }],
          "OverrideAction": {
            "Type": "string"
          }
        }],
        "WebAclId": "string"
      },
      "AwsWafRule": {
        "MetricName": "string",
        "Name": "string",
        "PredicateList": [{
```

```
        "DataId": "string",
        "Negated": boolean,
        "Type": "string"
       }],
       "RuleId": "string"
      },
      "AwsWafRuleGroup": {
       "MetricName": "string",
       "Name": "string",
       "RuleGroupId": "string",
       "Rules": [{
        "Action": {
         "Type": "string"
        },
        "Priority": number,
        "RuleId": "string",
        "Type": "string"
       }]
      },
      "AwsWafv2RuleGroup": {
       "Arn": "string",
       "Capacity": number,
       "Description": "string",
       "Id": "string",
       "Name": "string",
       "Rules": [{
        "Action": {
        "Allow": {
         "CustomRequestHandling": {
          "InsertHeaders": [
           {
           "Name": "string",
           "Value": "string"
           },
           {
           "Name": "string",
           "Value": "string"
           }
          ]
         }
        }
        },
        "Name": "string",
        "Priority": number,
```

```
        "VisibilityConfig": {
         "CloudWatchMetricsEnabled": boolean,
         "MetricName": "string",
         "SampledRequestsEnabled": boolean
        }
       }],
       "VisibilityConfig": {
        "CloudWatchMetricsEnabled": boolean,
        "MetricName": "string",
        "SampledRequestsEnabled": boolean
       }
      },
      "AwsWafWebAcl": {
       "DefaultAction": "string",
       "Name": "string",
       "Rules": [{
        "Action": {
         "Type": "string"
        },
        "ExcludedRules": [{
         "RuleId": "string"
        }],
        "OverrideAction": {
         "Type": "string"
        },
        "Priority": number,
        "RuleId": "string",
        "Type": "string"
       }],
       "WebAclId": "string"
      },
      "AwsWafv2WebAcl": {
       "Arn": "string",
       "Capacity": number,
       "CaptchaConfig": {
        "ImmunityTimeProperty": {
         "ImmunityTime": number
        }
       },
       "DefaultAction": {
        "Block": {}
       },
       "Description": "string",
       "ManagedbyFirewallManager": boolean,
```

```
        "Name": "string",
        "Rules": [{
         "Action": {
          "RuleAction": {
           "Block": {}
          }
         },
         "Name": "string",
         "Priority": number,
         "VisibilityConfig": {
          "SampledRequestsEnabled": boolean,
          "CloudWatchMetricsEnabled": boolean,
          "MetricName": "string"
         }
        }],
        "VisibilityConfig": {
         "SampledRequestsEnabled": boolean,
         "CloudWatchMetricsEnabled": boolean,
         "MetricName": "string"
        }
       },
       "AwsXrayEncryptionConfig": {
        "KeyId": "string",
        "Status": "string",
        "Type": "string"
       },
       "CodeRepository": {
        "CodeSecurityIntegrationArn": "string",
        "ProjectName": "string",
        "ProviderType": "string"
       },
       "Container": {
        "ContainerRuntime": "string",
        "ImageId": "string",
        "ImageName": "string",
        "LaunchedAt": "string",
        "Name": "string",
        "Privileged": boolean,
        "VolumeMounts": [{
         "Name": "string",
         "MountPath": "string"
        }]
       },
       "Other": {
```

```
        "string": "string"
       },
       "Id": "string",
       "Partition": "string",
       "Region": "string",
       "ResourceRole": "string",
       "Tags": {
        "string": "string"
       },
       "Type": "string"
      }],
      "SchemaVersion": "string",
      "Severity": {
       "Label": "string",
       "Normalized": number,
       "Original": "string"
      },
      "Sample": boolean,
      "SourceUrl": "string",
      "Threats": [{
       "FilePaths": [{
        "FileName": "string",
        "FilePath": "string",
        "Hash": "string",
        "ResourceId": "string"
       }],
       "ItemCount": number,
       "Name": "string",
       "Severity": "string"
      }],
      "ThreatIntelIndicators": [{
       "Category": "string",
       "LastObservedAt": "string",
       "Source": "string",
       "SourceUrl": "string",
       "Type": "string",
       "Value": "string"
      }],
      "Title": "string",
      "Types": ["string"],
      "UpdatedAt": "string",
      "UserDefinedFields": {
       "string": "string"
      },
```

```
      "VerificationState": "string",
      "Vulnerabilities": [{
       "CodeVulnerabilities": [{
        "Cwes": [
         "string",
         "string"
        ],
        "FilePath": {
         "EndLine": integer,
         "FileName": "string",
         "FilePath": "string",
         "StartLine": integer
        },
        "SourceArn":"string"
       }],
       "Cvss": [{
        "Adjustments": [{
         "Metric": "string",
         "Reason": "string"
        }],
        "BaseScore": number,
        "BaseVector": "string",
        "Source": "string",
        "Version": "string"
       }],
       "EpssScore": number,
       "ExploitAvailable": "string",
       "FixAvailable": "string",
       "Id": "string",
       "LastKnownExploitAt": "string",
       "ReferenceUrls": ["string"],
       "RelatedVulnerabilities": ["string"],
       "Vendor": {
        "Name": "string",
        "Url": "string",
        "VendorCreatedAt": "string",
        "VendorSeverity": "string",
        "VendorUpdatedAt": "string"
       },
       "VulnerablePackages": [{
        "Architecture": "string",
        "Epoch": "string",
        "FilePath": "string",
        "FixedInVersion": "string",
```

```
        "Name": "string",
        "PackageManager": "string",
        "Release": "string",
        "Remediation": "string",
        "SourceLayerArn": "string",
        "SourceLayerHash": "string",
        "Version": "string"
       }]
      }],
      "Workflow": {
       "Status": "string"
      },
      "WorkflowState": "string"
     }
]
```

Impact of consolidation on ASFF fields and values

AWS Security Hub CSPM offers two types of consolidation for controls:

- **Consolidated controls view** – With this type of consolidation, each control has a single identifier across all standards. In addition, on the Security Hub CSPM console, the **Controls** page displays all controls across all standards.
- **Consolidated control findings** – With this type of consolidation, Security Hub CSPM produces a single finding for a control, even if the control applies to multiple enabled standards. This can reduce finding noise.

You can't enable or disable consolidated controls view. Consolidated control findings is enabled by default if you enable Security Hub CSPM on or after February 23, 2023. Otherwise, it's disabled by default. However, for organizations, consolidated control findings is enabled for Security Hub CSPM member accounts only if it's enabled for the administrator account. To learn more about consolidated control findings, see Generating and updating control findings.

Both types of consolidation affect fields and values for control findings in the AWS Security Finding Format (ASFF).

Topics

- Consolidated controls view – ASFF changes
- Consolidated control findings – ASFF changes

- Generator IDs before and after enabling consolidated control findings
- How consolidation impacts control IDs and titles
- Updating workflows for consolidation

Consolidated controls view – ASFF changes

The consolidated controls view feature introduced the following changes to fields and values for control findings in the ASFF. If your workflows don't rely on values for these ASFF fields, no action is required. If you have workflows that rely on specific values for these fields, update your workflows to use the current values.

ASFF field	Sample value before consolidated controls view	Sample value after consolidated controls view, and a description of the change
Compliance.SecurityControlId	Not applicable (new field)	EC2.2 Introduces a single control ID across standards. `ProductFields.RuleId` still provides the standard-based control ID for CIS v1.2.0 controls. `ProductFields.ControlId` still provides the standard-based control ID for controls in other standards.
Compliance.AssociatedStandards	Not applicable (new field)	[{"StandardsId": "standards/aws-foundational-security-best-practices/v/1.0.0"}] Shows which standards a control is enabled in.
ProductFields.ArchivalReasons:0/Description	Not applicable (new field)	"The finding is in an ARCHIVED state because consolidated control findings

ASFF field	Sample value before consolidated controls view	Sample value after consolidated controls view, and a description of the change
		has been turned on or off. This causes findings in the previous state to be archived when new findings are being generated." Describes why Security Hub CSPM has archived existing findings.
ProductFields.ArchivalReasons:0/ReasonCode	Not applicable (new field)	"CONSOLIDATED_CONTROL_FINDINGS_UPDATE" Provides the reason why Security Hub CSPM has archived existing findings.
ProductFields.RecommendationUrl	https://docs.aws.amazon.com/console/securityhub/PCI.EC2.2/remediation	https://docs.aws.amazon.com/console/securityhub/EC2.2/remediation This field no longer references a standard.
Remediation.Recommendation.Text	"For directions on how to fix this issue, consult the AWS Security Hub CSPM PCI DSS documentation."	"For directions on how to correct this issue, consult the AWS Security Hub CSPM controls documentation." This field no longer references a standard.

ASFF field	Sample value before consolidated controls view	Sample value after consolidated controls view, and a description of the change
Remediation.Recommendation.Url	https://docs.aws.amazon.com/console/securityhub/PCI.EC2.2/remediation	https://docs.aws.amazon.com/console/securityhub/EC2.2/remediation This field no longer references a standard.

Consolidated control findings – ASFF changes

If you enable consolidated control findings, you might be affected by the following changes to fields and values for control findings in the ASFF. These changes are in addition to the changes introduced by the consolidated controls view feature. If your workflows don't rely on values for these ASFF fields, no action is required. If you have workflows that rely on specific values for these fields, update your workflows to use the current values.

> **Tip**
>
> If you use the Automated Security Response on AWS v2.0.0 solution, note that it supports consolidated control findings. This means that you can maintain your current workflows if you enable consolidated control findings.

ASFF field	Example value before enabling consolidated control findings	Example value after enabling consolidated control findings, and a description of the change
GeneratorId	aws-foundational-security-best-practices/v/1.0.0/Config.1	security-control/Config.1 This field no longer references a standard.
Title	PCI.Config.1 AWS Config should be enabled	AWS Config should be enabled

ASFF field	Example value before enabling consolidated control findings	Example value after enabling consolidated control findings, and a description of the change
		This field no longer references standard-specific information.
Id	arn:aws:securityhub:eu-central-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.IAM.5/finding/ab6d6a26-a156-48f0-9403-115983e5a956	arn:aws:securityhub:eu-central-1:123456789012:security-control/iam.9/finding/ab6d6a26-a156-48f0-9403-115983e5a956 This field no longer references a standard.
ProductFields.ControlId	PCI.EC2.2	Removed. See `Compliance.SecurityControlId` instead. This field is removed in favor of a single, standard-agnostic control ID.
ProductFields.RuleId	1.3	Removed. See `Compliance.SecurityControlId` instead. This field is removed in favor of a single, standard-agnostic control ID.
Description	This PCI DSS control checks whether AWS Config is enabled in the current account and region.	This AWS control checks whether AWS Config is enabled in the current account and region. This field no longer references a standard.

ASFF field	Example value before enabling consolidated control findings	Example value after enabling consolidated control findings, and a description of the change

Severity	"Severity": {
	"Product": 90,
	"Label": "CRITICAL",
	"Normalized": 90,
	"Original": "CRITICAL"
	}
		"Severity": {
		"Label": "CRITICAL",
		"Normalized": 90,
		"Original": "CRITICAL"
		}
		Security Hub CSPM no longer uses the Product field to describe the severity of a finding.

Types	["Software and Configuration Checks/Industry and Regulatory Standards/PCI-DSS"]	["Software and Configuration Checks/Industry and Regulatory Standards"] This field no longer references a standard.
Compliance.Related Requirements	["PCI DSS 10.5.2", "PCI DSS 11.5", "CIS AWS Foundations 2.5"]	["PCI DSS v3.2.1/10.5.2", "PCI DSS v3.2.1/11.5", "CIS AWS Foundations Benchmark v1.2.0/2.5"] This field shows related requirements in all enabled standards.

ASFF field	Example value before enabling consolidated control findings	Example value after enabling consolidated control findings, and a description of the change
CreatedAt	2022-05-05T08:18:13.138Z	2022-09-25T08:18:13.138Z Format remains the same, but value resets when you enable consolidated control findings.
FirstObservedAt	2022-05-07T08:18:13.138Z	2022-09-28T08:18:13.138Z Format remains the same, but value resets when you enable consolidated control findings.
ProductFields.RecommendationUrl	https://docs.aws.amazon.com/console/securityhub/EC2.2/remediation	Removed. See `Remediation.Recommendation.Url` instead.
ProductFields.StandardsArn	arn:aws:securityhub:::standards/aws-foundational-security-best-practices/v/1.0.0	Removed. See `Compliance.AssociatedStandards` instead.
ProductFields.StandardsControlArn	arn:aws:securityhub:us-east-1:123456789012:control/aws-foundational-security-best-practices/v/1.0.0/Config.1	Removed. Security Hub CSPM generates one finding for a security check across standards.
ProductFields.StandardsGuideArn	arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0	Removed. See `Compliance.AssociatedStandards` instead.
ProductFields.StandardsGuideSubscriptionArn	arn:aws:securityhub:us-east-2:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0	Removed. Security Hub CSPM generates one finding for a security check across standards.

ASFF field	Example value before enabling consolidated control findings	Example value after enabling consolidated control findings, and a description of the change
ProductFields.StandardsSubscriptionArn	arn:aws:securityhub:us-east-1:123456789012:subscription/aws-foundational-security-best-practices/v/1.0.0	Removed. Security Hub CSPM generates one finding for a security check across standards.
ProductFields.aws/securityhub/FindingId	arn:aws:securityhub:us-east-1::product/aws/securityhub/arn:aws:securityhub:us-east-1:123456789012:subscription/aws-foundational-security-best-practices/v/1.0.0/Config.1/finding/751c2173-7372-4e12-8656-a5210dfb1d67	arn:aws:securityhub:us-east-1::product/aws/securityhub/arn:aws:securityhub:us-east-1:123456789012:security-control/Config.1/finding/751c2173-7372-4e12-8656-a5210dfb1d67 This field no longer references a standard.

Values for customer-provided ASFF fields after turning on consolidated control findings

If you enable consolidated control findings, Security Hub CSPM generates one finding across standards and archives the original findings (separate findings for each standard).

Updates that you made to the original findings by using the Security Hub CSPM console or the BatchUpdateFindings operation won't be preserved in the new findings. If necessary, you can recover this data by referring to the archived findings. To review archived findings, you can use the **Findings** page on the Security Hub CSPM console and set the **Record state** filter to **ARCHIVED**. Alternatively, you can use the GetFindings operation of the Security Hub CSPM API.

Customer-provided ASFF field	Description of change after enabling consolidated control findings
Confidence	Resets to empty state.
Criticality	Resets to empty state.

Customer-provided ASFF field	Description of change after enabling consolidated control findings
Note	Resets to empty state.
RelatedFindings	Resets to empty state.
Severity	Default severity of the finding (matches the severity of the control).
Types	Resets to standard-agnostic value.
UserDefinedFields	Resets to empty state.
VerificationState	Resets to empty state.
Workflow	New failed findings have a default value of `NEW`. New passed findings have a default value of `RESOLVED`.

Generator IDs before and after enabling consolidated control findings

The following table lists changes to generator ID values for controls when you enable consolidated control findings. These changes apply to controls that Security Hub CSPM supported as of February 15, 2023.

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.1	security-control/CloudWatch.1
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.10	security-control/IAM.16
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.11	security-control/IAM.17

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.12	security-control/IAM.4
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.13	security-control/IAM.9
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.14	security-control/IAM.6
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.16	security-control/IAM.2
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.2	security-control/IAM.5
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.20	security-control/IAM.18
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.22	security-control/IAM.1
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.3	security-control/IAM.8
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.4	security-control/IAM.3
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.5	security-control/IAM.11
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.6	security-control/IAM.12
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.7	security-control/IAM.13

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.8	security-control/IAM.14
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.9	security-control/IAM.15
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/2.1	security-control/CloudTrail.1
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/2.2	security-control/CloudTrail.4
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/2.3	security-control/CloudTrail.6
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/2.4	security-control/CloudTrail.5
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/2.5	security-control/Config.1
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/2.6	security-control/CloudTrail.7
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/2.7	security-control/CloudTrail.2
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/2.8	security-control/KMS.4
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/2.9	security-control/EC2.6
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.1	security-control/CloudWatch.2

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.2	security-control/CloudWatch.3
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.3	security-control/CloudWatch.1
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.4	security-control/CloudWatch.4
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.5	security-control/CloudWatch.5
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.6	security-control/CloudWatch.6
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.7	security-control/CloudWatch.7
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.8	security-control/CloudWatch.8
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.9	security-control/CloudWatch.9
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.10	security-control/CloudWatch.10
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.11	security-control/CloudWatch.11
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.12	security-control/CloudWatch.12
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.13	security-control/CloudWatch.13

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.14	security-control/CloudWatch.14
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/4.1	security-control/EC2.13
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/4.2	security-control/EC2.14
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/4.3	security-control/EC2.2
cis-aws-foundations-benchmark/v/1.4.0/1.10	security-control/IAM.5
cis-aws-foundations-benchmark/v/1.4.0/1.14	security-control/IAM.3
cis-aws-foundations-benchmark/v/1.4.0/1.16	security-control/IAM.1
cis-aws-foundations-benchmark/v/1.4.0/1.17	security-control/IAM.18
cis-aws-foundations-benchmark/v/1.4.0/1.4	security-control/IAM.4
cis-aws-foundations-benchmark/v/1.4.0/1.5	security-control/IAM.9
cis-aws-foundations-benchmark/v/1.4.0/1.6	security-control/IAM.6
cis-aws-foundations-benchmark/v/1.4.0/1.7	security-control/CloudWatch.1
cis-aws-foundations-benchmark/v/1.4.0/1.8	security-control/IAM.15
cis-aws-foundations-benchmark/v/1.4.0/1.9	security-control/IAM.16
cis-aws-foundations-benchmark/v/1.4.0/2.1.2	security-control/S3.5
cis-aws-foundations-benchmark/v/1.4.0/2.1.5.1	security-control/S3.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
cis-aws-foundations-benchmark/v/1.4.0/2.1.5.2	security-control/S3.8
cis-aws-foundations-benchmark/v/1.4.0/2.2.1	security-control/EC2.7
cis-aws-foundations-benchmark/v/1.4.0/2.3.1	security-control/RDS.3
cis-aws-foundations-benchmark/v/1.4.0/3.1	security-control/CloudTrail.1
cis-aws-foundations-benchmark/v/1.4.0/3.2	security-control/CloudTrail.4
cis-aws-foundations-benchmark/v/1.4.0/3.4	security-control/CloudTrail.5
cis-aws-foundations-benchmark/v/1.4.0/3.5	security-control/Config.1
cis-aws-foundations-benchmark/v/1.4.0/3.6	security-control/S3.9
cis-aws-foundations-benchmark/v/1.4.0/3.7	security-control/CloudTrail.2
cis-aws-foundations-benchmark/v/1.4.0/3.8	security-control/KMS.4
cis-aws-foundations-benchmark/v/1.4.0/3.9	security-control/EC2.6
cis-aws-foundations-benchmark/v/1.4.0/4.3	security-control/CloudWatch.1
cis-aws-foundations-benchmark/v/1.4.0/4.4	security-control/CloudWatch.4
cis-aws-foundations-benchmark/v/1.4.0/4.5	security-control/CloudWatch.5
cis-aws-foundations-benchmark/v/1.4.0/4.6	security-control/CloudWatch.6
cis-aws-foundations-benchmark/v/1.4.0/4.7	security-control/CloudWatch.7
cis-aws-foundations-benchmark/v/1.4.0/4.8	security-control/CloudWatch.8
cis-aws-foundations-benchmark/v/1.4.0/4.9	security-control/CloudWatch.9
cis-aws-foundations-benchmark/v/1.4.0/4.10	security-control/CloudWatch.10

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
cis-aws-foundations-benchmark/v/1.4.0/4.11	security-control/CloudWatch.11
cis-aws-foundations-benchmark/v/1.4.0/4.12	security-control/CloudWatch.12
cis-aws-foundations-benchmark/v/1.4.0/4.13	security-control/CloudWatch.13
cis-aws-foundations-benchmark/v/1.4.0/4.14	security-control/CloudWatch.14
cis-aws-foundations-benchmark/v/1.4.0/5.1	security-control/EC2.21
cis-aws-foundations-benchmark/v/1.4.0/5.3	security-control/EC2.2
aws-foundational-security-best-practices/v/1.0.0/Account.1	security-control/Account.1
aws-foundational-security-best-practices/v/1.0.0/ACM.1	security-control/ACM.1
aws-foundational-security-best-practices/v/1.0.0/APIGateway.1	security-control/APIGateway.1
aws-foundational-security-best-practices/v/1.0.0/APIGateway.2	security-control/APIGateway.2
aws-foundational-security-best-practices/v/1.0.0/APIGateway.3	security-control/APIGateway.3
aws-foundational-security-best-practices/v/1.0.0/APIGateway.4	security-control/APIGateway.4
aws-foundational-security-best-practices/v/1.0.0/APIGateway.5	security-control/APIGateway.5
aws-foundational-security-best-practices/v/1.0.0/APIGateway.8	security-control/APIGateway.8

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/APIGateway.9	security-control/APIGateway.9
aws-foundational-security-best-practices/v/1.0.0/AutoScaling.1	security-control/AutoScaling.1
aws-foundational-security-best-practices/v/1.0.0/AutoScaling.2	security-control/AutoScaling.2
aws-foundational-security-best-practices/v/1.0.0/AutoScaling.3	security-control/AutoScaling.3
aws-foundational-security-best-practices/v/1.0.0/Autoscaling.5	security-control/Autoscaling.5
aws-foundational-security-best-practices/v/1.0.0/AutoScaling.6	security-control/AutoScaling.6
aws-foundational-security-best-practices/v/1.0.0/AutoScaling.9	security-control/AutoScaling.9
aws-foundational-security-best-practices/v/1.0.0/CloudFront.1	security-control/CloudFront.1
aws-foundational-security-best-practices/v/1.0.0/CloudFront.3	security-control/CloudFront.3
aws-foundational-security-best-practices/v/1.0.0/CloudFront.4	security-control/CloudFront.4
aws-foundational-security-best-practices/v/1.0.0/CloudFront.5	security-control/CloudFront.5
aws-foundational-security-best-practices/v/1.0.0/CloudFront.6	security-control/CloudFront.6

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/CloudFront.7	security-control/CloudFront.7
aws-foundational-security-best-practices/v/1.0.0/CloudFront.8	security-control/CloudFront.8
aws-foundational-security-best-practices/v/1.0.0/CloudFront.9	security-control/CloudFront.9
aws-foundational-security-best-practices/v/1.0.0/CloudFront.10	security-control/CloudFront.10
aws-foundational-security-best-practices/v/1.0.0/CloudFront.12	security-control/CloudFront.12
aws-foundational-security-best-practices/v/1.0.0/CloudTrail.1	security-control/CloudTrail.1
aws-foundational-security-best-practices/v/1.0.0/CloudTrail.2	security-control/CloudTrail.2
aws-foundational-security-best-practices/v/1.0.0/CloudTrail.4	security-control/CloudTrail.4
aws-foundational-security-best-practices/v/1.0.0/CloudTrail.5	security-control/CloudTrail.5
aws-foundational-security-best-practices/v/1.0.0/CodeBuild.1	security-control/CodeBuild.1
aws-foundational-security-best-practices/v/1.0.0/CodeBuild.2	security-control/CodeBuild.2
aws-foundational-security-best-practices/v/1.0.0/CodeBuild.3	security-control/CodeBuild.3

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/CodeBuild.4	security-control/CodeBuild.4
aws-foundational-security-best-practices/v/1.0.0/Config.1	security-control/Config.1
aws-foundational-security-best-practices/v/1.0.0/DMS.1	security-control/DMS.1
aws-foundational-security-best-practices/v/1.0.0/DynamoDB.1	security-control/DynamoDB.1
aws-foundational-security-best-practices/v/1.0.0/DynamoDB.2	security-control/DynamoDB.2
aws-foundational-security-best-practices/v/1.0.0/DynamoDB.3	security-control/DynamoDB.3
aws-foundational-security-best-practices/v/1.0.0/EC2.1	security-control/EC2.1
aws-foundational-security-best-practices/v/1.0.0/EC2.3	security-control/EC2.3
aws-foundational-security-best-practices/v/1.0.0/EC2.4	security-control/EC2.4
aws-foundational-security-best-practices/v/1.0.0/EC2.6	security-control/EC2.6
aws-foundational-security-best-practices/v/1.0.0/EC2.7	security-control/EC2.7
aws-foundational-security-best-practices/v/1.0.0/EC2.8	security-control/EC2.8

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/EC2.9	security-control/EC2.9
aws-foundational-security-best-practices/v/1.0.0/EC2.10	security-control/EC2.10
aws-foundational-security-best-practices/v/1.0.0/EC2.15	security-control/EC2.15
aws-foundational-security-best-practices/v/1.0.0/EC2.16	security-control/EC2.16
aws-foundational-security-best-practices/v/1.0.0/EC2.17	security-control/EC2.17
aws-foundational-security-best-practices/v/1.0.0/EC2.18	security-control/EC2.18
aws-foundational-security-best-practices/v/1.0.0/EC2.19	security-control/EC2.19
aws-foundational-security-best-practices/v/1.0.0/EC2.2	security-control/EC2.2
aws-foundational-security-best-practices/v/1.0.0/EC2.20	security-control/EC2.20
aws-foundational-security-best-practices/v/1.0.0/EC2.21	security-control/EC2.21
aws-foundational-security-best-practices/v/1.0.0/EC2.23	security-control/EC2.23
aws-foundational-security-best-practices/v/1.0.0/EC2.24	security-control/EC2.24

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/EC2.25	security-control/EC2.25
aws-foundational-security-best-practices/v/1.0.0/ECR.1	security-control/ECR.1
aws-foundational-security-best-practices/v/1.0.0/ECR.2	security-control/ECR.2
aws-foundational-security-best-practices/v/1.0.0/ECR.3	security-control/ECR.3
aws-foundational-security-best-practices/v/1.0.0/ECS.1	security-control/ECS.1
aws-foundational-security-best-practices/v/1.0.0/ECS.10	security-control/ECS.10
aws-foundational-security-best-practices/v/1.0.0/ECS.12	security-control/ECS.12
aws-foundational-security-best-practices/v/1.0.0/ECS.2	security-control/ECS.2
aws-foundational-security-best-practices/v/1.0.0/ECS.3	security-control/ECS.3
aws-foundational-security-best-practices/v/1.0.0/ECS.4	security-control/ECS.4
aws-foundational-security-best-practices/v/1.0.0/ECS.5	security-control/ECS.5
aws-foundational-security-best-practices/v/1.0.0/ECS.8	security-control/ECS.8

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/EFS.1	security-control/EFS.1
aws-foundational-security-best-practices/v/1.0.0/EFS.2	security-control/EFS.2
aws-foundational-security-best-practices/v/1.0.0/EFS.3	security-control/EFS.3
aws-foundational-security-best-practices/v/1.0.0/EFS.4	security-control/EFS.4
aws-foundational-security-best-practices/v/1.0.0/EKS.2	security-control/EKS.2
aws-foundational-security-best-practices/v/1.0.0/ElasticBeanstalk.1	security-control/ElasticBeanstalk.1
aws-foundational-security-best-practices/v/1.0.0/ElasticBeanstalk.2	security-control/ElasticBeanstalk.2
aws-foundational-security-best-practices/v/1.0.0/ELBv2.1	security-control/ELB.1
aws-foundational-security-best-practices/v/1.0.0/ELB.2	security-control/ELB.2
aws-foundational-security-best-practices/v/1.0.0/ELB.3	security-control/ELB.3
aws-foundational-security-best-practices/v/1.0.0/ELB.4	security-control/ELB.4
aws-foundational-security-best-practices/v/1.0.0/ELB.5	security-control/ELB.5

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/ELB.6	security-control/ELB.6
aws-foundational-security-best-practices/v/1.0.0/ELB.7	security-control/ELB.7
aws-foundational-security-best-practices/v/1.0.0/ELB.8	security-control/ELB.8
aws-foundational-security-best-practices/v/1.0.0/ELB.9	security-control/ELB.9
aws-foundational-security-best-practices/v/1.0.0/ELB.10	security-control/ELB.10
aws-foundational-security-best-practices/v/1.0.0/ELB.11	security-control/ELB.11
aws-foundational-security-best-practices/v/1.0.0/ELB.12	security-control/ELB.12
aws-foundational-security-best-practices/v/1.0.0/ELB.13	security-control/ELB.13
aws-foundational-security-best-practices/v/1.0.0/ELB.14	security-control/ELB.14
aws-foundational-security-best-practices/v/1.0.0/EMR.1	security-control/EMR.1
aws-foundational-security-best-practices/v/1.0.0/ES.1	security-control/ES.1
aws-foundational-security-best-practices/v/1.0.0/ES.2	security-control/ES.2

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/ES.3	security-control/ES.3
aws-foundational-security-best-practices/v/1.0.0/ES.4	security-control/ES.4
aws-foundational-security-best-practices/v/1.0.0/ES.5	security-control/ES.5
aws-foundational-security-best-practices/v/1.0.0/ES.6	security-control/ES.6
aws-foundational-security-best-practices/v/1.0.0/ES.7	security-control/ES.7
aws-foundational-security-best-practices/v/1.0.0/ES.8	security-control/ES.8
aws-foundational-security-best-practices/v/1.0.0/GuardDuty.1	security-control/GuardDuty.1
aws-foundational-security-best-practices/v/1.0.0/IAM.1	security-control/IAM.1
aws-foundational-security-best-practices/v/1.0.0/IAM.2	security-control/IAM.2
aws-foundational-security-best-practices/v/1.0.0/IAM.21	security-control/IAM.21
aws-foundational-security-best-practices/v/1.0.0/IAM.3	security-control/IAM.3
aws-foundational-security-best-practices/v/1.0.0/IAM.4	security-control/IAM.4

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/IAM.5	security-control/IAM.5
aws-foundational-security-best-practices/v/1.0.0/IAM.6	security-control/IAM.6
aws-foundational-security-best-practices/v/1.0.0/IAM.7	security-control/IAM.7
aws-foundational-security-best-practices/v/1.0.0/IAM.8	security-control/IAM.8
aws-foundational-security-best-practices/v/1.0.0/Kinesis.1	security-control/Kinesis.1
aws-foundational-security-best-practices/v/1.0.0/KMS.1	security-control/KMS.1
aws-foundational-security-best-practices/v/1.0.0/KMS.2	security-control/KMS.2
aws-foundational-security-best-practices/v/1.0.0/KMS.3	security-control/KMS.3
aws-foundational-security-best-practices/v/1.0.0/Lambda.1	security-control/Lambda.1
aws-foundational-security-best-practices/v/1.0.0/Lambda.2	security-control/Lambda.2
aws-foundational-security-best-practices/v/1.0.0/Lambda.5	security-control/Lambda.5
aws-foundational-security-best-practices/v/1.0.0/NetworkFirewall.3	security-control/NetworkFirewall.3

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/NetworkFirewall.4	security-control/NetworkFirewall.4
aws-foundational-security-best-practices/v/1.0.0/NetworkFirewall.5	security-control/NetworkFirewall.5
aws-foundational-security-best-practices/v/1.0.0/NetworkFirewall.6	security-control/NetworkFirewall.6
aws-foundational-security-best-practices/v/1.0.0/Opensearch.1	security-control/Opensearch.1
aws-foundational-security-best-practices/v/1.0.0/Opensearch.2	security-control/Opensearch.2
aws-foundational-security-best-practices/v/1.0.0/Opensearch.3	security-control/Opensearch.3
aws-foundational-security-best-practices/v/1.0.0/Opensearch.4	security-control/Opensearch.4
aws-foundational-security-best-practices/v/1.0.0/Opensearch.5	security-control/Opensearch.5
aws-foundational-security-best-practices/v/1.0.0/Opensearch.6	security-control/Opensearch.6
aws-foundational-security-best-practices/v/1.0.0/Opensearch.7	security-control/Opensearch.7
aws-foundational-security-best-practices/v/1.0.0/Opensearch.8	security-control/Opensearch.8
aws-foundational-security-best-practices/v/1.0.0/RDS.1	security-control/RDS.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/RDS.10	security-control/RDS.10
aws-foundational-security-best-practices/v/1.0.0/RDS.11	security-control/RDS.11
aws-foundational-security-best-practices/v/1.0.0/RDS.12	security-control/RDS.12
aws-foundational-security-best-practices/v/1.0.0/RDS.13	security-control/RDS.13
aws-foundational-security-best-practices/v/1.0.0/RDS.14	security-control/RDS.14
aws-foundational-security-best-practices/v/1.0.0/RDS.15	security-control/RDS.15
aws-foundational-security-best-practices/v/1.0.0/RDS.16	security-control/RDS.16
aws-foundational-security-best-practices/v/1.0.0/RDS.17	security-control/RDS.17
aws-foundational-security-best-practices/v/1.0.0/RDS.19	security-control/RDS.19
aws-foundational-security-best-practices/v/1.0.0/RDS.2	security-control/RDS.2
aws-foundational-security-best-practices/v/1.0.0/RDS.20	security-control/RDS.20
aws-foundational-security-best-practices/v/1.0.0/RDS.21	security-control/RDS.21

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/RDS.22	security-control/RDS.22
aws-foundational-security-best-practices/v/1.0.0/RDS.23	security-control/RDS.23
aws-foundational-security-best-practices/v/1.0.0/RDS.24	security-control/RDS.24
aws-foundational-security-best-practices/v/1.0.0/RDS.25	security-control/RDS.25
aws-foundational-security-best-practices/v/1.0.0/RDS.3	security-control/RDS.3
aws-foundational-security-best-practices/v/1.0.0/RDS.4	security-control/RDS.4
aws-foundational-security-best-practices/v/1.0.0/RDS.5	security-control/RDS.5
aws-foundational-security-best-practices/v/1.0.0/RDS.6	security-control/RDS.6
aws-foundational-security-best-practices/v/1.0.0/RDS.7	security-control/RDS.7
aws-foundational-security-best-practices/v/1.0.0/RDS.8	security-control/RDS.8
aws-foundational-security-best-practices/v/1.0.0/RDS.9	security-control/RDS.9
aws-foundational-security-best-practices/v/1.0.0/Redshift.1	security-control/Redshift.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/Redshift.2	security-control/Redshift.2
aws-foundational-security-best-practices/v/1.0.0/Redshift.3	security-control/Redshift.3
aws-foundational-security-best-practices/v/1.0.0/Redshift.4	security-control/Redshift.4
aws-foundational-security-best-practices/v/1.0.0/Redshift.6	security-control/Redshift.6
aws-foundational-security-best-practices/v/1.0.0/Redshift.7	security-control/Redshift.7
aws-foundational-security-best-practices/v/1.0.0/Redshift.8	security-control/Redshift.8
aws-foundational-security-best-practices/v/1.0.0/Redshift.9	security-control/Redshift.9
aws-foundational-security-best-practices/v/1.0.0/S3.1	security-control/S3.1
aws-foundational-security-best-practices/v/1.0.0/S3.12	security-control/S3.12
aws-foundational-security-best-practices/v/1.0.0/S3.13	security-control/S3.13
aws-foundational-security-best-practices/v/1.0.0/S3.2	security-control/S3.2
aws-foundational-security-best-practices/v/1.0.0/S3.3	security-control/S3.3

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/S3.5	security-control/S3.5
aws-foundational-security-best-practices/v/1.0.0/S3.6	security-control/S3.6
aws-foundational-security-best-practices/v/1.0.0/S3.8	security-control/S3.8
aws-foundational-security-best-practices/v/1.0.0/S3.9	security-control/S3.9
aws-foundational-security-best-practices/v/1.0.0/SageMaker.1	security-control/SageMaker.1
aws-foundational-security-best-practices/v/1.0.0/SageMaker.2	security-control/SageMaker.2
aws-foundational-security-best-practices/v/1.0.0/SageMaker.3	security-control/SageMaker.3
aws-foundational-security-best-practices/v/1.0.0/SecretsManager.1	security-control/SecretsManager.1
aws-foundational-security-best-practices/v/1.0.0/SecretsManager.2	security-control/SecretsManager.2
aws-foundational-security-best-practices/v/1.0.0/SecretsManager.3	security-control/SecretsManager.3
aws-foundational-security-best-practices/v/1.0.0/SecretsManager.4	security-control/SecretsManager.4
aws-foundational-security-best-practices/v/1.0.0/SQS.1	security-control/SQS.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
aws-foundational-security-best-practices/v/1.0.0/SSM.1	security-control/SSM.1
aws-foundational-security-best-practices/v/1.0.0/SSM.2	security-control/SSM.2
aws-foundational-security-best-practices/v/1.0.0/SSM.3	security-control/SSM.3
aws-foundational-security-best-practices/v/1.0.0/SSM.4	security-control/SSM.4
aws-foundational-security-best-practices/v/1.0.0/WAF.1	security-control/WAF.1
aws-foundational-security-best-practices/v/1.0.0/WAF.2	security-control/WAF.2
aws-foundational-security-best-practices/v/1.0.0/WAF.3	security-control/WAF.3
aws-foundational-security-best-practices/v/1.0.0/WAF.4	security-control/WAF.4
aws-foundational-security-best-practices/v/1.0.0/WAF.6	security-control/WAF.6
aws-foundational-security-best-practices/v/1.0.0/WAF.7	security-control/WAF.7
aws-foundational-security-best-practices/v/1.0.0/WAF.8	security-control/WAF.8
aws-foundational-security-best-practices/v/1.0.0/WAF.10	security-control/WAF.10
pci-dss/v/3.2.1/PCI.AutoScaling.1	security-control/AutoScaling.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
pci-dss/v/3.2.1/PCI.CloudTrail.1	security-control/CloudTrail.2
pci-dss/v/3.2.1/PCI.CloudTrail.2	security-control/CloudTrail.3
pci-dss/v/3.2.1/PCI.CloudTrail.3	security-control/CloudTrail.4
pci-dss/v/3.2.1/PCI.CloudTrail.4	security-control/CloudTrail.5
pci-dss/v/3.2.1/PCI.CodeBuild.1	security-control/CodeBuild.1
pci-dss/v/3.2.1/PCI.CodeBuild.2	security-control/CodeBuild.2
pci-dss/v/3.2.1/PCI.Config.1	security-control/Config.1
pci-dss/v/3.2.1/PCI.CW.1	security-control/CloudWatch.1
pci-dss/v/3.2.1/PCI.DMS.1	security-control/DMS.1
pci-dss/v/3.2.1/PCI.EC2.1	security-control/EC2.1
pci-dss/v/3.2.1/PCI.EC2.2	security-control/EC2.2
pci-dss/v/3.2.1/PCI.EC2.4	security-control/EC2.12
pci-dss/v/3.2.1/PCI.EC2.5	security-control/EC2.13
pci-dss/v/3.2.1/PCI.EC2.6	security-control/EC2.6
pci-dss/v/3.2.1/PCI.ELBv2.1	security-control/ELB.1
pci-dss/v/3.2.1/PCI.ES.1	security-control/ES.2
pci-dss/v/3.2.1/PCI.ES.2	security-control/ES.1
pci-dss/v/3.2.1/PCI.GuardDuty.1	security-control/GuardDuty.1
pci-dss/v/3.2.1/PCI.IAM.1	security-control/IAM.4
pci-dss/v/3.2.1/PCI.IAM.2	security-control/IAM.2

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
pci-dss/v/3.2.1/PCI.IAM.3	security-control/IAM.1
pci-dss/v/3.2.1/PCI.IAM.4	security-control/IAM.6
pci-dss/v/3.2.1/PCI.IAM.5	security-control/IAM.9
pci-dss/v/3.2.1/PCI.IAM.6	security-control/IAM.19
pci-dss/v/3.2.1/PCI.IAM.7	security-control/IAM.8
pci-dss/v/3.2.1/PCI.IAM.8	security-control/IAM.10
pci-dss/v/3.2.1/PCI.KMS.1	security-control/KMS.4
pci-dss/v/3.2.1/PCI.Lambda.1	security-control/Lambda.1
pci-dss/v/3.2.1/PCI.Lambda.2	security-control/Lambda.3
pci-dss/v/3.2.1/PCI.Opensearch.1	security-control/Opensearch.2
pci-dss/v/3.2.1/PCI.Opensearch.2	security-control/Opensearch.1
pci-dss/v/3.2.1/PCI.RDS.1	security-control/RDS.1
pci-dss/v/3.2.1/PCI.RDS.2	security-control/RDS.2
pci-dss/v/3.2.1/PCI.Redshift.1	security-control/Redshift.1
pci-dss/v/3.2.1/PCI.S3.1	security-control/S3.3
pci-dss/v/3.2.1/PCI.S3.2	security-control/S3.2
pci-dss/v/3.2.1/PCI.S3.3	security-control/S3.7
pci-dss/v/3.2.1/PCI.S3.5	security-control/S3.5
pci-dss/v/3.2.1/PCI.S3.6	security-control/S3.1
pci-dss/v/3.2.1/PCI.SageMaker.1	security-control/SageMaker.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
pci-dss/v/3.2.1/PCI.SSM.1	security-control/SSM.2
pci-dss/v/3.2.1/PCI.SSM.2	security-control/SSM.3
pci-dss/v/3.2.1/PCI.SSM.3	security-control/SSM.1
service-managed-aws-control-tower/v/1.0.0/ACM.1	security-control/ACM.1
service-managed-aws-control-tower/v/1.0.0/APIGateway.1	security-control/APIGateway.1
service-managed-aws-control-tower/v/1.0.0/APIGateway.2	security-control/APIGateway.2
service-managed-aws-control-tower/v/1.0.0/APIGateway.3	security-control/APIGateway.3
service-managed-aws-control-tower/v/1.0.0/APIGateway.4	security-control/APIGateway.4
service-managed-aws-control-tower/v/1.0.0/APIGateway.5	security-control/APIGateway.5
service-managed-aws-control-tower/v/1.0.0/AutoScaling.1	security-control/AutoScaling.1
service-managed-aws-control-tower/v/1.0.0/AutoScaling.2	security-control/AutoScaling.2
service-managed-aws-control-tower/v/1.0.0/AutoScaling.3	security-control/AutoScaling.3
service-managed-aws-control-tower/v/1.0.0/AutoScaling.4	security-control/AutoScaling.4

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/Autoscaling.5	security-control/Autoscaling.5
service-managed-aws-control-tower/v/1.0.0/AutoScaling.6	security-control/AutoScaling.6
service-managed-aws-control-tower/v/1.0.0/AutoScaling.9	security-control/AutoScaling.9
service-managed-aws-control-tower/v/1.0.0/CloudTrail.1	security-control/CloudTrail.1
service-managed-aws-control-tower/v/1.0.0/CloudTrail.2	security-control/CloudTrail.2
service-managed-aws-control-tower/v/1.0.0/CloudTrail.4	security-control/CloudTrail.4
service-managed-aws-control-tower/v/1.0.0/CloudTrail.5	security-control/CloudTrail.5
service-managed-aws-control-tower/v/1.0.0/CodeBuild.1	security-control/CodeBuild.1
service-managed-aws-control-tower/v/1.0.0/CodeBuild.2	security-control/CodeBuild.2
service-managed-aws-control-tower/v/1.0.0/CodeBuild.4	security-control/CodeBuild.4
service-managed-aws-control-tower/v/1.0.0/CodeBuild.5	security-control/CodeBuild.5
service-managed-aws-control-tower/v/1.0.0/DMS.1	security-control/DMS.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/DynamoDB.1	security-control/DynamoDB.1
service-managed-aws-control-tower/v/1.0.0/DynamoDB.2	security-control/DynamoDB.2
service-managed-aws-control-tower/v/1.0.0/EC2.1	security-control/EC2.1
service-managed-aws-control-tower/v/1.0.0/EC2.2	security-control/EC2.2
service-managed-aws-control-tower/v/1.0.0/EC2.3	security-control/EC2.3
service-managed-aws-control-tower/v/1.0.0/EC2.4	security-control/EC2.4
service-managed-aws-control-tower/v/1.0.0/EC2.6	security-control/EC2.6
service-managed-aws-control-tower/v/1.0.0/EC2.7	security-control/EC2.7
service-managed-aws-control-tower/v/1.0.0/EC2.8	security-control/EC2.8
service-managed-aws-control-tower/v/1.0.0/EC2.9	security-control/EC2.9
service-managed-aws-control-tower/v/1.0.0/EC2.10	security-control/EC2.10
service-managed-aws-control-tower/v/1.0.0/EC2.15	security-control/EC2.15

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/EC2.16	security-control/EC2.16
service-managed-aws-control-tower/v/1.0.0/EC2.17	security-control/EC2.17
service-managed-aws-control-tower/v/1.0.0/EC2.18	security-control/EC2.18
service-managed-aws-control-tower/v/1.0.0/EC2.19	security-control/EC2.19
service-managed-aws-control-tower/v/1.0.0/EC2.20	security-control/EC2.20
service-managed-aws-control-tower/v/1.0.0/EC2.21	security-control/EC2.21
service-managed-aws-control-tower/v/1.0.0/EC2.22	security-control/EC2.22
service-managed-aws-control-tower/v/1.0.0/ECR.1	security-control/ECR.1
service-managed-aws-control-tower/v/1.0.0/ECR.2	security-control/ECR.2
service-managed-aws-control-tower/v/1.0.0/ECR.3	security-control/ECR.3
service-managed-aws-control-tower/v/1.0.0/ECS.1	security-control/ECS.1
service-managed-aws-control-tower/v/1.0.0/ECS.2	security-control/ECS.2

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/ECS.3	security-control/ECS.3
service-managed-aws-control-tower/v/1.0.0/ECS.4	security-control/ECS.4
service-managed-aws-control-tower/v/1.0.0/ECS.5	security-control/ECS.5
service-managed-aws-control-tower/v/1.0.0/ECS.8	security-control/ECS.8
service-managed-aws-control-tower/v/1.0.0/ECS.10	security-control/ECS.10
service-managed-aws-control-tower/v/1.0.0/ECS.12	security-control/ECS.12
service-managed-aws-control-tower/v/1.0.0/EFS.1	security-control/EFS.1
service-managed-aws-control-tower/v/1.0.0/EFS.2	security-control/EFS.2
service-managed-aws-control-tower/v/1.0.0/EFS.3	security-control/EFS.3
service-managed-aws-control-tower/v/1.0.0/EFS.4	security-control/EFS.4
service-managed-aws-control-tower/v/1.0.0/EKS.2	security-control/EKS.2
service-managed-aws-control-tower/v/1.0.0/ELB.2	security-control/ELB.2

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/ELB.3	security-control/ELB.3
service-managed-aws-control-tower/v/1.0.0/ELB.4	security-control/ELB.4
service-managed-aws-control-tower/v/1.0.0/ELB.5	security-control/ELB.5
service-managed-aws-control-tower/v/1.0.0/ELB.6	security-control/ELB.6
service-managed-aws-control-tower/v/1.0.0/ELB.7	security-control/ELB.7
service-managed-aws-control-tower/v/1.0.0/ELB.8	security-control/ELB.8
service-managed-aws-control-tower/v/1.0.0/ELB.9	security-control/ELB.9
service-managed-aws-control-tower/v/1.0.0/ELB.10	security-control/ELB.10
service-managed-aws-control-tower/v/1.0.0/ELB.12	security-control/ELB.12
service-managed-aws-control-tower/v/1.0.0/ELB.13	security-control/ELB.13
service-managed-aws-control-tower/v/1.0.0/ELB.14	security-control/ELB.14
service-managed-aws-control-tower/v/1.0.0/ELBv2.1	security-control/ELBv2.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/EMR.1	security-control/EMR.1
service-managed-aws-control-tower/v/1.0.0/ES.1	security-control/ES.1
service-managed-aws-control-tower/v/1.0.0/ES.2	security-control/ES.2
service-managed-aws-control-tower/v/1.0.0/ES.3	security-control/ES.3
service-managed-aws-control-tower/v/1.0.0/ES.4	security-control/ES.4
service-managed-aws-control-tower/v/1.0.0/ES.5	security-control/ES.5
service-managed-aws-control-tower/v/1.0.0/ES.6	security-control/ES.6
service-managed-aws-control-tower/v/1.0.0/ES.7	security-control/ES.7
service-managed-aws-control-tower/v/1.0.0/ES.8	security-control/ES.8
service-managed-aws-control-tower/v/1.0.0/ElasticBeanstalk.1	security-control/ElasticBeanstalk.1
service-managed-aws-control-tower/v/1.0.0/ElasticBeanstalk.2	security-control/ElasticBeanstalk.2
service-managed-aws-control-tower/v/1.0.0/GuardDuty.1	security-control/GuardDuty.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/IAM.1	security-control/IAM.1
service-managed-aws-control-tower/v/1.0.0/IAM.2	security-control/IAM.2
service-managed-aws-control-tower/v/1.0.0/IAM.3	security-control/IAM.3
service-managed-aws-control-tower/v/1.0.0/IAM.4	security-control/IAM.4
service-managed-aws-control-tower/v/1.0.0/IAM.5	security-control/IAM.5
service-managed-aws-control-tower/v/1.0.0/IAM.6	security-control/IAM.6
service-managed-aws-control-tower/v/1.0.0/IAM.7	security-control/IAM.7
service-managed-aws-control-tower/v/1.0.0/IAM.8	security-control/IAM.8
service-managed-aws-control-tower/v/1.0.0/IAM.21	security-control/IAM.21
service-managed-aws-control-tower/v/1.0.0/Kinesis.1	security-control/Kinesis.1
service-managed-aws-control-tower/v/1.0.0/KMS.1	security-control/KMS.1
service-managed-aws-control-tower/v/1.0.0/KMS.2	security-control/KMS.2

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/KMS.3	security-control/KMS.3
service-managed-aws-control-tower/v/1.0.0/Lambda.1	security-control/Lambda.1
service-managed-aws-control-tower/v/1.0.0/Lambda.2	security-control/Lambda.2
service-managed-aws-control-tower/v/1.0.0/Lambda.5	security-control/Lambda.5
service-managed-aws-control-tower/v/1.0.0/NetworkFirewall.3	security-control/NetworkFirewall.3
service-managed-aws-control-tower/v/1.0.0/NetworkFirewall.4	security-control/NetworkFirewall.4
service-managed-aws-control-tower/v/1.0.0/NetworkFirewall.5	security-control/NetworkFirewall.5
service-managed-aws-control-tower/v/1.0.0/NetworkFirewall.6	security-control/NetworkFirewall.6
service-managed-aws-control-tower/v/1.0.0/Opensearch.1	security-control/Opensearch.1
service-managed-aws-control-tower/v/1.0.0/Opensearch.2	security-control/Opensearch.2
service-managed-aws-control-tower/v/1.0.0/Opensearch.3	security-control/Opensearch.3
service-managed-aws-control-tower/v/1.0.0/Opensearch.4	security-control/Opensearch.4

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/Opensearch.5	security-control/Opensearch.5
service-managed-aws-control-tower/v/1.0.0/Opensearch.6	security-control/Opensearch.6
service-managed-aws-control-tower/v/1.0.0/Opensearch.7	security-control/Opensearch.7
service-managed-aws-control-tower/v/1.0.0/Opensearch.8	security-control/Opensearch.8
service-managed-aws-control-tower/v/1.0.0/RDS.1	security-control/RDS.1
service-managed-aws-control-tower/v/1.0.0/RDS.2	security-control/RDS.2
service-managed-aws-control-tower/v/1.0.0/RDS.3	security-control/RDS.3
service-managed-aws-control-tower/v/1.0.0/RDS.4	security-control/RDS.4
service-managed-aws-control-tower/v/1.0.0/RDS.5	security-control/RDS.5
service-managed-aws-control-tower/v/1.0.0/RDS.6	security-control/RDS.6
service-managed-aws-control-tower/v/1.0.0/RDS.8	security-control/RDS.8
service-managed-aws-control-tower/v/1.0.0/RDS.9	security-control/RDS.9

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/RDS.10	security-control/RDS.10
service-managed-aws-control-tower/v/1.0.0/RDS.11	security-control/RDS.11
service-managed-aws-control-tower/v/1.0.0/RDS.13	security-control/RDS.13
service-managed-aws-control-tower/v/1.0.0/RDS.17	security-control/RDS.17
service-managed-aws-control-tower/v/1.0.0/RDS.18	security-control/RDS.18
service-managed-aws-control-tower/v/1.0.0/RDS.19	security-control/RDS.19
service-managed-aws-control-tower/v/1.0.0/RDS.20	security-control/RDS.20
service-managed-aws-control-tower/v/1.0.0/RDS.21	security-control/RDS.21
service-managed-aws-control-tower/v/1.0.0/RDS.22	security-control/RDS.22
service-managed-aws-control-tower/v/1.0.0/RDS.23	security-control/RDS.23
service-managed-aws-control-tower/v/1.0.0/RDS.25	security-control/RDS.25
service-managed-aws-control-tower/v/1.0.0/Redshift.1	security-control/Redshift.1

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/Redshift.2	security-control/Redshift.2
service-managed-aws-control-tower/v/1.0.0/Redshift.4	security-control/Redshift.4
service-managed-aws-control-tower/v/1.0.0/Redshift.6	security-control/Redshift.6
service-managed-aws-control-tower/v/1.0.0/Redshift.7	security-control/Redshift.7
service-managed-aws-control-tower/v/1.0.0/Redshift.8	security-control/Redshift.8
service-managed-aws-control-tower/v/1.0.0/Redshift.9	security-control/Redshift.9
service-managed-aws-control-tower/v/1.0.0/S3.1	security-control/S3.1
service-managed-aws-control-tower/v/1.0.0/S3.2	security-control/S3.2
service-managed-aws-control-tower/v/1.0.0/S3.3	security-control/S3.3
service-managed-aws-control-tower/v/1.0.0/S3.5	security-control/S3.5
service-managed-aws-control-tower/v/1.0.0/S3.6	security-control/S3.6
service-managed-aws-control-tower/v/1.0.0/S3.8	security-control/S3.8

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/S3.9	security-control/S3.9
service-managed-aws-control-tower/v/1.0.0/S3.12	security-control/S3.12
service-managed-aws-control-tower/v/1.0.0/S3.13	security-control/S3.13
service-managed-aws-control-tower/v/1.0.0/SageMaker.1	security-control/SageMaker.1
service-managed-aws-control-tower/v/1.0.0/SecretsManager.1	security-control/SecretsManager.1
service-managed-aws-control-tower/v/1.0.0/SecretsManager.2	security-control/SecretsManager.2
service-managed-aws-control-tower/v/1.0.0/SecretsManager.3	security-control/SecretsManager.3
service-managed-aws-control-tower/v/1.0.0/SecretsManager.4	security-control/SecretsManager.4
service-managed-aws-control-tower/v/1.0.0/SQS.1	security-control/SQS.1
service-managed-aws-control-tower/v/1.0.0/SSM.1	security-control/SSM.1
service-managed-aws-control-tower/v/1.0.0/SSM.2	security-control/SSM.2
service-managed-aws-control-tower/v/1.0.0/SSM.3	security-control/SSM.3

GeneratorID before enabling consolidated control findings	GeneratorID after enabling consolidated control findings
service-managed-aws-control-tower/v/1.0.0/SSM.4	security-control/SSM.4
service-managed-aws-control-tower/v/1.0.0/WAF.2	security-control/WAF.2
service-managed-aws-control-tower/v/1.0.0/WAF.3	security-control/WAF.3
service-managed-aws-control-tower/v/1.0.0/WAF.4	security-control/WAF.4

How consolidation impacts control IDs and titles

Consolidated controls view and consolidated control findings standardize control IDs and titles across standards. The terms *security control ID* and *security control title* refer to these standard-agnostic values.

The Security Hub CSPM console displays standard-agnostic security control IDs and security control titles, regardless of whether consolidated control findings is enabled or disabled for your account. However, Security Hub CSPM findings contain standard-specific control titles, for PCI DSS and CIS v1.2.0, if consolidated control findings is disabled for your account. In addition, Security Hub CSPM findings contain the standard-specific control ID and security control ID. For examples of how consolidation impacts control findings, see Samples of control findings.

For controls that are part of the AWS Control Tower service-managed standard, the prefix `CT.` is removed from the control ID and title in findings when consolidated control findings is enabled.

To disable a security control in Security Hub CSPM, you must disable all standard controls that correspond to the security control. The following table shows the mapping of security control IDs and titles to standard-specific control IDs and titles. IDs and titles for controls that belong to the AWS Foundational Security Best Practices (FSBP) standard are already standard-agnostic. For a mapping of controls to the requirements of Center for Internet Security (CIS) v3.0.0, see Mapping of controls to CIS requirements in each version. To run your own scripts on this table, you can download it as a .csv file.

Standard	Standard control ID and title	Security control ID and title
CIS v1.2.0	1.1 Avoid the use of the root user	[CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user
CIS v1.2.0	1.10 Ensure IAM password policy prevents password reuse	[IAM.16] Ensure IAM password policy prevents password reuse
CIS v1.2.0	1.11 Ensure IAM password policy expires passwords within 90 days or less	[IAM.17] Ensure IAM password policy expires passwords within 90 days or less
CIS v1.2.0	1.12 Ensure no root user access key exists	[IAM.4] IAM root user access key should not exist
CIS v1.2.0	1.13 Ensure MFA is enabled for the root user	[IAM.9] MFA should be enabled for the root user
CIS v1.2.0	1.14 Ensure hardware MFA is enabled for the root user	[IAM.6] Hardware MFA should be enabled for the root user
CIS v1.2.0	1.16 Ensure IAM policies are attached only to groups or roles	[IAM.2] IAM users should not have IAM policies attached
CIS v1.2.0	1.2 Ensure multi-factor authentic ation (MFA) is enabled for all IAM users that have a console password	[IAM.5] MFA should be enabled for all IAM users that have a console password
CIS v1.2.0	1.20 Ensure a support role has been created to manage incidents with Support	[IAM.18] Ensure a support role has been created to manage incidents with AWS Support
CIS v1.2.0	1.22 Ensure IAM policies that allow full "*:*" administrative privileges are not created	[IAM.1] IAM policies should not allow full "*" administrative privileges
CIS v1.2.0	1.3 Ensure credentials unused for 90 days or greater are disabled	[IAM.8] Unused IAM user credentials should be removed

Standard	Standard control ID and title	Security control ID and title
CIS v1.2.0	1.4 Ensure access keys are rotated every 90 days or less	[IAM.3] IAM users' access keys should be rotated every 90 days or less
CIS v1.2.0	1.5 Ensure IAM password policy requires at least one uppercase letter	[IAM.11] Ensure IAM password policy requires at least one uppercase letter
CIS v1.2.0	1.6 Ensure IAM password policy requires at least one lowercase letter	[IAM.12] Ensure IAM password policy requires at least one lowercase letter
CIS v1.2.0	1.7 Ensure IAM password policy requires at least one symbol	[IAM.13] Ensure IAM password policy requires at least one symbol
CIS v1.2.0	1.8 Ensure IAM password policy requires at least one number	[IAM.14] Ensure IAM password policy requires at least one number
CIS v1.2.0	1.9 Ensure IAM password policy requires minimum password length of 14 or greater	[IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
CIS v1.2.0	2.1 Ensure CloudTrail is enabled in all regions	[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events
CIS v1.2.0	2.2 Ensure CloudTrail log file validation is enabled	[CloudTrail.4] CloudTrail log file validation should be enabled
CIS v1.2.0	2.3 Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible	[CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
CIS v1.2.0	2.4 Ensure CloudTrail trails are integrated with CloudWatch Logs	[CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs

Standard	Standard control ID and title	Security control ID and title
CIS v1.2.0	2.5 Ensure AWS Config is enabled	[Config.1] AWS Config should be enabled and use the service-linked role for resource recording
CIS v1.2.0	2.6 Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket	[CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket
CIS v1.2.0	2.7 Ensure CloudTrail logs are encrypted at rest using KMS CMKs	[CloudTrail.2] CloudTrail should have encryption at-rest enabled
CIS v1.2.0	2.8 Ensure rotation for customer created CMKs is enabled	[KMS.4] AWS KMS key rotation should be enabled
CIS v1.2.0	2.9 Ensure VPC flow logging is enabled in all VPCs	[EC2.6] VPC flow logging should be enabled in all VPCs
CIS v1.2.0	3.1 Ensure a log metric filter and alarm exist for unauthorized API calls	[CloudWatch.2] Ensure a log metric filter and alarm exist for unauthorized API calls
CIS v1.2.0	3.10 Ensure a log metric filter and alarm exist for security group changes	[CloudWatch.10] Ensure a log metric filter and alarm exist for security group changes
CIS v1.2.0	3.11 Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)	[CloudWatch.11] Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)
CIS v1.2.0	3.12 Ensure a log metric filter and alarm exist for changes to network gateways	[CloudWatch.12] Ensure a log metric filter and alarm exist for changes to network gateways
CIS v1.2.0	3.13 Ensure a log metric filter and alarm exist for route table changes	[CloudWatch.13] Ensure a log metric filter and alarm exist for route table changes

Standard	Standard control ID and title	Security control ID and title
CIS v1.2.0	3.14 Ensure a log metric filter and alarm exist for VPC changes	[CloudWatch.14] Ensure a log metric filter and alarm exist for VPC changes
CIS v1.2.0	3.2 Ensure a log metric filter and alarm exist for Management Console sign-in without MFA	[CloudWatch.3] Ensure a log metric filter and alarm exist for Management Console sign-in without MFA
CIS v1.2.0	3.3 Ensure a log metric filter and alarm exist for usage of root user	[CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user
CIS v1.2.0	3.4 Ensure a log metric filter and alarm exist for IAM policy changes	[CloudWatch.4] Ensure a log metric filter and alarm exist for IAM policy changes
CIS v1.2.0	3.5 Ensure a log metric filter and alarm exist for CloudTrail configura tion changes	[CloudWatch.5] Ensure a log metric filter and alarm exist for CloudTrail configuration changes
CIS v1.2.0	3.6 Ensure a log metric filter and alarm exist for AWS Management Console authentication failures	[CloudWatch.6] Ensure a log metric filter and alarm exist for AWS Management Console authentic ation failures
CIS v1.2.0	3.7 Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs	[CloudWatch.7] Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys
CIS v1.2.0	3.8 Ensure a log metric filter and alarm exist for S3 bucket policy changes	[CloudWatch.8] Ensure a log metric filter and alarm exist for S3 bucket policy changes

Standard	Standard control ID and title	Security control ID and title
CIS v1.2.0	3.9 Ensure a log metric filter and alarm exist for AWS Config configuration changes	[CloudWatch.9] Ensure a log metric filter and alarm exist for AWS Config configuration changes
CIS v1.2.0	4.1 Ensure no security groups allow ingress from 0.0.0.0/0 to port 22	[EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22
CIS v1.2.0	4.2 Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389	[EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
CIS v1.2.0	4.3 Ensure the default security group of every VPC restricts all traffic	[EC2.2] VPC default security groups should not allow inbound or outbound traffic
CIS v1.4.0	1.10 Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password	[IAM.5] MFA should be enabled for all IAM users that have a console password
CIS v1.4.0	1.14 Ensure access keys are rotated every 90 days or less	[IAM.3] IAM users' access keys should be rotated every 90 days or less
CIS v1.4.0	1.16 Ensure IAM policies that allow full "*:*" administrative privileges are not attached	[IAM.1] IAM policies should not allow full "*" administrative privileges
CIS v1.4.0	1.17 Ensure a support role has been created to manage incidents with Support	[IAM.18] Ensure a support role has been created to manage incidents with AWS Support
CIS v1.4.0	1.4 Ensure no root user account access key exists	[IAM.4] IAM root user access key should not exist
CIS v1.4.0	1.5 Ensure MFA is enabled for the root user account	[IAM.9] MFA should be enabled for the root user

Standard	Standard control ID and title	Security control ID and title
CIS v1.4.0	1.6 Ensure hardware MFA is enabled for the root user account	[IAM.6] Hardware MFA should be enabled for the root user
CIS v1.4.0	1.7 Eliminate use of the root user for administrative and daily tasks	[CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user
CIS v1.4.0	1.8 Ensure IAM password policy requires minimum length of 14 or greater	[IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
CIS v1.4.0	1.9 Ensure IAM password policy prevents password reuse	[IAM.16] Ensure IAM password policy prevents password reuse
CIS v1.4.0	2.1.2 Ensure S3 Bucket Policy is set to deny HTTP requests	[S3.5] S3 general purpose buckets should require requests to use SSL
CIS v1.4.0	2.1.5.1 S3 Block Public Access setting should be enabled	[S3.1] S3 general purpose buckets should have block public access settings enabled
CIS v1.4.0	2.1.5.2 S3 Block Public Access setting should be enabled at the bucket level	[S3.8] S3 general purpose buckets should block public access
CIS v1.4.0	2.2.1 Ensure EBS volume encryption is enabled	[EC2.7] EBS default encryption should be enabled
CIS v1.4.0	2.3.1 Ensure that encryption is enabled for RDS Instances	[RDS.3] RDS DB instances should have encryption at-rest enabled
CIS v1.4.0	3.1 Ensure CloudTrail is enabled in all regions	[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events

Standard	Standard control ID and title	Security control ID and title
CIS v1.4.0	3.2 Ensure CloudTrail log file validation is enabled	[CloudTrail.4] CloudTrail log file validation should be enabled
CIS v1.4.0	3.4 Ensure CloudTrail trails are integrated with CloudWatch Logs	[CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs
CIS v1.4.0	3.5 Ensure AWS Config is enabled in all regions	[Config.1] AWS Config should be enabled and use the service-linked role for resource recording
CIS v1.4.0	3.6 Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket	[CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket
CIS v1.4.0	3.7 Ensure CloudTrail logs are encrypted at rest using KMS CMKs	[CloudTrail.2] CloudTrail should have encryption at-rest enabled
CIS v1.4.0	3.8 Ensure rotation for customer created CMKs is enabled	[KMS.4] AWS KMS key rotation should be enabled
CIS v1.4.0	3.9 Ensure VPC flow logging is enabled in all VPCs	[EC2.6] VPC flow logging should be enabled in all VPCs
CIS v1.4.0	4.4 Ensure a log metric filter and alarm exist for IAM policy changes	[CloudWatch.4] Ensure a log metric filter and alarm exist for IAM policy changes
CIS v1.4.0	4.5 Ensure a log metric filter and alarm exist for CloudTrail configura tion changes	[CloudWatch.5] Ensure a log metric filter and alarm exist for CloudTrail configuration changes
CIS v1.4.0	4.6 Ensure a log metric filter and alarm exist for AWS Management Console authentication failures	[CloudWatch.6] Ensure a log metric filter and alarm exist for AWS Management Console authentic ation failures

Standard	Standard control ID and title	Security control ID and title
CIS v1.4.0	4.7 Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs	[CloudWatch.7] Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys
CIS v1.4.0	4.8 Ensure a log metric filter and alarm exist for S3 bucket policy changes	[CloudWatch.8] Ensure a log metric filter and alarm exist for S3 bucket policy changes
CIS v1.4.0	4.9 Ensure a log metric filter and alarm exist for AWS Config configuration changes	[CloudWatch.9] Ensure a log metric filter and alarm exist for AWS Config configuration changes
CIS v1.4.0	4.10 Ensure a log metric filter and alarm exist for security group changes	[CloudWatch.10] Ensure a log metric filter and alarm exist for security group changes
CIS v1.4.0	4.11 Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)	[CloudWatch.11] Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)
CIS v1.4.0	4.12 Ensure a log metric filter and alarm exist for changes to network gateways	[CloudWatch.12] Ensure a log metric filter and alarm exist for changes to network gateways
CIS v1.4.0	4.13 Ensure a log metric filter and alarm exist for route table changes	[CloudWatch.13] Ensure a log metric filter and alarm exist for route table changes
CIS v1.4.0	4.14 Ensure a log metric filter and alarm exist for VPC changes	[CloudWatch.14] Ensure a log metric filter and alarm exist for VPC changes
CIS v1.4.0	5.1 Ensure no Network ACLs allow ingress from 0.0.0.0/0 to remote server administration ports	[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389

Standard	Standard control ID and title	Security control ID and title
CIS v1.4.0	5.3 Ensure the default security group of every VPC restricts all traffic	[EC2.2] VPC default security groups should not allow inbound or outbound traffic
PCI DSS v3.2.1	PCI.AutoScaling.1 Auto scaling groups associated with a load balancer should use load balancer health checks	[AutoScaling.1] Auto Scaling groups associated with a load balancer should use ELB health checks
PCI DSS v3.2.1	PCI.CloudTrail.1 CloudTrail logs should be encrypted at rest using AWS KMS CMKs	[CloudTrail.2] CloudTrail should have encryption at-rest enabled
PCI DSS v3.2.1	PCI.CloudTrail.2 CloudTrail should be enabled	[CloudTrail.3] At least one CloudTrail trail should be enabled
PCI DSS v3.2.1	PCI.CloudTrail.3 CloudTrail log file validation should be enabled	[CloudTrail.4] CloudTrail log file validation should be enabled
PCI DSS v3.2.1	PCI.CloudTrail.4 CloudTrail trails should be integrated with Amazon CloudWatch Logs	[CloudTrail.5] CloudTrail trails should be integrated with Amazon CloudWatch Logs
PCI DSS v3.2.1	PCI.CodeBuild.1 CodeBuild GitHub or Bitbucket source repository URLs should use OAuth	[CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
PCI DSS v3.2.1	PCI.CodeBuild.2 CodeBuild project environment variables should not contain clear text credentials	[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
PCI DSS v3.2.1	PCI.Config.1 AWS Config should be enabled	[Config.1] AWS Config should be enabled and use the service-linked role for resource recording

Standard	Standard control ID and title	Security control ID and title
PCI DSS v3.2.1	PCI.CW.1 A log metric filter and alarm should exist for usage of the "root" user	[CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user
PCI DSS v3.2.1	PCI.DMS.1 Database Migration Service replication instances should not be public	[DMS.1] Database Migration Service replication instances should not be public
PCI DSS v3.2.1	PCI.EC2.1 EBS snapshots should not be publicly restorable	[EC2.1] Amazon EBS snapshots should not be publicly restorable
PCI DSS v3.2.1	PCI.EC2.2 VPC default security group should prohibit inbound and outbound traffic	[EC2.2] VPC default security groups should not allow inbound or outbound traffic
PCI DSS v3.2.1	PCI.EC2.4 Unused EC2 EIPs should be removed	[EC2.12] Unused Amazon EC2 EIPs should be removed
PCI DSS v3.2.1	PCI.EC2.5 Security groups should not allow ingress from 0.0.0.0/0 to port 22	[EC2.13] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 22
PCI DSS v3.2.1	PCI.EC2.6 VPC flow logging should be enabled in all VPCs	[EC2.6] VPC flow logging should be enabled in all VPCs
PCI DSS v3.2.1	PCI.ELBv2.1 Application Load Balancer should be configured to redirect all HTTP requests to HTTPS	[ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS
PCI DSS v3.2.1	PCI.ES.1 Elasticsearch domains should be in a VPC	[ES.2] Elasticsearch domains should not be publicly accessible
PCI DSS v3.2.1	PCI.ES.2 Elasticsearch domains should have encryption at-rest enabled	[ES.1] Elasticsearch domains should have encryption at-rest enabled

Standard	Standard control ID and title	Security control ID and title
PCI DSS v3.2.1	PCI.GuardDuty.1 GuardDuty should be enabled	[GuardDuty.1] GuardDuty should be enabled
PCI DSS v3.2.1	PCI.IAM.1 IAM root user access key should not exist	[IAM.4] IAM root user access key should not exist
PCI DSS v3.2.1	PCI.IAM.2 IAM users should not have IAM policies attached	[IAM.2] IAM users should not have IAM policies attached
PCI DSS v3.2.1	PCI.IAM.3 IAM policies should not allow full "*" administrative privileges	[IAM.1] IAM policies should not allow full "*" administrative privileges
PCI DSS v3.2.1	PCI.IAM.4 Hardware MFA should be enabled for the root user	[IAM.6] Hardware MFA should be enabled for the root user
PCI DSS v3.2.1	PCI.IAM.5 Virtual MFA should be enabled for the root user	[IAM.9] MFA should be enabled for the root user
PCI DSS v3.2.1	PCI.IAM.6 MFA should be enabled for all IAM users	[IAM.19] MFA should be enabled for all IAM users
PCI DSS v3.2.1	PCI.IAM.7 IAM user credentials should be disabled if not used within a pre-defined number days	[IAM.8] Unused IAM user credentials should be removed
PCI DSS v3.2.1	PCI.IAM.8 Password policies for IAM users should have strong configurations	[IAM.10] Password policies for IAM users should have strong configurations
PCI DSS v3.2.1	PCI.KMS.1 Customer master key (CMK) rotation should be enabled	[KMS.4] AWS KMS key rotation should be enabled
PCI DSS v3.2.1	PCI.Lambda.1 Lambda functions should prohibit public access	[Lambda.1] Lambda function policies should prohibit public access

Standard	Standard control ID and title	Security control ID and title
PCI DSS v3.2.1	PCI.Lambda.2 Lambda functions should be in a VPC	[Lambda.3] Lambda functions should be in a VPC
PCI DSS v3.2.1	PCI.Opensearch.1 OpenSearch domains should be in a VPC	[Opensearch.2] OpenSearch domains should not be publicly accessible
PCI DSS v3.2.1	PCI.Opensearch.2 EBS snapshots should not be publicly restorable	[Opensearch.1] OpenSearch domains should have encryption at rest enabled
PCI DSS v3.2.1	PCI.RDS.1 RDS snapshot should be private	[RDS.1] RDS snapshot should be private
PCI DSS v3.2.1	PCI.RDS.2 RDS DB Instances should prohibit public access	[RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration
PCI DSS v3.2.1	PCI.Redshift.1 Amazon Redshift clusters should prohibit public access	[Redshift.1] Amazon Redshift clusters should prohibit public access
PCI DSS v3.2.1	PCI.S3.1 S3 buckets should prohibit public write access	[S3.3] S3 general purpose buckets should block public write access
PCI DSS v3.2.1	PCI.S3.2 S3 buckets should prohibit public read access	[S3.2] S3 general purpose buckets should block public read access
PCI DSS v3.2.1	PCI.S3.3 S3 buckets should have cross-region replication enabled	[S3.7] S3 general purpose buckets should use cross-Region replication
PCI DSS v3.2.1	PCI.S3.5 S3 buckets should require requests to use Secure Socket Layer	[S3.5] S3 general purpose buckets should require requests to use SSL

Standard	Standard control ID and title	Security control ID and title
PCI DSS v3.2.1	PCI.S3.6 S3 Block Public Access setting should be enabled	[S3.1] S3 general purpose buckets should have block public access settings enabled
PCI DSS v3.2.1	PCI.SageMaker.1 Amazon SageMaker notebook instances should not have direct internet access	[SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
PCI DSS v3.2.1	PCI.SSM.1 EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installat ion	[SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation
PCI DSS v3.2.1	PCI.SSM.2 EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT	[SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
PCI DSS v3.2.1	PCI.SSM.3 EC2 instances should be managed by AWS Systems Manager	[SSM.1] Amazon EC2 instances should be managed by AWS Systems Manager

Updating workflows for consolidation

If your workflows don't rely on the specific format of any fields in control findings, no action is required.

If your workflows rely on the specific format of one or more fields in control findings, as noted in the preceding tables, you should update your workflows. For example, If you created an Amazon EventBridge rule that triggered an action for a specific control ID, such as invoking an AWS Lambda function if the control ID equals CIS 2.7, update the rule to use CloudTrail.2, which is the value for the `Compliance.SecurityControlId` field for that control.

If you created custom insights that use any of the fields or values that changed, update those insights to use the new fields or values.

Required top-level ASFF attributes

The following top-level attributes in the AWS Security Finding Format (ASFF) are required for all findings in Security Hub CSPM. For more information about these attributes, see AwsSecurityFinding in the *AWS Security Hub API Reference*.

AwsAccountId

The AWS account ID that the finding applies to.

Example

```
"AwsAccountId": "111111111111"
```

CreatedAt

Indicates when the potential security issue or event captured by a finding was created.

Example

```
"CreatedAt": "2017-03-22T13:22:13.933Z"
```

Description

A finding's description. This field can be nonspecific boilerplate text or details that are specific to the instance of the finding.

For control findings that Security Hub CSPM generates, this field provides a description of the control.

This field doesn't reference a standard if you turn on consolidated control findings.

Example

```
"Description": "This AWS control checks whether AWS Config is enabled in the current
 account and Region."
```

GeneratorId

The identifier for the solution-specific component (a discrete unit of logic) that generated a finding.

For control findings that Security Hub CSPM generates, this field doesn't reference a standard if you turn on consolidated control findings.

Example

```
"GeneratorId": "security-control/Config.1"
```

Id

The product-specific identifier for a finding. For control findings that Security Hub CSPM generates, this field provides the Amazon Resource Name (ARN) of the finding.

This field doesn't reference a standard if you turn on consolidated control findings.

Example

```
"Id": "arn:aws:securityhub:eu-central-1:123456789012:security-control/iam.9/finding/
ab6d6a26-a156-48f0-9403-115983e5a956"
```

ProductArn

The Amazon Resource Name (ARN) generated by Security Hub CSPM that uniquely identifies a third-party findings product after the product is registered with Security Hub CSPM.

The format of this field is `arn:`*`partition`*`:securityhub:`*`region`*`:`*`account-id`*`:product/`*`company-id`*`/`*`product-id`*.

- For AWS services that are integrated with Security Hub CSPM, the `company-id` must be "aws", and the `product-id` must be the AWS public service name. Because AWS products and services aren't associated with an account, the `account-id` section of the ARN is empty. AWS services that are not yet integrated with Security Hub CSPM are considered third-party products.
- For public products, the `company-id` and `product-id` must be the ID values specified at the time of registration.
- For private products, the `company-id` must be the account ID. The `product-id` must be the reserved word "default" or the ID that was specified at the time of registration.

Example

```
// Private ARN
```

```
    "ProductArn": "arn:aws:securityhub:us-east-1:111111111111:product/111111111111/
default"

// Public ARN
    "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty"
    "ProductArn": "arn:aws:securityhub:us-west-2:222222222222:product/generico/secure-
pro"
```

Resources

The `Resources` array of objects provides a set of resource data types that describe the AWS resources that the finding refers to. For details about the fields that a `Resources` object might contain, including which fields are required, see Resource in the *AWS Security Hub API Reference*. For examples of `Resources` objects for specific AWS services, see Resources ASFF object.

Example

```
"Resources": [
  {
    "ApplicationArn": "arn:aws:resource-groups:us-west-2:123456789012:group/
SampleApp/1234567890abcdef0",
    "ApplicationName": "SampleApp",
    "DataClassification": {
    "DetailedResultsLocation": "Path_to_Folder_Or_File",
    "Result": {
        "MimeType": "text/plain",
        "SizeClassified": 2966026,
        "AdditionalOccurrences": false,
        "Status": {
            "Code": "COMPLETE",
            "Reason": "Unsupportedfield"
        },
       "SensitiveData": [
            {
                "Category": "PERSONAL_INFORMATION",
                "Detections": [
                    {
                        "Count": 34,
                        "Type": "GE_PERSONAL_ID",
                        "Occurrences": {
                            "LineRanges": [
                                {
                                    "Start": 1,
```

```
                        "End": 10,
                        "StartColumn": 20
                    }
                ],
                "Pages": [],
                "Records": [],
                "Cells": []
            }
        },
        {
            "Count": 59,
            "Type": "EMAIL_ADDRESS",
            "Occurrences": {
                "Pages": [
                    {
                        "PageNumber": 1,
                        "OffsetRange": {
                            "Start": 1,
                            "End": 100,
                            "StartColumn": 10
                         },
                        "LineRange": {
                            "Start": 1,
                            "End": 100,
                            "StartColumn": 10
                        }
                    }
                ]
            }
        },
        {
            "Count": 2229,
            "Type": "URL",
            "Occurrences": {
               "LineRanges": [
                    {
                        "Start": 1,
                        "End": 13
                    }
               ]
            }
        },
        {
            "Count": 13826,
```

```
                        "Type": "NameDetection",
                        "Occurrences": {
                             "Records": [
                                 {
                                     "RecordIndex": 1,
                                     "JsonPath": "$.ssn.value"
                                 }
                             ]
                         }
                    },
                    {
                        "Count": 32,
                        "Type": "AddressDetection"
                    }
                ],
                "TotalCount": 32
            }
        ],
        "CustomDataIdentifiers": {
            "Detections": [
                 {
                     "Arn": "1712be25e7c7f53c731fe464f1c869b8",
                     "Name": "1712be25e7c7f53c731fe464f1c869b8",
                     "Count": 2
                 }
            ],
            "TotalCount": 2
        }
    }
},
 "Type": "AwsEc2Instance",
 "Id": "arn:aws:ec2:us-west-2:123456789012:instance/i-abcdef01234567890",
 "Partition": "aws",
 "Region": "us-west-2",
 "ResourceRole": "Target",
 "Tags": {
  "billingCode": "Lotus-1-2-3",
  "needsPatching": true
 },
 "Details": {
  "IamInstanceProfileArn": "arn:aws:iam::123456789012:role/IamInstanceProfileArn",
  "ImageId": "ami-79fd7eee",
  "IpV4Addresses": ["1.1.1.1"],
  "IpV6Addresses": ["2001:db8:1234:1a2b::123"],
```

```
  "KeyName": "testkey",
  "LaunchedAt": "2018-09-29T01:25:54Z",
  "MetadataOptions": {
   "HttpEndpoint": "enabled",
   "HttpProtocolIpv6": "enabled",
   "HttpPutResponseHopLimit": 1,
   "HttpTokens": "optional",
   "InstanceMetadataTags": "disabled"
  }
 },
  "NetworkInterfaces": [
  {
   "NetworkInterfaceId": "eni-e5aa89a3"
  }
  ],
  "SubnetId": "PublicSubnet",
  "Type": "i3.xlarge",
  "VirtualizationType": "hvm",
  "VpcId": "TestVPCIpv6"
 }

]
```

SchemaVersion

The schema version that a finding is formatted for. The value of this field must be one of the officially published versions identified by AWS. In the current release, the AWS Security Finding Format schema version is `2018-10-08`.

Example

```
"SchemaVersion": "2018-10-08"
```

Severity

Defines the importance of a finding. For details about this object, see `Severity` in the *AWS Security Hub CSPM API Reference*.

`Severity` is both a top-level object in a finding and nested under the `FindingProviderFields` object.

The value of the top-level `Severity` object for a finding should be updated only by using the `BatchUpdateFindings` API.

To provide severity information, finding providers should update the `Severity` object under `FindingProviderFields` when making a `BatchImportFindings` API request.
If a `BatchImportFindings` request for a new finding only provides `Label` or only provides `Normalized`, Security Hub CSPM automatically populates the value of the other field.
The `Product` and `Original` fields may also be populated.

If the top-level `Finding.Severity` object is present but `Finding.FindingProviderFields` is not present, Security Hub CSPM creates the `FindingProviderFields.Severity` object and copies the entire `Finding.Severity object` into it. This ensures that the original, provider-supplied details are retained within the `FindingProviderFields.Severity` structure, even if the top-level `Severity` object is overwritten.

The finding severity does not consider the criticality of the involved assets or the underlying resource. Criticality is defined as the level of importance of the resources that are associated with the finding. For example, a resource that is associated with a mission critical application has higher criticality than one that is associated with nonproduction testing. To capture information about resource criticality, use the `Criticality` field.

We recommend using the following guidance when translating findings' native severity scores to the value of `Severity.Label` in the ASFF.

- `INFORMATIONAL` – This category may include a finding for a `PASSED`, `WARNING`, or `NOT AVAILABLE` check or a sensitive data identification.
- `LOW` – Findings that could result in future compromises. For example, this category may include vulnerabilities, configuration weaknesses, and exposed passwords.
- `MEDIUM` – Findings that indicate an active compromise, but no indication that an adversary completed their objectives. For example, this category may include malware activity, hacking activity, and unusual behavior detection.
- `HIGH` or `CRITICAL` – Findings that indicate that an adversary completed their objectives, such as active data loss or compromise or a denial of service.

Example

```
"Severity": {
    "Label": "CRITICAL",
    "Normalized": 90,
    "Original": "CRITICAL"
}
```

Title

A finding's title. This field can contain nonspecific boilerplate text or details that are specific to this instance of the finding.

For control findings, this field provides the title of the control. This field doesn't reference a standard if you turn on consolidated control findings.

Example

```
"Title": "AWS Config should be enabled"
```

Types

One or more finding types in the format of *namespace*/*category*/*classifier* that classify a finding. This field doesn't reference a standard if you turn on consolidated control findings.

`Types` should be updated only by using the `BatchUpdateFindings` API.

Finding providers who want to provide a value for `Types` should use the `Types` attribute under `FindingProviderFields`.

In the following list, the top-level bullets are namespaces, the second-level bullets are categories, and the third-level bullets are classifiers. We recommend that finding providers use defined namespaces to help sort and group findings. The defined categories and classifiers may also be used, but are not required. Only the Software and Configuration Checks namespace has defined classifiers.

You may define a partial path for namespace/category/classifier. For example, the following finding types are all valid:

- TTPs
- TTPs/Defense Evasion
- TTPs/Defense Evasion/CloudTrailStopped

The tactics, techniques, and procedures (TTPs) categories in the following list align to the MITRE ATT&CK MatrixTM. The Unusual Behaviors namespace reflects general unusual behavior, such as general statistical anomalies, and are not aligned with a specific TTP. However, you could classify a finding with both Unusual Behaviors and TTPs finding types.

List of namespaces, categories, and classifiers:

- Software and Configuration Checks
  - Vulnerabilities
    - CVE
  - AWS Security Best Practices
    - Network Reachability
    - Runtime Behavior Analysis
  - Industry and Regulatory Standards
    - AWS Foundational Security Best Practices
    - CIS Host Hardening Benchmarks
    - CIS AWS Foundations Benchmark
    - PCI-DSS
    - Cloud Security Alliance Controls
    - ISO 90001 Controls
    - ISO 27001 Controls
    - ISO 27017 Controls
    - ISO 27018 Controls
    - SOC 1
    - SOC 2
    - HIPAA Controls (USA)
    - NIST 800-53 Controls (USA)
    - NIST CSF Controls (USA)
    - IRAP Controls (Australia)
    - K-ISMS Controls (Korea)
    - MTCS Controls (Singapore)
    - FISC Controls (Japan)
    - My Number Act Controls (Japan)
    - ENS Controls (Spain)
    - Cyber Essentials Plus Controls (UK)
    - G-Cloud Controls (UK)

    - C5 Controls (Germany)
    - IT-Grundschutz Controls (Germany)
    - GDPR Controls (Europe)
    - TISAX Controls (Europe)
  - Patch Management
- TTPs
  - Initial Access
  - Execution
  - Persistence
  - Privilege Escalation
  - Defense Evasion
  - Credential Access
  - Discovery
  - Lateral Movement
  - Collection
  - Command and Control
- Effects
  - Data Exposure
  - Data Exfiltration
  - Data Destruction
  - Denial of Service
  - Resource Consumption
- Unusual Behaviors
  - Application
  - Network Flow
  - IP address
  - User
  - VM
  - Container
  - Serverless

  - Process
  - Database
  - Data
- Sensitive Data Identifications
  - PII
  - Passwords
  - Legal
  - Financial
  - Security
  - Business

Example

```
"Types": [
    "Software and Configuration Checks/Vulnerabilities/CVE"
]
```

UpdatedAt

Indicates when the finding provider last updated the finding record.

This timestamp reflects the time when the finding record was last or most recently updated. Consequently, it can differ from the `LastObservedAt` timestamp, which reflects when the event or vulnerability was last or most recently observed.

When you update the finding record, you must update this timestamp to the current timestamp. Upon creation of a finding record, the `CreatedAt` and `UpdatedAt` timestamps must be the same. After an update to the finding record, the value of this field must be more recent than all of the previous values that it contained.

Note that `UpdatedAt` cannot be updated by using the `BatchUpdateFindings` operation. You can update it only by using `BatchImportFindings` operation.

Example

```
"UpdatedAt": "2017-04-22T13:22:13.933Z"
```

Optional top-level ASFF attributes

The following top-level attributes in the AWS Security Finding Format (ASFF) are optional for findings in Security Hub CSPM. For more information about these attributes, see AwsSecurityFinding in the *AWS Security Hub API Reference*.

Action

The `Action` object provides details about an action that affects or was taken on a resource.

Example

```
"Action": {
    "ActionType": "PORT_PROBE",
    "PortProbeAction": {
        "PortProbeDetails": [
            {
                "LocalPortDetails": {
                    "Port": 80,
                    "PortName": "HTTP"
                  },
                "LocalIpDetails": {
                     "IpAddressV4": "192.0.2.0"
                 },
                "RemoteIpDetails": {
                    "Country": {
                        "CountryName": "Example Country"
                    },
                    "City": {
                        "CityName": "Example City"
                    },
                   "GeoLocation": {
                       "Lon": 0,
                       "Lat": 0
                   },
                   "Organization": {
                       "AsnOrg": "ExampleASO",
                       "Org": "ExampleOrg",
                       "Isp": "ExampleISP",
                       "Asn": 64496
                   }
                }
            }
```

```
        ],
        "Blocked": false
    }
}
```

AwsAccountName

The AWS account name that the finding applies to.

Example

```
"AwsAccountName": "jane-doe-testaccount"
```

CompanyName

The name of the company for the product that generated the finding. For control-based findings, the company is AWS.

Security Hub CSPM populates this attribute automatically for each finding. You cannot update it using `BatchImportFindings` or `BatchUpdateFindings`. The exception to this is when you use a custom integration. See the section called “Custom product integrations”.

When you use the Security Hub CSPM console to filter findings by company name, you use this attribute. When you use the Security Hub CSPM API to filter findings by company name, you use the `aws/securityhub/CompanyName` attribute under `ProductFields`. Security Hub CSPM does not synchronize those two attributes.

Example

```
"CompanyName": "AWS"
```

Compliance

The `Compliance` object typically provides details about a control finding, such as applicable standards and the status of the control check.

Example

```
"Compliance": {
    "AssociatedStandards": [
        {"StandardsId": "standards/aws-foundational-security-best-practices/v/1.0.0"},
        {"StandardsId": "standards/service-managed-aws-control-tower/v/1.0.0"},
```

```
        {"StandardsId": "standards/nist-800-53/v/5.0.0"}
    ],
    "RelatedRequirements": [
        "NIST.800-53.r5 AC-4",
        "NIST.800-53.r5 AC-4(21)",
        "NIST.800-53.r5 SC-7",
        "NIST.800-53.r5 SC-7(11)",
        "NIST.800-53.r5 SC-7(16)",
        "NIST.800-53.r5 SC-7(21)",
        "NIST.800-53.r5 SC-7(4)",
        "NIST.800-53.r5 SC-7(5)"
    ],
    "SecurityControlId": "EC2.18",
    "SecurityControlParameters":[
        {
            "Name": "authorizedTcpPorts",
            "Value": ["80", "443"]
        },
        {
            "Name": "authorizedUdpPorts",
            "Value": ["427"]
        }
    ],
    "Status": "NOT_AVAILABLE",
    "StatusReasons": [
        {
            "ReasonCode": "CONFIG_RETURNS_NOT_APPLICABLE",
            "Description": "This finding has a compliance status of NOT AVAILABLE
 because AWS Config sent Security Hub CSPM a finding with a compliance state of Not
 Applicable. The potential reasons for a Not Applicable finding from Config are that
 (1) a resource has been moved out of scope of the Config rule; (2) the Config rule has
 been deleted; (3) the resource has been deleted; or (4) the logic of the Config rule
 itself includes scenarios where Not Applicable is returned. The specific reason why
 Not Applicable is returned is not available in the Config rule evaluation."
        }
    ]
}
```

Confidence

The likelihood that a finding accurately identifies the behavior or issue that it was intended to identify.

`Confidence` should only be updated using `BatchUpdateFindings`.

Finding providers who want to provide a value for `Confidence` should use the `Confidence` attribute under `FindingProviderFields`. See the section called "Updating findings with FindingProviderFields".

`Confidence` is scored on a 0–100 basis using a ratio scale. 0 means 0 percent confidence, and 100 means 100 percent confidence. For example, a data exfiltration detection based on a statistical deviation of network traffic has low confidence because an actual exfiltration hasn't been verified.

Example

```
"Confidence": 42
```

Criticality

The level of importance that is assigned to the resources that are associated with a finding.

`Criticality` should only be updated by calling the `BatchUpdateFindings` API operation. Don't update this object with `BatchImportFindings`.

Finding providers who want to provide a value for `Criticality` should use the `Criticality` attribute under `FindingProviderFields`. See the section called "Updating findings with FindingProviderFields".

`Criticality` is scored on a 0–100 basis, using a ratio scale that supports only full integers. A score of 0 means that the underlying resources have no criticality, and a score of 100 is reserved for the most critical resources.

For each resource, consider the following when assigning `Criticality`:

- Does the affected resource contain sensitive data (for example, an S3 bucket with PII)?
- Does the affected resource enable an adversary to deepen their access or extend their capabilities to carry out additional malicious activity (for example, a compromised sysadmin account)?
- Is the resource a business-critical asset (for example, a key business system that if compromised could have significant revenue impact)?

You can use the following guidelines:

- A resource powering mission-critical systems or containing highly sensitive data can be scored in the 75–100 range.

- A resource powering important (but not critical systems) or containing moderately important data can be scored in the 25–74 range.
- A resource powering unimportant systems or containing nonsensitive data should be scored in the 0–24 range.

Example

```
"Criticality": 99
```

Detection

The `Detection` object provides details about an attack sequence finding from Amazon GuardDuty Extended Threat Detection. GuardDuty generates an attack sequence finding when multiple events align to a potentially suspicious activity. To receive GuardDuty attack sequence findings in AWS Security Hub CSPM, you must have GuardDuty enabled in your account. For more information, see Amazon GuardDuty Extended Threat Detection in the *Amazon GuardDuty User Guide*.

Example

```
"Detection": {
    "Sequence": {
     "Uid": "1111111111111-184ec3b9-cf8d-452d-9aad-f5bdb7afb010",
     "Actors": [{
      "Id": "USER:AROA987654321EXAMPLE:i-b188560f:1234567891",
      "Session": {
       "Uid": "1234567891",
       "MfAStatus": "DISABLED",
       "CreatedTime": "1716916944000",
       "Issuer": "arn:aws:s3:::amzn-s3-demo-destination-bucket"
      },
      "User": {
       "CredentialUid": "ASIAIOSFODNN7EXAMPLE",
       "Name": "ec2_instance_role_production",
       "Type": "AssumedRole",
       "Uid": "AROA987654321EXAMPLE:i-b188560f",
       "Account": {
        "Uid": "AccountId",
        "Name": "AccountName"
       }
      }
     }],
```

```
     "Endpoints": [{
      "Id": "EndpointId",
      "Ip": "203.0.113.1",
      "Domain": "example.com",
      "Port": 4040,
      "Location": {
       "City": "New York",
       "Country": "US",
       "Lat": 40.7123,
       "Lon": -74.0068
      },
      "AutonomousSystem": {
       "Name": "AnyCompany",
       "Number": 64496
      },
      "Connection": {
       "Direction": "INBOUND"
      }
     }],
     "Signals": [{
      "Id": "arn:aws:guardduty:us-east-1:123456789012:detector/
d0bfe135ab8b4dd8c3eaae7df9900073/finding/535a382b1bcc44d6b219517a29058fb7",
      "Title": "Someone ran a penetration test tool on your account.",
      "ActorIds": ["USER:AROA987654321EXAMPLE:i-b188560f:1234567891"],
      "Count": 19,
      "FirstSeenAt": 1716916943000,
      "SignalIndicators": [
       {
        "Key": "ATTACK_TACTIC",
        "Title": "Attack Tactic",
        "Values": [
         "Impact"
        ]
       },
       {
        "Key": "HIGH_RISK_API",
        "Title": "High Risk Api",
        "Values": [
         "s3:DeleteObject"
        ]
       },
       {
        "Key": "ATTACK_TECHNIQUE",
        "Title": "Attack Technique",
```

```
        "Values": [
         "Data Destruction"
        ]
       },
      ],
      "LastSeenAt": 1716916944000,
      "Name": "Test:IAMUser/KaliLinux",
      "ResourceIds": [
       "arn:aws:s3:::amzn-s3-demo-destination-bucket"
      ],
      "Type": "FINDING"
     }],
     "SequenceIndicators": [
      {
       "Key": "ATTACK_TACTIC",
       "Title": "Attack Tactic",
       "Values": [
        "Discovery",
        "Exfiltration",
        "Impact"
       ]
      },
      {
       "Key": "HIGH_RISK_API",
       "Title": "High Risk Api",
       "Values": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListBuckets"
        "s3:ListObjects"
       ]
      },
      {
       "Key": "ATTACK_TECHNIQUE",
       "Title": "Attack Technique",
       "Values": [
        "Cloud Service Discovery",
        "Data Destruction"
       ]
      }
     ]
    }
}
```

FindingProviderFields

`FindingProviderFields` includes the following attributes:

- `Confidence`
- `Criticality`
- `RelatedFindings`
- `Severity`
- `Types`

The preceding fields are nested under the `FindingProviderFields` object, but have analogues of the same name as top-level ASFF fields. When a new finding is sent to Security Hub CSPM by a finding provider, Security Hub CSPM populates the `FindingProviderFields` object automatically if it is empty based on the corresponding top-level fields.

Finding providers can update `FindingProviderFields` by using the`BatchImportFindings` operation of the Security Hub CSPM API. Finding providers cannot update this object with `BatchUpdateFindings`.

For details on how Security Hub CSPM handles updates from `BatchImportFindings` to `FindingProviderFields` and to the corresponding top-level attributes, see the section called "Updating findings with FindingProviderFields".

Customers can update the top-level fields by using the `BatchUpdateFindings` operation. Customers can't update `FindingProviderFields`.

Example

```
"FindingProviderFields": {
    "Confidence": 42,
    "Criticality": 99,
    "RelatedFindings":[
      {
        "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",
        "Id": "123e4567-e89b-12d3-a456-426655440000"
      }
    ],
    "Severity": {
        "Label": "MEDIUM",
        "Original": "MEDIUM"
```

```
    },
    "Types": [ "Software and Configuration Checks/Vulnerabilities/CVE" ]
}
```

FirstObservedAt

Indicates when the potential security issue or event captured by a finding was first observed.

This timestamp specifies when the event or vulnerability was first observed. Consequently, it can differ from the `CreatedAt` timestamp, which reflects when this finding record was created.

For control findings that Security Hub CSPM generates and updates, this timestamp can also indicate when the compliance status of a resource most recently changed. For other types of findings, this timestamp should be immutable between updates of the finding record, but can be updated if a more accurate timestamp is determined.

Example

```
"FirstObservedAt": "2017-03-22T13:22:13.933Z"
```

LastObservedAt

Indicates when the potential security issue or event captured by a finding was most recently observed by the security findings product.

This timestamp specifies when the event or vulnerability was last or most recently observed. Consequently, it can differ from the `UpdatedAt` timestamp, which reflects when this finding record was last or most recently updated.

You can provide this timestamp, but it isn't required upon first observation. If you populate this field upon first observation, the timestamp should be the same as the `FirstObservedAt` timestamp. You should update this field to reflect the last or most recently observed timestamp each time a finding is observed.

Example

```
"LastObservedAt": "2017-03-23T13:22:13.933Z"
```

Malware

The `Malware` object provides a list of malware related to a finding.

Example

```
"Malware": [
    {
        "Name": "Stringler",
        "Type": "COIN_MINER",
        "Path": "/usr/sbin/stringler",
        "State": "OBSERVED"
    }
]
```

Network (Retired)

The `Network` object provides network-related information about a finding.

This object is retired. To provide this data, you can either map the data to a resource in `Resources`, or use the `Action` object.

Example

```
"Network": {
    "Direction": "IN",
    "OpenPortRange": {
        "Begin": 443,
        "End": 443
    },
    "Protocol": "TCP",
    "SourceIpV4": "1.2.3.4",
    "SourceIpV6": "FE80:CD00:0000:0CDE:1257:0000:211E:729C",
    "SourcePort": "42",
    "SourceDomain": "example1.com",
    "SourceMac": "00:0d:83:b1:c0:8e",
    "DestinationIpV4": "2.3.4.5",
    "DestinationIpV6": "FE80:CD00:0000:0CDE:1257:0000:211E:729C",
    "DestinationPort": "80",
    "DestinationDomain": "example2.com"
}
```

NetworkPath

The `NetworkPath` object provides information about a network path that is related to a finding. Each entry in `NetworkPath` represents a component of the path.

Example

```
"NetworkPath" : [
    {
        "ComponentId": "abc-01a234bc56d8901ee",
        "ComponentType": "AWS::EC2::InternetGateway",
        "Egress": {
            "Destination": {
                "Address": [ "192.0.2.0/24" ],
                "PortRanges": [
                    {
                        "Begin": 443,
                        "End": 443
                    }
                ]
            },
            "Protocol": "TCP",
            "Source": {
                "Address": ["203.0.113.0/24"]
            }
        },
        "Ingress": {
            "Destination": {
                "Address": [ "198.51.100.0/24" ],
                "PortRanges": [
                    {
                        "Begin": 443,
                        "End": 443
                    }
                 ]
            },
            "Protocol": "TCP",
            "Source": {
                "Address": [ "203.0.113.0/24" ]
            }
        }
     }
]
```

Note

The `Note` object specifies a user-defined note that you can add to a finding.

A finding provider can provide an initial note for a finding, but cannot add notes after that. You can only update a note using `BatchUpdateFindings`.

Example

```
"Note": {
    "Text": "Don't forget to check under the mat.",
    "UpdatedBy": "jsmith",
    "UpdatedAt": "2018-08-31T00:15:09Z"
}
```

PatchSummary

The `PatchSummary` object provides a summary of the patch compliance status for an instance against a selected compliance standard.

Example

```
"PatchSummary" : {
    "FailedCount" : 0,
    "Id" : "pb-123456789098",
    "InstalledCount" : 100,
    "InstalledOtherCount" : 1023,
    "InstalledPendingReboot" : 0,
    "InstalledRejectedCount" : 0,
    "MissingCount" : 100,
    "Operation" : "Install",
    "OperationEndTime" : "2018-09-27T23:39:31Z",
    "OperationStartTime" : "2018-09-27T23:37:31Z",
    "RebootOption" : "RebootIfNeeded"
}
```

Process

The `Process` object provides process-related details about a finding.

Example:

```
"Process": {
    "LaunchedAt": "2018-09-27T22:37:31Z",
    "Name": "syslogd",
    "ParentPid": 56789,
    "Path": "/usr/sbin/syslogd",
```

```
    "Pid": 12345,
    "TerminatedAt": "2018-09-27T23:37:31Z"
}
```

ProcessedAt

Indicates when Security Hub CSPM received a finding and began to process it.

This differs from `CreatedAt` and `UpdatedAt`, which are required timestamps that relate to the finding provider's interaction with the security issue and finding. The `ProcessedAt` timestamp indicates when Security Hub CSPM starts to process a finding. A finding appears in a user's account after processing is complete.

```
"ProcessedAt": "2023-03-23T13:22:13.933Z"
```

ProductFields

A data type where security findings products can include additional solution-specific details that are not part of the defined AWS Security Finding Format.

For findings generated by Security Hub CSPM controls, `ProductFields` includes information about the control. See the section called “Generating and updating control findings”.

This field should not contain redundant data and must not contain data that conflicts with AWS Security Finding Format fields.

The "`aws/`" prefix represents a reserved namespace for AWS products and services only and must not be submitted with findings from third-party integrations.

Although not required, products should format field names as `company-id/product-id/field-name`, where the `company-id` and `product-id` match those supplied in the `ProductArn` of the finding.

The fields referencing `Archival` are used when Security Hub CSPM archives an existing finding. For example, Security Hub CSPM archives existing findings when you disable a control or standard and when you turn consolidated control findings on or off.

This field may also include information about the standard that includes the control that produced the finding.

Example

```
"ProductFields": {
    "API", "DeleteTrail",
    "ArchivalReasons:0/Description": "The finding is in an ARCHIVED state because
 consolidated control findings has been turned on or off. This causes findings in the
 previous state to be archived when new findings are being generated.",
    "ArchivalReasons:0/ReasonCode": "CONSOLIDATED_CONTROL_FINDINGS_UPDATE",
    "aws/inspector/AssessmentTargetName": "My prod env",
    "aws/inspector/AssessmentTemplateName": "My daily CVE assessment",
    "aws/inspector/RulesPackageName": "Common Vulnerabilities and Exposures",
    "generico/secure-pro/Action.Type", "AWS_API_CALL",
    "generico/secure-pro/Count": "6",
    "Service_Name": "cloudtrail.amazonaws.com"
}
```

ProductName

Provides the name of the product that generated the finding. For control-based findings, the product name is Security Hub CSPM.

Security Hub CSPM populates this attribute automatically for each finding. You cannot update it using `BatchImportFindings` or `BatchUpdateFindings`. The exception to this is when you use a custom integration. See the section called “Custom product integrations”.

When you use the Security Hub CSPM console to filter findings by product name, you use this attribute.

When you use the Security Hub CSPM API to filter findings by product name, you use the `aws/securityhub/ProductName` attribute under `ProductFields`.

Security Hub CSPM does not synchronize those two attributes.

RecordState

Provides the record state of a finding.

By default, when initially generated by a service, findings are considered `ACTIVE`.

The `ARCHIVED` state indicates that a finding should be hidden from view. Archived findings are not deleted immediately. You can search, review, and report on them. Security Hub CSPM automatically archives control-based findings if the associated resource is deleted, the resource does not exist, or the control is disabled.

`RecordState` is intended for finding providers, and can be updated only by using the `BatchImportFindings` operation. You cannot update it by using the `BatchUpdateFindings` operation.

To track the status of your investigation into a finding, use `Workflow` instead of `RecordState`.

If the record state changes from `ARCHIVED` to `ACTIVE`, and the workflow status of the finding is `NOTIFIED` or `RESOLVED`, Security Hub CSPM automatically changes the workflow status to `NEW`.

Example

```
"RecordState": "ACTIVE"
```

Region

Specifies the AWS Region from which the finding was generated.

Security Hub CSPM populates this attribute automatically for each finding. You cannot update it using `BatchImportFindings` or `BatchUpdateFindings`.

Example

```
"Region": "us-west-2"
```

RelatedFindings

Provides a list of findings that are related to the current finding.

`RelatedFindings` should only be updated with the `BatchUpdateFindings` API operation. You should not update this object with `BatchImportFindings`.

For `BatchImportFindings` requests, finding providers should use the `RelatedFindings` object under `FindingProviderFields`.

To view descriptions of `RelatedFindings` attributes, see `RelatedFinding` in the *AWS Security Hub CSPM API Reference*.

Example

```
"RelatedFindings": [
    { "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",
      "Id": "123e4567-e89b-12d3-a456-426655440000" },
```

```
    { "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",
      "Id": "AcmeNerfHerder-111111111111-x189dx7824" }
]
```

RiskAssessment

Example

```
"RiskAssessment": {
    "Posture": {
        "FindingTotal": 4,
        "Indicators": [
            {
                "Type": "Reachability",
                "Findings": [
                    {
                        "Id": "arn:aws:inspector2:us-
east-2:123456789012:finding/1234567890abcdef0",
                        "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
inspector",
                        "Title": "Finding title"
                    },
                    {
                        "Id": "arn:aws:inspector2:us-east-2:123456789012:finding/
abcdef01234567890",
                        "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
inspector",
                        "Title": "Finding title"
                    }
                ]
            },
            {
                "Type": "Vulnerability",
                "Findings": [
                    {
                        "Id": "arn:aws:inspector2:us-
east-2:123456789012:finding/021345abcdef6789",
                        "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
inspector",
                        "Title": "Finding title"
                    },
                    {
                        "Id": "arn:aws:inspector2:us-
east-2:123456789012:finding/021345ghijkl6789",
```

```
                        "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
inspector",
                        "Title": "Finding title"
                    }
                ]
            }
        ]
    }
}
```

Remediation

The `Remediation` object provides information about recommended remediation steps to address the finding.

Example

```
"Remediation": {
    "Recommendation": {
        "Text": "For instructions on how to fix this issue, see the AWS Security Hub
 CSPM documentation for EC2.2.",
        "Url": "https://docs.aws.amazon.com/console/securityhub/EC2.2/remediation"
    }
}
```

Sample

Specifies whether the finding is a sample finding.

```
"Sample": true
```

SourceUrl

The `SourceUrl` object provides a URL that links to a page about the current finding in the finding product.

```
"SourceUrl": "http://sourceurl.com"
```

ThreatIntelIndicators

The `ThreatIntelIndicator` object provides threat intelligence details that are related to a finding.

Example

```
"ThreatIntelIndicators": [
  {
    "Category": "BACKDOOR",
    "LastObservedAt": "2018-09-27T23:37:31Z",
    "Source": "Threat Intel Weekly",
    "SourceUrl": "http://threatintelweekly.org/backdoors/8888",
    "Type": "IPV4_ADDRESS",
    "Value": "8.8.8.8",
  }
]
```

Threats

The Threats object provides details about the threat detected by a finding.

Example

```
"Threats": [{
    "FilePaths": [{
        "FileName": "b.txt",
        "FilePath": "/tmp/b.txt",
        "Hash": "sha256",
        "ResourceId": "arn:aws:ec2:us-west-2:123456789012:volume/vol-032f3bdd89aee112f"
    }],
    "ItemCount": 3,
    "Name": "Iot.linux.mirai.vwisi",
    "Severity": "HIGH"
}]
```

UserDefinedFields

Provides a list of name-value string pairs that are associated with the finding. These are custom, user-defined fields that are added to a finding. These fields can be generated automatically through your specific configuration.

Finding providers should not use this field for data that the product generates. Instead, finding providers can use the `ProductFields` field for data that does not map to any standard AWS Security Finding Format field.

These fields can only be updated using `BatchUpdateFindings`.

Example

```
"UserDefinedFields": {
    "reviewedByCio": "true",
    "comeBackToLater": "Check this again on Monday"
}
```

VerificationState

Provides the veracity of a finding. Findings products can provide a value of `UNKNOWN` for this field. A findings product should provide a value for this field if there is a meaningful analog in the findings product's system. This field is typically populated by a user determination or action after investigating a finding.

A finding provider can provide an initial value for this attribute, but cannot update it after that. You can only update this attribute by using `BatchUpdateFindings`.

```
"VerificationState": "Confirmed"
```

Vulnerabilities

The Vulnerabilities object provides a list of vulnerabilities that are associated with a finding.

Example

```
"Vulnerabilities" : [
    {
        "CodeVulnerabilities": [{
            "Cwes": [
                "CWE-798",
                "CWE-799"
            ],
            "FilePath": {
                "EndLine": 421,
                "FileName": "package-lock.json",
                "FilePath": "package-lock.json",
                "StartLine": 420
            },
                "SourceArn":"arn:aws:lambda:us-east-1:123456789012:layer:AWS-AppConfig-
Extension:114"
        }],
        "Cvss": [
```

```
            {
                "BaseScore": 4.7,
                "BaseVector": "AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N",
                "Version": "V3"
            },
            {
                "BaseScore": 4.7,
                "BaseVector": "AV:L/AC:M/Au:N/C:C/I:N/A:N",
                "Version": "V2"
            }
        ],
        "EpssScore": 0.015,
        "ExploitAvailable": "YES",
        "FixAvailable": "YES",
        "Id": "CVE-2020-12345",
        "LastKnownExploitAt": "2020-01-16T00:01:35Z",
        "ReferenceUrls":[
           "http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-12418",
            "http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-17563"
        ],
        "RelatedVulnerabilities": ["CVE-2020-12345"],
        "Vendor": {
            "Name": "Alas",
            "Url":"https://alas.aws.amazon.com/ALAS-2020-1337.html",
            "VendorCreatedAt":"2020-01-16T00:01:43Z",
            "VendorSeverity":"Medium",
            "VendorUpdatedAt":"2020-01-16T00:01:43Z"
        },
        "VulnerablePackages": [
            {
                "Architecture": "x86_64",
                "Epoch": "1",
                "FilePath": "/tmp",
                "FixedInVersion": "0.14.0",
                "Name": "openssl",
                "PackageManager": "OS",
                "Release": "16.amzn2.0.3",
                "Remediation": "Update aws-crt to 0.14.0",
                "SourceLayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:id",
                "SourceLayerHash":
 "sha256:c1962c35b63a6ff6ce7df6e042ee82371a605ca9515569edec46ff14f926f001",
                "Version": "1.0.2k"
            }
        ]
```

```
    }
]
```

Workflow

The `Workflow` object provides information about the status of the investigation into a finding.

This field is intended for customers to use with remediation, orchestration, and ticketing tools. It is not intended for finding providers.

You can only update the `Workflow` field with `BatchUpdateFindings`. Customers can also update it from the console. See the section called "Setting the workflow status of findings".

Example

```
"Workflow": {
    "Status": "NEW"
}
```

WorkflowState (Retired)

This object is retired and has been replaced by the `Status` field of the `Workflow` object.

This field provides the workflow state of a finding. Findings products can provide the value of `NEW` for this field. A findings product can provide a value for this field if there is a meaningful analog in the findings product's system.

Example

```
"WorkflowState": "NEW"
```

Resources ASFF object

In the AWS Security Finding Format (ASFF), the `Resources` object provides information about the resources involved in a finding. It contains an array of up to 32 resource objects. To determine how resource names are formatted, see AWS Security Finding Format (ASFF). For examples of each resource object, select a resource from the following list.

Topics

- Resource attributes in the ASFF

- AwsAmazonMQ resources in ASFF
- AwsApiGateway resources in ASFF
- AwsAppSync resources in ASFF
- AwsAthena resources in ASFF
- AwsAutoScaling resources in ASFF
- AwsBackup resources in ASFF
- AwsCertificateManager resources in ASFF
- AwsCloudFormation resources in ASFF
- AwsCloudFront resources in ASFF
- AwsCloudTrail resources in ASFF
- AwsCloudWatch resources in ASFF
- AwsCodeBuild resources in ASFF
- AwsDms resources in ASFF
- AwsDynamoDB resources in ASFF
- AwsEc2 resources in ASFF
- AwsEcr resources in ASFF
- AwsEcs resources in ASFF
- AwsEfs resources in ASFF
- AwsEks resources in ASFF
- AwsElasticBeanstalk resources in ASFF
- AwsElasticSearch resources in ASFF
- AwsElb resources in ASFF
- AwsEventBridge resources in ASFF
- AwsGuardDuty resources in ASFF
- AwsIam resources in ASFF
- AwsKinesis resources in ASFF
- AwsKms resources in ASFF
- AwsLambda
- AwsMsk resources in ASFF
- AwsNetworkFirewall resources in ASFF

- AwsOpenSearchService resources in ASFF
- AwsRds resources in ASFF
- AwsRedshift resources in ASFF
- AwsRoute53 resources in ASFF
- AwsS3 resources in ASFF
- AwsSageMaker resources in ASFF
- AwsSecretsManager resources in ASFF
- AwsSns resources in ASFF
- AwsSqs resources in ASFF
- AwsSsm resources in ASFF
- AwsStepFunctions resources in ASFF
- AwsWaf resources in ASFF
- AwsXray resources in ASFF
- CodeRepository object in ASFF
- Container object in ASFF
- Other object in ASFF

Resource attributes in the ASFF

Here are descriptions and examples for the `Resources` object in the AWS Security Finding Format (ASFF). For more information about these fields, see Resources.

ApplicationArn

Identifies the Amazon Resource Name (ARN) of the application involved in the finding.

Example

```
"ApplicationArn": "arn:aws:resource-groups:us-west-2:123456789012:group/
SampleApp/1234567890abcdef0"
```

ApplicationName

Identifies the name of the application involved in the finding.

Example

```
"ApplicationName": "SampleApp"
```

DataClassification

The DataClassification field provides information about sensitive data that was detected on the resource.

Example

```
"DataClassification": {
    "DetailedResultsLocation": "Path_to_Folder_Or_File",
    "Result": {
        "MimeType": "text/plain",
        "SizeClassified": 2966026,
        "AdditionalOccurrences": false,
        "Status": {
            "Code": "COMPLETE",
            "Reason": "Unsupportedfield"
        },
       "SensitiveData": [
            {
                "Category": "PERSONAL_INFORMATION",
                "Detections": [
                    {
                        "Count": 34,
                        "Type": "GE_PERSONAL_ID",
                        "Occurrences": {
                            "LineRanges": [
                                {
                                    "Start": 1,
                                    "End": 10,
                                    "StartColumn": 20
                                }
                            ],
                            "Pages": [],
                            "Records": [],
                            "Cells": []
                        }
                    },
                    {
                        "Count": 59,
                        "Type": "EMAIL_ADDRESS",
                        "Occurrences": {
```

```
                    "Pages": [
                        {
                            "PageNumber": 1,
                            "OffsetRange": {
                                "Start": 1,
                                "End": 100,
                                "StartColumn": 10
                             },
                            "LineRange": {
                                "Start": 1,
                                "End": 100,
                                "StartColumn": 10
                            }
                        }
                    ]
                }
            },
            {
                "Count": 2229,
                "Type": "URL",
                "Occurrences": {
                   "LineRanges": [
                       {
                           "Start": 1,
                           "End": 13
                       }
                   ]
               }
           },
           {
               "Count": 13826,
               "Type": "NameDetection",
               "Occurrences": {
                    "Records": [
                        {
                            "RecordIndex": 1,
                            "JsonPath": "$.ssn.value"
                        }
                    ]
                }
           },
           {
               "Count": 32,
               "Type": "AddressDetection"
```

```
                    }
                ],
                "TotalCount": 32
            }
        ],
        "CustomDataIdentifiers": {
            "Detections": [
                {
                    "Arn": "1712be25e7c7f53c731fe464f1c869b8",
                    "Name": "1712be25e7c7f53c731fe464f1c869b8",
                    "Count": 2,
                }
            ],
            "TotalCount": 2
        }
    }
}
```

Details

The Details field provides additional information about a single resource using the appropriate objects. Each resource must be provided in a separate resource object in the `Resources` object.

Note that if the finding size exceeds the maximum of 240 KB, then the `Details` object is removed from the finding. For control findings that use AWS Config rules, you can view the resource details on the AWS Config console.

Security Hub provides a set of available resource details for its supported resource types. These details correspond to values of the `Type` object. Use the provided types whenever possible.

For example, if the resource is an S3 bucket, then set the resource `Type` to `AwsS3Bucket` and provide the resource details in the `AwsS3Bucket` object.

The `Other` object allows you to provide custom fields and values. You use the `Other` object in the following cases:

- The resource type (the value of the resource `Type`) does not have a corresponding details object. To provide details for the resource, you use the `Other` object.
- The object for the resource type does not include all of the fields that you want to populate. In this case, use the details object for the resource type to populate the available fields. Use the `Other` object to populate the fields that are not in the type-specific object.

- The resource type is not one of the provided types. In this case, set `Resource.Type` to `Other`, and use the `Other` object to populate the details.

Example

```
"Details": {
  "AwsEc2Instance": {
    "IamInstanceProfileArn": "arn:aws:iam::123456789012:role/IamInstanceProfileArn",
    "ImageId": "ami-79fd7eee",
    "IpV4Addresses": ["1.1.1.1"],
    "IpV6Addresses": ["2001:db8:1234:1a2b::123"],
    "KeyName": "testkey",
    "LaunchedAt": "2018-09-29T01:25:54Z",
    "MetadataOptions": {
      "HttpEndpoint": "enabled",
      "HttpProtocolIpv6": "enabled",
      "HttpPutResponseHopLimit": 1,
      "HttpTokens": "optional",
      "InstanceMetadataTags": "disabled"
    },
    "NetworkInterfaces": [
    {
      "NetworkInterfaceId": "eni-e5aa89a3"
    }
    ],
    "SubnetId": "PublicSubnet",
    "Type": "i3.xlarge",
    "VirtualizationType": "hvm",
    "VpcId": "TestVPCIpv6"
  },
  "AwsS3Bucket": {
    "OwnerId": "da4d66eac431652a4d44d490a00500bded52c97d235b7b4752f9f688566fe6de",
    "OwnerName": "acmes3bucketowner"
  },
  "Other": { "LightPen": "blinky", "SerialNo": "1234abcd"}
}
```

Id

The identifier for the given resource type.

For AWS resources that are identified by Amazon Resource Names (ARNs), this is the ARN.

For AWS resources that lack ARNs, this is the identifier as defined by the AWS service that created the resource.

For non-AWS resources, this is a unique identifier that is associated with the resource.

Example

```
"Id": "arn:aws:s3:::amzn-s3-demo-bucket"
```

Partition

The partition in which the resource is located. A partition is a group of AWS Regions. Each AWS account is scoped to one partition.

The following partitions are supported:

- `aws` – AWS Regions
- `aws-cn` – China Regions
- `aws-us-gov` – AWS GovCloud (US) Region

Example

```
"Partition": "aws"
```

Region

The code for the AWS Region where this resource is located. For a list of Region codes, see Regional endpoints.

Example

```
"Region": "us-west-2"
```

ResourceRole

Identifies the role of the resource in the finding. A resource is either the target of the finding activity or the actor that performed the activity.

Example

```
"ResourceRole": "target"
```

Tags

This field provides tag key and value information for the resource involved in a finding. You can tag resources that are supported by the `GetResources` operation of the AWS Resource Groups Tagging API. Security Hub calls this operation though the service-linked role and retrieves the resource tags if the AWS Security Finding Format (ASFF) `Resource.Id` field is populated with the AWS resource ARN. Invalid resource IDs are ignored.

You can add resource tags to findings that Security Hub ingests, including findings from integrated AWS services and third-party products.

Adding tags tells you the tags that were associated with a resource at the time the finding was processed. You can include the `Tags` attribute only for resources that have an associated tag. If a resource has no associated tag, don't include a `Tags` attribute in the finding.

The inclusion of resource tags in findings eliminates the need to build data enrichment pipelines or manually enrich the metadata of security findings. You can also use tags to search or filter findings and insights and create automation rules.

For information about restrictions that apply to tags, see Tag naming limits and requirements.

You can only provide tags that exist on an AWS resource in this field. To provide data that isn't defined in the AWS Security Finding Format, use the `Other` details subfield.

Example

```
"Tags": {
    "billingCode": "Lotus-1-2-3",
    "needsPatching": "true"
}
```

Type

The type of resource that you are providing details for.

Whenever possible, use one of the provided resource types, such as `AwsEc2Instance` or `AwsS3Bucket`.

If the resource type does not match any of the provided resource types, then set the resource `Type` to `Other`, and use the `Other` details subfield to populate the details.

Supported values are listed under Resources.

Example

```
"Type": "AwsS3Bucket"
```

AwsAmazonMQ resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsAmazonMQ` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsAmazonMQBroker

`AwsAmazonMQBroker` provides information about an Amazon MQ broker, which is a message broker environment running on Amazon MQ.

The following example shows the ASFF for the `AwsAmazonMQBroker` object. To view descriptions of `AwsAmazonMQBroker` attributes, see AwsAmazonMQBroker in the *AWS Security Hub API Reference*.

Example

```
"AwsAmazonMQBroker": {
    "AutoMinorVersionUpgrade": true,
    "BrokerArn": "arn:aws:mq:us-east-1:123456789012:broker:TestBroker:b-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "BrokerId": "b-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "BrokerName": "TestBroker",
    "Configuration": {
        "Id": "c-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "Revision": 1
    },
    "DeploymentMode": "ACTIVE_STANDBY_MULTI_AZ",
    "EncryptionOptions": {
        "UseAwsOwnedKey": true
    },
    "EngineType": "ActiveMQ",
```

```
    "EngineVersion": "5.17.2",
    "HostInstanceType": "mq.t2.micro",
    "Logs": {
        "Audit": false,
        "AuditLogGroup": "/aws/amazonmq/broker/b-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/
audit",
        "General": false,
        "GeneralLogGroup": "/aws/amazonmq/broker/b-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111/general"
    },
    "MaintenanceWindowStartTime": {
        "DayOfWeek": "MONDAY",
        "TimeOfDay": "22:00",
        "TimeZone": "UTC"
    },
    "PubliclyAccessible": true,
    "SecurityGroups": [
        "sg-021345abcdef6789"
    ],
    "StorageType": "efs",
    "SubnetIds": [
        "subnet-1234567890abcdef0",
        "subnet-abcdef01234567890"
    ],
    "Users": [
        {
            "Username": "admin"
        }
    ]
}
```

AwsApiGateway resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsApiGateway` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsApiGatewayRestApi

The `AwsApiGatewayRestApi` object contains information about a REST API in version 1 of Amazon API Gateway.

The following is an example `AwsApiGatewayRestApi` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayRestApi` attributes, see AwsApiGatewayRestApiDetails in the *AWS Security Hub API Reference*.

Example

```
AwsApiGatewayRestApi: {
    "Id": "exampleapi",
    "Name": "Security Hub",
    "Description": "AWS Security Hub",
    "CreatedDate": "2018-11-18T10:20:05-08:00",
    "Version": "2018-10-26",
    "BinaryMediaTypes" : ["-'*~1*'"],
    "MinimumCompressionSize": 1024,
    "ApiKeySource": "AWS_ACCOUNT_ID",
    "EndpointConfiguration": {
        "Types": [
            "REGIONAL"
        ]
    }
}
```

AwsApiGatewayStage

The `AwsApiGatewayStage` object provides information about a version 1 Amazon API Gateway stage.

The following is an example `AwsApiGatewayStage` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayStage` attributes, see AwsApiGatewayStageDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsApiGatewayStage": {
    "DeploymentId": "n7hlmf",
    "ClientCertificateId": "a1b2c3",
    "StageName": "Prod",
    "Description" : "Stage Description",
    "CacheClusterEnabled": false,
    "CacheClusterSize" : "1.6",
    "CacheClusterStatus": "NOT_AVAILABLE",
    "MethodSettings": [
        {
```

```
            "MetricsEnabled": true,
            "LoggingLevel": "INFO",
            "DataTraceEnabled": false,
            "ThrottlingBurstLimit": 100,
            "ThrottlingRateLimit": 5.0,
            "CachingEnabled": false,
            "CacheTtlInSeconds": 300,
            "CacheDataEncrypted": false,
            "RequireAuthorizationForCacheControl": true,
            "UnauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER",
            "HttpMethod": "POST",
            "ResourcePath": "/echo"
        }
    ],
    "Variables": {"test": "value"},
    "DocumentationVersion": "2.0",
    "AccessLogSettings": {
        "Format": "{\"requestId\": \"$context.requestId\", \"extendedRequestId
\": \"$context.extendedRequestId\", \"ownerAccountId\": \"$context.accountId\",
 \"requestAccountId\": \"$context.identity.accountId\", \"callerPrincipal\":
 \"$context.identity.caller\", \"httpMethod\": \"$context.httpMethod\", \"resourcePath
\": \"$context.resourcePath\", \"status\": \"$context.status\", \"requestTime
\": \"$context.requestTime\", \"responseLatencyMs\": \"$context.responseLatency
\", \"errorMessage\": \"$context.error.message\", \"errorResponseType\":
 \"$context.error.responseType\", \"apiId\": \"$context.apiId\", \"awsEndpointRequestId
\": \"$context.awsEndpointRequestId\", \"domainName\": \"$context.domainName\", \"stage
\": \"$context.stage\", \"xrayTraceId\": \"$context.xrayTraceId\", \"sourceIp\":
 \"$context.identity.sourceIp\", \"user\": \"$context.identity.user\", \"userAgent
\": \"$context.identity.userAgent\", \"userArn\": \"$context.identity.userArn\",
 \"integrationLatency\": \"$context.integrationLatency\", \"integrationStatus
\": \"$context.integrationStatus\", \"authorizerIntegrationLatency\":
 \"$context.authorizer.integrationLatency\" }",
        "DestinationArn": "arn:aws:logs:us-west-2:111122223333:log-
group:SecurityHubAPIAccessLog/Prod"
    },
    "CanarySettings": {
        "PercentTraffic": 0.0,
        "DeploymentId": "ul73s8",
        "StageVariableOverrides" : [
            "String" : "String"
        ],
        "UseStageCache": false
    },
    "TracingEnabled": false,
```

```
    "CreatedDate": "2018-07-11T10:55:18-07:00",
    "LastUpdatedDate": "2020-08-26T11:51:04-07:00",
    "WebAclArn" : "arn:aws:waf-regional:us-west-2:111122223333:webacl/
cb606bd8-5b0b-4f0b-830a-dd304e48a822"
}
```

AwsApiGatewayV2Api

The `AwsApiGatewayV2Api` object contains information about a version 2 API in Amazon API Gateway.

The following is an example `AwsApiGatewayV2Api` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayV2Api` attributes, see AwsApiGatewayV2ApiDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsApiGatewayV2Api": {
    "ApiEndpoint": "https://example.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-03-28T00:32:37Z",
   "Description": "ApiGatewayV2 Api",
   "Version": "string",
    "Name": "my-api",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
   "CorsConfiguration": {
        "AllowOrigins": [ "*" ],
        "AllowCredentials": true,
        "ExposeHeaders": [ "string" ],
        "MaxAge": 3000,
        "AllowMethods": [
          "GET",
          "PUT",
          "POST",
          "DELETE",
          "HEAD"
        ],
        "AllowHeaders": [ "*" ]
    }
}
```

AwsApiGatewayV2Stage

`AwsApiGatewayV2Stage` contains information about a version 2 stage for Amazon API Gateway.

The following is an example `AwsApiGatewayV2Stage` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayV2Stage` attributes, see AwsApiGatewayV2StageDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsApiGatewayV2Stage": {
    "CreatedDate": "2020-04-08T00:36:05Z",
    "Description" : "ApiGatewayV2",
    "DefaultRouteSettings": {
        "DetailedMetricsEnabled": false,
        "LoggingLevel": "INFO",
        "DataTraceEnabled": true,
        "ThrottlingBurstLimit": 100,
        "ThrottlingRateLimit": 50
    },
    "DeploymentId": "x1zwyv",
    "LastUpdatedDate": "2020-04-08T00:36:13Z",
    "RouteSettings": {
        "DetailedMetricsEnabled": false,
        "LoggingLevel": "INFO",
        "DataTraceEnabled": true,
        "ThrottlingBurstLimit": 100,
        "ThrottlingRateLimit": 50
    },
    "StageName": "prod",
    "StageVariables": [
        "function": "my-prod-function"
    ],
    "AccessLogSettings": {
        "Format": "{\"requestId\": \"$context.requestId\", \"extendedRequestId
\": \"$context.extendedRequestId\", \"ownerAccountId\": \"$context.accountId\",
 \"requestAccountId\": \"$context.identity.accountId\", \"callerPrincipal\":
 \"$context.identity.caller\", \"httpMethod\": \"$context.httpMethod\", \"resourcePath
\": \"$context.resourcePath\", \"status\": \"$context.status\", \"requestTime
\": \"$context.requestTime\", \"responseLatencyMs\": \"$context.responseLatency
\", \"errorMessage\": \"$context.error.message\", \"errorResponseType\":
 \"$context.error.responseType\", \"apiId\": \"$context.apiId\", \"awsEndpointRequestId
\": \"$context.awsEndpointRequestId\", \"domainName\": \"$context.domainName\", \"stage
\": \"$context.stage\", \"xrayTraceId\": \"$context.xrayTraceId\", \"sourceIp\":
```

```
 \"$context.identity.sourceIp\", \"user\": \"$context.identity.user\", \"userAgent
\": \"$context.identity.userAgent\", \"userArn\": \"$context.identity.userArn\",
 \"integrationLatency\": \"$context.integrationLatency\", \"integrationStatus
\": \"$context.integrationStatus\", \"authorizerIntegrationLatency\":
 \"$context.authorizer.integrationLatency\" }",
        "DestinationArn": "arn:aws:logs:us-west-2:111122223333:log-
group:SecurityHubAPIAccessLog/Prod"
    },
    "AutoDeploy": false,
    "LastDeploymentStatusMessage": "Message",
    "ApiGatewayManaged": true,
}
```

AwsAppSync resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for AwsAppSync resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsAppSyncGraphQLApi

AwsAppSyncGraphQLApi provides information about an AWS AppSync GraphQL API, which is a top-level construct for your application.

The following example shows the ASFF for the AwsAppSyncGraphQLApi object. To view descriptions of AwsAppSyncGraphQLApi attributes, see AwsAppSyncGraphQLApi in the *AWS Security Hub API Reference*.

Example

```
"AwsAppSyncGraphQLApi": {
    "AdditionalAuthenticationProviders": [
    {
     "AuthenticationType": "AWS_LAMBDA",
     "LambdaAuthorizerConfig": {
      "AuthorizerResultTtlInSeconds": 300,
      "AuthorizerUri": "arn:aws:lambda:us-east-1:123456789012:function:mylambdafunc"
     }
    },
    {
     "AuthenticationType": "AWS_IAM"
```

```
    }
    ],
    "ApiId": "021345abcdef6789",
    "Arn": "arn:aws:appsync:eu-central-1:123456789012:apis/021345abcdef6789",
    "AuthenticationType": "API_KEY",
    "Id": "021345abcdef6789",
    "LogConfig": {
     "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/appsync-
graphqlapi-logs-eu-central-1",
     "ExcludeVerboseContent": true,
     "FieldLogLevel": "ALL"
    },
    "Name": "My AppSync App",
    "XrayEnabled": true,
}
```

AwsAthena resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsAthena` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsAthenaWorkGroup

`AwsAthenaWorkGroup` provides information about an Amazon Athena workgroup. A workgroup helps you separate users, teams, applications, or workloads. It also helps you set limits on data processing and track costs.

The following example shows the ASFF for the `AwsAthenaWorkGroup` object. To view descriptions of `AwsAthenaWorkGroup` attributes, see AwsAthenaWorkGroup in the *AWS Security Hub API Reference*.

Example

```
"AwsAthenaWorkGroup": {
    "Description": "My workgroup for prod workloads",
    "Name": "MyWorkgroup",
    "WorkgroupConfiguration" {
        "ResultConfiguration": {
            "EncryptionConfiguration": {
                "EncryptionOption": "SSE_KMS",
```

```
                "KmsKey": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
            }
        }
    },
        "State": "ENABLED"
}
```

AwsAutoScaling resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsAutoScaling` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsAutoScalingAutoScalingGroup

The `AwsAutoScalingAutoScalingGroup` object provides details about an automatic scaling group.

The following is an example `AwsAutoScalingAutoScalingGroup` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsAutoScalingAutoScalingGroup` attributes, see AwsAutoScalingAutoScalingGroupDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsAutoScalingAutoScalingGroup": {
        "CreatedTime": "2017-10-17T14:47:11Z",
        "HealthCheckGracePeriod": 300,
        "HealthCheckType": "EC2",
        "LaunchConfigurationName": "mylaunchconf",
        "LoadBalancerNames": [],
        "LaunchTemplate": {
            "LaunchTemplateId": "string",
            "LaunchTemplateName": "string",
            "Version": "string"
        },
        "MixedInstancesPolicy": {
            "InstancesDistribution": {
                "OnDemandAllocationStrategy": "prioritized",
                "OnDemandBaseCapacity": number,
```

```
                "OnDemandPercentageAboveBaseCapacity": number,
                "SpotAllocationStrategy": "lowest-price",
                "SpotInstancePools": number,
                "SpotMaxPrice": "string"
            },
            "LaunchTemplate": {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "string",
                    "LaunchTemplateName": "string",
                    "Version": "string"
                 },
                "CapacityRebalance": true,
                "Overrides": [
                    {
                       "InstanceType": "string",
                       "WeightedCapacity": "string"
                    }
                ]
            }
        }
    }
}
```

AwsAutoScalingLaunchConfiguration

The `AwsAutoScalingLaunchConfiguration` object provides details about a launch configuration.

The following is an example `AwsAutoScalingLaunchConfiguration` finding in the AWS Security Finding Format (ASFF).

To view descriptions of `AwsAutoScalingLaunchConfiguration` attributes, see AwsAutoScalingLaunchConfigurationDetails in the *AWS Security Hub API Reference*.

Example

```
AwsAutoScalingLaunchConfiguration: {
    "LaunchConfigurationName": "newtest",
    "ImageId": "ami-058a3739b02263842",
    "KeyName": "55hundredinstance",
    "SecurityGroups": [ "sg-01fce87ad6e019725" ],
    "ClassicLinkVpcSecurityGroups": [],
    "UserData": "...Base64-Encoded user data..."
```

```
    "InstanceType": "a1.metal",
    "KernelId": "",
    "RamdiskId": "ari-a51cf9cc",
    "BlockDeviceMappings": [
        {
            "DeviceName": "/dev/sdh",
            "Ebs": {
                "VolumeSize": 30,
                "VolumeType": "gp2",
                "DeleteOnTermination": false,
                "Encrypted": true,
                "SnapshotId": "snap-ffaa1e69",
                "VirtualName": "ephemeral1"
            }
        },
        {
            "DeviceName": "/dev/sdb",
            "NoDevice": true
        },
        {
            "DeviceName": "/dev/sda1",
            "Ebs": {
                "SnapshotId": "snap-02420cd3d2dea1bc0",
                "VolumeSize": 8,
                "VolumeType": "gp2",
                "DeleteOnTermination": true,
                "Encrypted": false
            }
        },
        {
            "DeviceName": "/dev/sdi",
            "Ebs": {
                "VolumeSize": 20,
                "VolumeType": "gp2",
                "DeleteOnTermination": false,
                "Encrypted": true
            }
        },
        {
            "DeviceName": "/dev/sdc",
            "NoDevice": true
        }
    ],
    "InstanceMonitoring": {
```

```
        "Enabled": false
    },
    "CreatedTime": 1620842933453,
    "EbsOptimized": false,
    "AssociatePublicIpAddress": true,
    "SpotPrice": "0.045"
}
```

AwsBackup resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsBackup` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsBackupBackupPlan

The `AwsBackupBackupPlan` object provides information about an AWS Backup backup plan. An AWS Backup backup plan is a policy expression that defines when and how you want to back up your AWS resources.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsBackupBackupPlan` object. To view descriptions of `AwsBackupBackupPlan` attributes, see AwsBackupBackupPlan in the *AWS Security Hub API Reference*.

Example

```
"AwsBackupBackupPlan": {
    "BackupPlan": {
     "AdvancedBackupSettings": [{
      "BackupOptions": {
       "WindowsVSS":"enabled"
      },
      "ResourceType":"EC2"
     }],
     "BackupPlanName": "test",
     "BackupPlanRule": [{
      "CompletionWindowMinutes": 10080,
      "CopyActions": [{
       "DestinationBackupVaultArn": "arn:aws:backup:us-east-1:858726136373:backup-
vault:aws/efs/automatic-backup-vault",
```

```
       "Lifecycle": {
        "DeleteAfterDays": 365,
        "MoveToColdStorageAfterDays": 30
       }
      }],
      "Lifecycle": {
       "DeleteAfterDays": 35
      },
      "RuleName": "DailyBackups",
      "ScheduleExpression": "cron(0 5 ? * * *)",
      "StartWindowMinutes": 480,
      "TargetBackupVault": "Default"
      },
      {
      "CompletionWindowMinutes": 10080,
      "CopyActions": [{
       "DestinationBackupVaultArn": "arn:aws:backup:us-east-1:858726136373:backup-
vault:aws/efs/automatic-backup-vault",
       "Lifecycle": {
        "DeleteAfterDays": 365,
        "MoveToColdStorageAfterDays": 30
       }
      }],
      "Lifecycle": {
       "DeleteAfterDays": 35
      },
      "RuleName": "Monthly",
      "ScheduleExpression": "cron(0 5 1 * ? *)",
      "StartWindowMinutes": 480,
      "TargetBackupVault": "Default"
     }]
    },
    "BackupPlanArn": "arn:aws:backup:us-east-1:858726136373:backup-
plan:b6d6b896-590d-4ee1-bf29-c5ccae63f4e7",
    "BackupPlanId": "b6d6b896-590d-4ee1-bf29-c5ccae63f4e7",
    "VersionId": "ZDVjNDIzMjItYTZiNS00NzczLTg4YzctNmExMWM2NjZhY2E1"
}
```


AwsBackupBackupVault

The `AwsBackupBackupVault` object provides information about an AWS Backup backup vault. A AWS Backup backup vault is a container that stores and organizes your backups.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsBackupBackupVault` object. To view descriptions of `AwsBackupBackupVault` attributes, see AwsBackupBackupVault in the *AWS Security Hub API Reference*.

Example

```
"AwsBackupBackupVault": {
    "AccessPolicy": {
     "Statement": [{
      "Action": [
       "backup:DeleteBackupVault",
       "backup:DeleteBackupVaultAccessPolicy",
       "backup:DeleteRecoveryPoint",
       "backup:StartCopyJob",
       "backup:StartRestoreJob",
       "backup:UpdateRecoveryPointLifecycle"
      ],
      "Effect": "Deny",
      "Principal": {
       "AWS": "*"
      },
      "Resource": "*"
     }],
     "Version": "2012-10-17"
    },
    "BackupVaultArn": "arn:aws:backup:us-east-1:123456789012:backup-vault:aws/efs/
automatic-backup-vault",
    "BackupVaultName": "aws/efs/automatic-backup-vault",
    "EncrytionKeyArn": "arn:aws:kms:us-east-1:444455556666:key/72ba68d4-5e43-40b0-
ba38-838bf8d06ca0",
    "Notifications": {
     "BackupVaultEvents": ["BACKUP_JOB_STARTED", "BACKUP_JOB_COMPLETED",
 "COPY_JOB_STARTED"],
     "SNSTopicArn": "arn:aws:sns:us-west-2:111122223333:MyVaultTopic"
    }
}
```

AwsBackupRecoveryPoint

The `AwsBackupRecoveryPoint` object provides information about an AWS Backup backup, also referred to as a recovery point. An AWS Backup recovery point represents the content of a resource at a specified time.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsBackupRecoveryPoint` object. To view descriptions of `AwsBackupBackupVault` attributes, see AwsBackupRecoveryPoint in the *AWS Security Hub API Reference*.

Example

```
"AwsBackupRecoveryPoint": {
    "BackupSizeInBytes": 0,
    "BackupVaultName": "aws/efs/automatic-backup-vault",
    "BackupVaultArn": "arn:aws:backup:us-east-1:111122223333:backup-vault:aws/efs/
automatic-backup-vault",
    "CalculatedLifecycle": {
     "DeleteAt": "2021-08-30T06:51:58.271Z",
     "MoveToColdStorageAt": "2020-08-10T06:51:58.271Z"
    },
    "CompletionDate": "2021-07-26T07:21:40.361Z",
    "CreatedBy": {
     "BackupPlanArn": "arn:aws:backup:us-east-1:111122223333:backup-plan:aws/
efs/73d922fb-9312-3a70-99c3-e69367f9fdad",
     "BackupPlanId": "aws/efs/73d922fb-9312-3a70-99c3-e69367f9fdad",
     "BackupPlanVersion": "ZGM4YzY5YjktMWYxNC00ZTBmLWE5MjYtZmU5OWNiZmM5ZjIz",
     "BackupRuleId": "2a600c2-42ad-4196-808e-084923ebfd25"
    },
    "CreationDate": "2021-07-26T06:51:58.271Z",
    "EncryptionKeyArn": "arn:aws:kms:us-east-1:111122223333:key/72ba68d4-5e43-40b0-
ba38-838bf8d06ca0",
    "IamRoleArn": "arn:aws:iam::111122223333:role/aws-service-role/
backup.amazonaws.com/AWSServiceRoleForBackup",
    "IsEncrypted": true,
    "LastRestoreTime": "2021-07-26T06:51:58.271Z",
    "Lifecycle": {
     "DeleteAfterDays": 35,
     "MoveToColdStorageAfterDays": 15
    },
    "RecoveryPointArn": "arn:aws:backup:us-east-1:111122223333:recovery-point:151a59e4-
f1d5-4587-a7fd-0774c6e91268",
    "ResourceArn": "arn:aws:elasticfilesystem:us-east-1:858726136373:file-system/
fs-15bd31a1",
    "ResourceType": "EFS",
    "SourceBackupVaultArn": "arn:aws:backup:us-east-1:111122223333:backup-vault:aws/
efs/automatic-backup-vault",
    "Status": "COMPLETED",
    "StatusMessage": "Failure message",
```

```
    "StorageClass": "WARM"
}
```

AwsCertificateManager resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsCertificateManager` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsCertificateManagerCertificate

The `AwsCertificateManagerCertificate` object provides details about an AWS Certificate Manager (ACM) certificate.

The following is an example `AwsCertificateManagerCertificate` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsCertificateManagerCertificate` attributes, see AwsCertificateManagerCertificateDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsCertificateManagerCertificate": {
    "CertificateAuthorityArn": "arn:aws:acm:us-west-2:444455556666:certificate-
authority/example",
    "CreatedAt": "2019-05-24T18:12:02.000Z",
    "DomainName": "example.amazondomains.com",
    "DomainValidationOptions": [
        {
            "DomainName": "example.amazondomains.com",
            "ResourceRecord": {
                "Name": "_1bacb61828d3a1020c40a560ceed08f7.example.amazondomains.com",
                "Type": "CNAME",
                "Value": "_example.acm-validations.aws."
             },
             "ValidationDomain": "example.amazondomains.com",
             "ValidationEmails": [sample_email@sample.com],
             "ValidationMethod": "DNS",
             "ValidationStatus": "SUCCESS"
        }
    ],
    "ExtendedKeyUsages": [
        {
```

```
            "Name": "TLS_WEB_SERVER_AUTHENTICATION",
            "OId": "1.3.6.1.5.5.7.3.1"
        },
        {
            "Name": "TLS_WEB_CLIENT_AUTHENTICATION",
            "OId": "1.3.6.1.5.5.7.3.2"
        }
    ],
    "FailureReason": "",
    "ImportedAt": "2018-08-17T00:13:00.000Z",
    "InUseBy": ["arn:aws:amazondomains:us-west-2:444455556666:loadbalancer/example"],
    "IssuedAt": "2020-04-26T00:41:17.000Z",
    "Issuer": "Amazon",
    "KeyAlgorithm": "RSA-1024",
    "KeyUsages": [
        {
            "Name": "DIGITAL_SIGNATURE",
        },
        {
            "Name": "KEY_ENCIPHERMENT",
        }
    ],
    "NotAfter": "2021-05-26T12:00:00.000Z",
    "NotBefore": "2020-04-26T00:00:00.000Z",
    "Options": {
        "CertificateTransparencyLoggingPreference": "ENABLED",
    }
    "RenewalEligibility": "ELIGIBLE",
    "RenewalSummary": {
        "DomainValidationOptions": [
            {
                "DomainName": "example.amazondomains.com",
                "ResourceRecord": {
                    "Name":
"_1bacb61828d3a1020c40a560ceed08f7.example.amazondomains.com",
                    "Type": "CNAME",
                    "Value": "_example.acm-validations.aws.com",
                },
                "ValidationDomain": "example.amazondomains.com",
                "ValidationEmails": ["sample_email@sample.com"],
                "ValidationMethod": "DNS",
                "ValidationStatus": "SUCCESS"
            }
        ],
```

```
        "RenewalStatus": "SUCCESS",
        "RenewalStatusReason": "",
        "UpdatedAt": "2020-04-26T00:41:35.000Z",
    },
    "Serial": "02:ac:86:b6:07:2f:0a:61:0e:3a:ac:fd:d9:ab:17:1a",
    "SignatureAlgorithm": "SHA256WITHRSA",
    "Status": "ISSUED",
    "Subject": "CN=example.amazondomains.com",
    "SubjectAlternativeNames": ["example.amazondomains.com"],
    "Type": "AMAZON_ISSUED"
}
```

AwsCloudFormation resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsCloudFormation` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsCloudFormationStack

The `AwsCloudFormationStack` object provides details about an AWS CloudFormation stack that is nested as a resource in a top-level template.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsCloudFormationStack` object. To view descriptions of `AwsCloudFormationStack` attributes, see AwsCloudFormationStackDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsCloudFormationStack": {
 "Capabilities": [
  "CAPABILITY_IAM",
  "CAPABILITY_NAMED_IAM"
 ],
 "CreationTime": "2022-02-18T15:31:53.161Z",
 "Description": "AWS CloudFormation Sample",
 "DisableRollback": true,
 "DriftInformation": {
  "StackDriftStatus": "DRIFTED"
 },
 "EnableTerminationProtection": false,
```

```
 "LastUpdatedTime": "2022-02-18T15:31:53.161Z",
 "NotificationArns": [
  "arn:aws:sns:us-east-1:978084797471:sample-sns-cfn"
 ],
 "Outputs": [{
  "Description": "URL for newly created LAMP stack",
  "OutputKey": "WebsiteUrl",
  "OutputValue": "http://ec2-44-193-18-241.compute-1.amazonaws.com"
 }],
 "RoleArn": "arn:aws:iam::012345678910:role/exampleRole",
 "StackId": "arn:aws:cloudformation:us-east-1:978084797471:stack/sample-stack/
e5d9f7e0-90cf-11ec-88c6-12ac1f91724b",
 "StackName": "sample-stack",
 "StackStatus": "CREATE_COMPLETE",
 "StackStatusReason": "Success",
 "TimeoutInMinutes": 1
}
```

AwsCloudFront resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsCloudFront` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsCloudFrontDistribution

The `AwsCloudFrontDistribution` object provides details about a Amazon CloudFront distribution configuration.

The following is an example `AwsCloudFrontDistribution` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsCloudFrontDistribution` attributes, see AwsCloudFrontDistributionDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsCloudFrontDistribution": {
    "CacheBehaviors": {
        "Items": [
            {
               "ViewerProtocolPolicy": "https-only"
            }
```

```
        ]
    },
    "DefaultCacheBehavior": {
        "ViewerProtocolPolicy": "https-only"
    },
    "DefaultRootObject": "index.html",
    "DomainName": "d2wkuj2w9l34gt.cloudfront.net",
    "Etag": "E37HOT42DHPVYH",
    "LastModifiedTime": "2015-08-31T21:11:29.093Z",
    "Logging": {
        "Bucket": "myawslogbucket.s3.amazonaws.com",
        "Enabled": false,
        "IncludeCookies": false,
        "Prefix": "myawslog/"
    },
    "OriginGroups": {
         "Items": [
             {
                "FailoverCriteria": {
                    "StatusCodes": {
                        "Items": [
                            200,
                            301,
                            404
                        ]
                        "Quantity": 3
                    }
                }
             }
          ]
    },
    "Origins": {
          "Items": [
              {
                 "CustomOriginConfig": {
                     "HttpPort": 80,
                     "HttpsPort": 443,
                     "OriginKeepaliveTimeout": 60,
                     "OriginProtocolPolicy": "match-viewer",
                     "OriginReadTimeout": 30,
                     "OriginSslProtocols": {
                       "Items": ["SSLv3", "TLSv1"],
                       "Quantity": 2
                     }
```

```
                }
            },
        ]
    },
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "Id": "my-origin",
                "OriginPath": "/production",
                "S3OriginConfig": {
                    "OriginAccessIdentity": "origin-access-identity/cloudfront/
E2YFS67H6VB6E4"
                }
        ]
    },
    "Status": "Deployed",
    "ViewerCertificate": {
           "AcmCertificateArn": "arn:aws:acm::123456789012:AcmCertificateArn",
           "Certificate": "ASCAJRRE5XYF52TKRY5M4",
           "CertificateSource": "iam",
           "CloudFrontDefaultCertificate": true,
           "IamCertificateId": "ASCAJRRE5XYF52TKRY5M4",
           "MinimumProtocolVersion": "TLSv1.2_2021",
           "SslSupportMethod": "sni-only"
     },
     "WebAclId": "waf-1234567890"
}
```

AwsCloudTrail resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsCloudTrail` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsCloudTrailTrail

The `AwsCloudTrailTrail` object provides details about a AWS CloudTrail trail.

The following is an example `AwsCloudTrailTrail` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsCloudTrailTrail` attributes, see AwsCloudTrailTrailDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsCloudTrailTrail": {
    "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-
group:CloudTrail/regression:*",
    "CloudWatchLogsRoleArn": "arn:aws:iam::866482105055:role/
CloudTrail_CloudWatchLogs",
    "HasCustomEventSelectors": true,
    "HomeRegion": "us-west-2",
    "IncludeGlobalServiceEvents": true,
    "IsMultiRegionTrail": true,
    "IsOrganizationTrail": false,
    "KmsKeyId": "kmsKeyId",
    "LogFileValidationEnabled": true,
    "Name": "regression-trail",
    "S3BucketName": "cloudtrail-bucket",
    "S3KeyPrefix": "s3KeyPrefix",
    "SnsTopicArn": "arn:aws:sns:us-east-2:123456789012:MyTopic",
    "SnsTopicName": "snsTopicName",
    "TrailArn": "arn:aws:cloudtrail:us-west-2:123456789012:trail"
}
```

AwsCloudWatch resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsCloudWatch` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsCloudWatchAlarm

The `AwsCloudWatchAlarm` object provides details about Amazon CloudWatch alarms that watch a metric or perform an action when an alarm changes state.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsCloudWatchAlarm` object. To view descriptions of `AwsCloudWatchAlarm` attributes, see AwsCloudWatchAlarmDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsCloudWatchAlarm": {
 "ActionsEnabled": true,
 "AlarmActions": [
```

```
   "arn:aws:automate:region:ec2:stop",
   "arn:aws:automate:region:ec2:terminate"
  ],
  "AlarmArn": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:sampleAlarm",
  "AlarmConfigurationUpdatedTimestamp": "2022-02-18T15:31:53.161Z",
  "AlarmDescription": "Alarm Example",
  "AlarmName": "Example",
  "ComparisonOperator": "GreaterThanOrEqualToThreshold",
  "DatapointsToAlarm": 1,
  "Dimensions": [{
   "Name": "InstanceId",
   "Value": "i-1234567890abcdef0"
  }],
  "EvaluateLowSampleCountPercentile": "evaluate",
  "EvaluationPeriods": 1,
  "ExtendedStatistic": "p99.9",
  "InsufficientDataActions": [
   "arn:aws:automate:region:ec2:stop"
  ],
  "MetricName": "Sample Metric",
  "Namespace": "YourNamespace",
  "OkActions": [
   "arn:aws:swf:region:account-id:action/actions/AWS_EC2.InstanceId.Stop/1.0"
  ],
  "Period": 1,
  "Statistic": "SampleCount",
  "Threshold": 12.3,
  "ThresholdMetricId": "t1",
  "TreatMissingData": "notBreaching",
  "Unit": "Kilobytes/Second"
}
```

AwsCodeBuild resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsCodeBuild` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsCodeBuildProject

The `AwsCodeBuildProject` object provides information about an AWS CodeBuild project.

The following is an example `AwsCodeBuildProject` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsCodeBuildProject` attributes, see AwsCodeBuildProjectDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsCodeBuildProject": {
   "Artifacts": [
      {
          "ArtifactIdentifier": "string",
          "EncryptionDisabled": boolean,
          "Location": "string",
          "Name": "string",
          "NamespaceType": "string",
          "OverrideArtifactName": boolean,
          "Packaging": "string",
          "Path": "string",
          "Type": "string"
       }
   ],
   "SecondaryArtifacts": [
      {
          "ArtifactIdentifier": "string",
          "EncryptionDisabled": boolean,
          "Location": "string",
          "Name": "string",
          "NamespaceType": "string",
          "OverrideArtifactName": boolean,
          "Packaging": "string",
          "Path": "string",
          "Type": "string"
       }
   ],
   "EncryptionKey": "string",
   "Certificate": "string",
   "Environment": {
      "Certificate": "string",
      "EnvironmentVariables": [
           {
                "Name": "string",
                "Type": "string",
                "Value": "string"
           }
```

```
      ],
   "ImagePullCredentialsType": "string",
   "PrivilegedMode": boolean,
   "RegistryCredential": {
       "Credential": "string",
       "CredentialProvider": "string"
   },
   "Type": "string"
   },
   "LogsConfig": {
        "CloudWatchLogs": {
             "GroupName": "string",
             "Status": "string",
             "StreamName": "string"
        },
        "S3Logs": {
             "EncryptionDisabled": boolean,
             "Location": "string",
             "Status": "string"
        }
   },
   "Name": "string",
   "ServiceRole": "string",
   "Source": {
        "Type": "string",
        "Location": "string",
        "GitCloneDepth": integer
   },
   "VpcConfig": {
        "VpcId": "string",
        "Subnets": ["string"],
        "SecurityGroupIds": ["string"]
   }
}
```

AwsDms resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsDms` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see [AWS Security Finding Format (ASFF)].

AwsDmsEndpoint

The `AwsDmsEndpoint` object provides information about an AWS Database Migration Service (AWS DMS) endpoint. An endpoint provides connection, data store type, and location information about your data store.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsDmsEndpoint` object. To view descriptions of `AwsDmsEndpoint` attributes, see AwsDmsEndpointDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsDmsEndpoint": {
    "CertificateArn": "arn:aws:dms:us-
east-1:123456789012:cert:EXAMPLEIGDURVZGVJQZDPWJ5A7F2YDJVSMTBWFI",
    "DatabaseName": "Test",
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EXAMPLEQB3CZY33F7XV253NAJVBNPK6MJQVFVQA",
    "EndpointIdentifier": "target-db",
    "EndpointType": "TARGET",
    "EngineName": "mariadb",
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
    "Port": 3306,
    "ServerName": "target-db.exampletafyu.us-east-1.rds.amazonaws.com",
    "SslMode": "verify-ca",
    "Username": "admin"
}
```

AwsDmsReplicationInstance

The `AwsDmsReplicationInstance` object provides information about an AWS Database Migration Service (AWS DMS) replication instance. DMS uses a replication instance to connect to your source data store, read the source data, and format the data for consumption by the target data store.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsDmsReplicationInstance` object. To view descriptions of `AwsDmsReplicationInstance` attributes, see AwsDmsReplicationInstanceDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsDmsReplicationInstance": {
    "AllocatedStorage": 50,
    "AutoMinorVersionUpgrade": true,
    "AvailabilityZone": "us-east-1b",
    "EngineVersion": "3.5.1",
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
    "MultiAZ": false,
    "PreferredMaintenanceWindow": "wed:08:08-wed:08:38",
    "PubliclyAccessible": true,
    "ReplicationInstanceClass": "dms.c5.xlarge",
    "ReplicationInstanceIdentifier": "second-replication-instance",
    "ReplicationSubnetGroup": {
        "ReplicationSubnetGroupIdentifier": "default-vpc-2344f44f"
    },
    "VpcSecurityGroups": [
        {
            "VpcSecurityGroupId": "sg-003a34e205138138b"
        }
    ]
}
```

AwsDmsReplicationTask

The `AwsDmsReplicationTask` object provides information about an AWS Database Migration Service (AWS DMS) replication task. A replication task moves a set of data from the source endpoint to the target endpoint.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsDmsReplicationInstance` object. To view descriptions of `AwsDmsReplicationInstance` attributes, see AwsDmsReplicationInstance in the *AWS Security Hub API Reference*.

Example

```
"AwsDmsReplicationTask": {
    "CdcStartPosition": "2023-08-28T14:26:22",
    "Id": "arn:aws:dms:us-
east-1:123456789012:task:YDYUOHZIXWKQSUCBMUCQCNY44SJW74VJNB5DFWQ",
    "MigrationType": "cdc",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T7V6RFDP23PYQWUL26N3PF5REKML4YOUGIMYJUI",
    "ReplicationTaskIdentifier": "test-task",
```

```
    "ReplicationTaskSettings": "{\"Logging\":{\"EnableLogging\":false,
\"EnableLogContext\":false,\"LogComponents\":[{\"Severity\":\"LOGGER_SEVERITY_DEFAULT
\",\"Id\":\"TRANSFORMATION\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",
\"Id\":\"SOURCE_UNLOAD\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":
\"IO\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"TARGET_LOAD\"},
{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"PERFORMANCE\"},{\"Severity
\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"SOURCE_CAPTURE\"},{\"Severity\":
\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"SORTER\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT
\",\"Id\":\"REST_SERVER\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id
\":\"VALIDATOR_EXT\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":
\"TARGET_APPLY\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"TASK_MANAGER
\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"TABLES_MANAGER\"},
{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"METADATA_MANAGER\"},
{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"FILE_FACTORY\"},{\"Severity\":
\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"COMMON\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT
\",\"Id\":\"ADDONS\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"DATA_STRUCTURE
\"},{\"Severity\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"COMMUNICATION\"},{\"Severity
\":\"LOGGER_SEVERITY_DEFAULT\",\"Id\":\"FILE_TRANSFER\"}],\"CloudWatchLogGroup
\":null,\"CloudWatchLogStream\":null},\"StreamBufferSettings\":{\"StreamBufferCount
\":3,\"CtrlStreamBufferSizeInMB\":5,\"StreamBufferSizeInMB\":8},\"ErrorBehavior
\":{\"FailOnNoTablesCaptured\":true,\"ApplyErrorUpdatePolicy\":\"LOG_ERROR\",
\"FailOnTransactionConsistencyBreached\":false,\"RecoverableErrorThrottlingMax\":1800,
\"DataErrorEscalationPolicy\":\"SUSPEND_TABLE\",\"ApplyErrorEscalationCount\":0,
\"RecoverableErrorStopRetryAfterThrottlingMax\":true,\"RecoverableErrorThrottling
\":true,\"ApplyErrorFailOnTruncationDdl\":false,\"DataTruncationErrorPolicy\":
\"LOG_ERROR\",\"ApplyErrorInsertPolicy\":\"LOG_ERROR\",\"EventErrorPolicy\":
\"IGNORE\",\"ApplyErrorEscalationPolicy\":\"LOG_ERROR\",\"RecoverableErrorCount
\":-1,\"DataErrorEscalationCount\":0,\"TableErrorEscalationPolicy\":\"STOP_TASK
\",\"RecoverableErrorInterval\":5,\"ApplyErrorDeletePolicy\":\"IGNORE_RECORD\",
\"TableErrorEscalationCount\":0,\"FullLoadIgnoreConflicts\":true,\"DataErrorPolicy
\":\"LOG_ERROR\",\"TableErrorPolicy\":\"SUSPEND_TABLE\"},\"TTSettings
\":{\"TTS3Settings\":null,\"TTRecordSettings\":null,\"EnableTT\":false},
\"FullLoadSettings\":{\"CommitRate\":10000,\"StopTaskCachedChangesApplied
\":false,\"StopTaskCachedChangesNotApplied\":false,\"MaxFullLoadSubTasks
\":8,\"TransactionConsistencyTimeout\":600,\"CreatePkAfterFullLoad\":false,
\"TargetTablePrepMode\":\"DO_NOTHING\"},\"TargetMetadata\":{\"ParallelApplyBufferSize
\":0,\"ParallelApplyQueuesPerThread\":0,\"ParallelApplyThreads\":0,\"TargetSchema
\":\"\",\"InlineLobMaxSize\":0,\"ParallelLoadQueuesPerThread\":0,\"SupportLobs
\":true,\"LobChunkSize\":64,\"TaskRecoveryTableEnabled\":false,\"ParallelLoadThreads
\":0,\"LobMaxSize\":0,\"BatchApplyEnabled\":false,\"FullLobMode\":true,
\"LimitedSizeLobMode\":false,\"LoadMaxFileSize\":0,\"ParallelLoadBufferSize\":0},
\"BeforeImageSettings\":null,\"ControlTablesSettings\":{\"historyTimeslotInMinutes
\":5,\"HistoryTimeslotInMinutes\":5,\"StatusTableEnabled\":false,
\"SuspendedTablesTableEnabled\":false,\"HistoryTableEnabled\":false,\"ControlSchema
```

```
\":\"\",\"FullLoadExceptionTableEnabled\":false},\"LoopbackPreventionSettings
\":null,\"CharacterSetSettings\":null,\"FailTaskWhenCleanTaskResourceFailed
\":false,\"ChangeProcessingTuning\":{\"StatementCacheSize\":50,\"CommitTimeout
\":1,\"BatchApplyPreserveTransaction\":true,\"BatchApplyTimeoutMin\":1,
\"BatchSplitSize\":0,\"BatchApplyTimeoutMax\":30,\"MinTransactionSize\":1000,
\"MemoryKeepTime\":60,\"BatchApplyMemoryLimit\":500,\"MemoryLimitTotal\":1024},
\"ChangeProcessingDdlHandlingPolicy\":{\"HandleSourceTableDropped\":true,
\"HandleSourceTableTruncated\":true,\"HandleSourceTableAltered\":true},
\"PostProcessingRules\":null}",
    "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:TZPWV2VCXEGHYOKVKRNHAKJ4Q3RUXACNGFGYWRI",
    "TableMappings": "{\"rules\":[{\"rule-type\":\"selection\",\"rule-id\":
\"969761702\",\"rule-name\":\"969761702\",\"object-locator\":{\"schema-name\":\"%table
\",\"table-name\":\"%example\"},\"rule-action\":\"exclude\",\"filters\":[]}]}",
    "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:ABR8LBOQB3CZY33F7XV253NAJVBNPK6MJQVFVQA"
}
```


AwsDynamoDB resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for AwsDynamoDB resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsDynamoDbTable

The AwsDynamoDbTable object provides details about an Amazon DynamoDB table.

The following is an example AwsDynamoDbTable finding in the AWS Security Finding Format (ASFF). To view descriptions of AwsDynamoDbTable attributes, see AwsDynamoDbTableDetails in the *AWS Security Hub API Reference*.

Example


```
"AwsDynamoDbTable": {
    "AttributeDefinitions": [
        {
            "AttributeName": "attribute1",
            "AttributeType": "value 1"
        },
        {
```

```
            "AttributeName": "attribute2",
            "AttributeType": "value 2"
        },
        {
            "AttributeName": "attribute3",
            "AttributeType": "value 3"
        }
    ],
    "BillingModeSummary": {
        "BillingMode": "PAY_PER_REQUEST",
        "LastUpdateToPayPerRequestDateTime": "2019-12-03T15:23:10.323Z"
    },
    "CreationDateTime": "2019-12-03T15:23:10.248Z",
    "DeletionProtectionEnabled": true,
    "GlobalSecondaryIndexes": [
        {
            "Backfilling": false,
            "IndexArn": "arn:aws:dynamodb:us-west-2:111122223333:table/exampleTable/
index/exampleIndex",
            "IndexName": "standardsControlArnIndex",
            "IndexSizeBytes": 1862513,
            "IndexStatus": "ACTIVE",
            "ItemCount": 20,
            "KeySchema": [
                {
                    "AttributeName": "City",
                    "KeyType": "HASH"
                },
                {
                    "AttributeName": "Date",
                    "KeyType": "RANGE"
                }
            ],
            "Projection": {
                "NonKeyAttributes": ["predictorName"],
                "ProjectionType": "ALL"
            },
            "ProvisionedThroughput": {
                "LastIncreaseDateTime": "2019-03-14T13:21:00.399Z",
                "LastDecreaseDateTime": "2019-03-14T12:47:35.193Z",
                "NumberOfDecreasesToday": 0,
                "ReadCapacityUnits": 100,
                "WriteCapacityUnits": 50
            },
```

```
        }
    ],
    "GlobalTableVersion": "V1",
    "ItemCount": 2705,
    "KeySchema": [
        {
            "AttributeName": "zipcode",
            "KeyType": "HASH"
        }
     ],
     "LatestStreamArn": "arn:aws:dynamodb:us-west-2:111122223333:table/exampleTable/
stream/2019-12-03T23:23:10.248",
     "LatestStreamLabel": "2019-12-03T23:23:10.248",
     "LocalSecondaryIndexes": [
        {
            "IndexArn": "arn:aws:dynamodb:us-east-1:111122223333:table/exampleGroup/
index/exampleId",
            "IndexName": "CITY_DATE_INDEX_NAME",
            "KeySchema": [
                {
                    "AttributeName": "zipcode",
                    "KeyType": "HASH"
                }
            ],
            "Projection": {
                "NonKeyAttributes": ["predictorName"],
                "ProjectionType": "ALL"
            },
        }
    ],
    "ProvisionedThroughput": {
        "LastIncreaseDateTime": "2019-03-14T13:21:00.399Z",
        "LastDecreaseDateTime": "2019-03-14T12:47:35.193Z",
        "NumberOfDecreasesToday": 0,
        "ReadCapacityUnits": 100,
        "WriteCapacityUnits": 50
    },
    "Replicas": [
        {
            "GlobalSecondaryIndexes":[
                {
                    "IndexName": "CITY_DATE_INDEX_NAME",
                    "ProvisionedThroughputOverride": {
                        "ReadCapacityUnits": 10
```

```
                    }
                }
            ],
            "KmsMasterKeyId" : "KmsKeyId"
            "ProvisionedThroughputOverride": {
                "ReadCapacityUnits": 10
            },
            "RegionName": "regionName",
            "ReplicaStatus": "CREATING",
            "ReplicaStatusDescription": "replicaStatusDescription"
        }
    ],
    "RestoreSummary" : {
        "SourceBackupArn": "arn:aws:dynamodb:us-west-2:111122223333:table/exampleTable/
backup/backup1",
        "SourceTableArn": "arn:aws:dynamodb:us-west-2:111122223333:table/exampleTable",
        "RestoreDateTime": "2020-06-22T17:40:12.322Z",
        "RestoreInProgress": true
    },
    "SseDescription": {
        "InaccessibleEncryptionDateTime": "2018-01-26T23:50:05.000Z",
        "Status": "ENABLED",
        "SseType": "KMS",
        "KmsMasterKeyArn": "arn:aws:kms:us-east-1:111122223333:key/key1"
    },
    "StreamSpecification" : {
        "StreamEnabled": true,
        "StreamViewType": "NEW_IMAGE"
    },
    "TableId": "example-table-id-1",
    "TableName": "example-table",
    "TableSizeBytes": 1862513,
    "TableStatus": "ACTIVE"
}
```

AwsEc2 resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsEc2` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsEc2ClientVpnEndpoint

The `AwsEc2ClientVpnEndpoint` object provides information about an AWS Client VPN endpoint. A Client VPN endpoint is the resource that you create and configure to enable and manage client VPN sessions. It's the termination point for all client VPN sessions.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2ClientVpnEndpoint` object. To view descriptions of `AwsEc2ClientVpnEndpoint` attributes, see AwsEc2ClientVpnEndpointDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2ClientVpnEndpoint": {
    "AuthenticationOptions": [
        {
            "MutualAuthentication": {
                "ClientRootCertificateChainArn": "arn:aws:acm:us-
east-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            },
            "Type": "certificate-authentication"
        }
    ],
    "ClientCidrBlock": "10.0.0.0/22",
    "ClientConnectOptions": {
        "Enabled": false
    },
    "ClientLoginBannerOptions": {
        "Enabled": false
    },
    "ClientVpnEndpointId": "cvpn-endpoint-00c5d11fc4729f2a5",
    "ConnectionLogOptions": {
        "Enabled": false
    },
    "Description": "test",
    "DnsServer": ["10.0.0.0"],
    "ServerCertificateArn": "arn:aws:acm:us-east-1:123456789012:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "SecurityGroupIdSet": [
        "sg-0f7a177b82b443691"
    ],
    "SelfServicePortalUrl": "https://self-service.clientvpn.amazonaws.com/endpoints/
cvpn-endpoint-00c5d11fc4729f2a5",
    "SessionTimeoutHours": 24,
```

```
    "SplitTunnel": false,
    "TransportProtocol": "udp",
    "VpcId": "vpc-1a2b3c4d5e6f1a2b3",
    "VpnPort": 443
}
```

AwsEc2Eip

The `AwsEc2Eip` object provides information about an Elastic IP address.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Eip` object. To view descriptions of `AwsEc2Eip` attributes, see AwsEc2EipDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2Eip": {
    "InstanceId": "instance1",
    "PublicIp": "192.0.2.04",
    "AllocationId": "eipalloc-example-id-1",
    "AssociationId": "eipassoc-example-id-1",
    "Domain": "vpc",
    "PublicIpv4Pool": "anycompany",
    "NetworkBorderGroup": "eu-central-1",
    "NetworkInterfaceId": "eni-example-id-1",
    "NetworkInterfaceOwnerId": "777788889999",
    "PrivateIpAddress": "192.0.2.03"
}
```

AwsEc2Instance

The `AwsEc2Instance` object provides details about an Amazon EC2 instance.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Instance` object. To view descriptions of `AwsEc2Instance` attributes, see AwsEc2InstanceDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2Instance": {
    "IamInstanceProfileArn": "arn:aws:iam::123456789012:instance-profile/AdminRole",
    "ImageId": "ami-1234",
    "IpV4Addresses": [ "1.1.1.1" ],
```

```
    "IpV6Addresses": [ "2001:db8:1234:1a2b::123" ],
    "KeyName": "my_keypair",
    "LaunchedAt": "2018-05-08T16:46:19.000Z",
    "MetadataOptions": {
     "HttpEndpoint": "enabled",
     "HttpProtocolIpv6": "enabled",
     "HttpPutResponseHopLimit": 1,
     "HttpTokens": "optional",
     "InstanceMetadataTags": "disabled",
    },
    "Monitoring": {
     "State": "disabled"
    },
    "NetworkInterfaces": [
      {
         "NetworkInterfaceId": "eni-e5aa89a3"
      }
    ],
    "SubnetId": "subnet-123",
    "Type": "i3.xlarge",
    "VpcId": "vpc-123"
}
```

AwsEc2LaunchTemplate

The `AwsEc2LaunchTemplate` object contains details about an Amazon Elastic Compute Cloud launch template that specifies instance configuration information.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2LaunchTemplate` object. To view descriptions of `AwsEc2LaunchTemplate` attributes, see AwsEc2LaunchTemplateDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2LaunchTemplate": {
    "DefaultVersionNumber": "1",
    "ElasticGpuSpecifications": ["string"],
    "ElasticInferenceAccelerators": ["string"],
    "Id": "lt-0a16e9802800bdd85",
    "ImageId": "ami-0d5eff06f840b45e9",
    "LatestVersionNumber": "1",
    "LaunchTemplateData": {
     "BlockDeviceMappings": [{
```

```
      "DeviceName": "/dev/xvda",
      "Ebs": {
       "DeleteonTermination": true,
       "Encrypted": true,
       "SnapshotId": "snap-01047646ec075f543",
       "VolumeSize": 8,
       "VolumeType:" "gp2"
      }
     }],
     "MetadataOptions": {
      "HttpTokens": "enabled",
      "HttpPutResponseHopLimit" : 1
     },
     "Monitoring": {
      "Enabled": true,
     "NetworkInterfaces": [{
      "AssociatePublicIpAddress" : true,
     }],
    "LaunchTemplateName": "string",
    "LicenseSpecifications": ["string"],
    "SecurityGroupIds": ["sg-01fce87ad6e019725"],
    "SecurityGroups": ["string"],
    "TagSpecifications": ["string"]
}
```

AwsEc2NetworkAcl

The `AwsEc2NetworkAcl` object contains details about an Amazon EC2 network access control list (ACL).

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2NetworkAcl` object. To view descriptions of `AwsEc2NetworkAcl` attributes, see AwsEc2NetworkAclDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2NetworkAcl": {
    "IsDefault": false,
    "NetworkAclId": "acl-1234567890abcdef0",
    "OwnerId": "123456789012",
    "VpcId": "vpc-1234abcd",
    "Associations": [{
        "NetworkAclAssociationId": "aclassoc-abcd1234",
```

```
        "NetworkAclId": "acl-021345abcdef6789",
        "SubnetId": "subnet-abcd1234"
    }],
    "Entries": [{
        "CidrBlock": "10.24.34.0/23",
        "Egress": true,
        "IcmpTypeCode": {
            "Code": 10,
            "Type": 30
        },
        "Ipv6CidrBlock": "2001:DB8::/32",
        "PortRange": {
            "From": 20,
            "To": 40
        },
        "Protocol": "tcp",
        "RuleAction": "allow",
        "RuleNumber": 100
    }]
}
```

AwsEc2NetworkInterface

The `AwsEc2NetworkInterface` object provides information about an Amazon EC2 network interface.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2NetworkInterface` object. To view descriptions of `AwsEc2NetworkInterface` attributes, see AwsEc2NetworkInterfaceDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2NetworkInterface": {
    "Attachment": {
        "AttachTime": "2019-01-01T03:03:21Z",
        "AttachmentId": "eni-attach-43348162",
        "DeleteOnTermination": true,
        "DeviceIndex": 123,
        "InstanceId": "i-1234567890abcdef0",
        "InstanceOwnerId": "123456789012",
        "Status": 'ATTACHED'
    },
    "SecurityGroups": [
```

```
        {
            "GroupName": "my-security-group",
            "GroupId": "sg-903004f8"
        },
    ],
    "NetworkInterfaceId": 'eni-686ea200',
    "SourceDestCheck": false
}
```

AwsEc2RouteTable

The `AwsEc2RouteTable` object provides information about an Amazon EC2 route table.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2RouteTable` object. To view descriptions of `AwsEc2RouteTable` attributes, see AwsEc2RouteTableDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2RouteTable": {
    "AssociationSet": [{
     "AssociationSet": {
      "State": "associated"
        },
     "Main": true,
     "RouteTableAssociationId": "rtbassoc-08e706c45de9f7512",
     "RouteTableId": "rtb-0a59bde9cf2548e34",
    }],
    "PropogatingVgwSet": [],
    "RouteTableId": "rtb-0a59bde9cf2548e34",
    "RouteSet": [
     {
      "DestinationCidrBlock": "10.24.34.0/23",
      "GatewayId": "local",
      "Origin": "CreateRouteTable",
      "State": "active"
     },
     {
      "DestinationCidrBlock": "10.24.34.0/24",
      "GatewayId": "igw-0242c2d7d513fc5d3",
      "Origin": "CreateRoute",
      "State": "active"
     }
```

```
    ],
    "VpcId": "vpc-0c250a5c33f51d456"
}
```

AwsEc2SecurityGroup

The `AwsEc2SecurityGroup` object describes an Amazon EC2 security group.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2SecurityGroup` object. To view descriptions of `AwsEc2SecurityGroup` attributes, see AwsEc2SecurityGroupDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2SecurityGroup": {
    "GroupName": "MySecurityGroup",
    "GroupId": "sg-903004f8",
    "OwnerId": "123456789012",
    "VpcId": "vpc-1a2b3c4d",
    "IpPermissions": [
        {
            "IpProtocol": "-1",
            "IpRanges": [],
            "UserIdGroupPairs": [
                {
                    "UserId": "123456789012",
                    "GroupId": "sg-903004f8"
                }
            ],
            "PrefixListIds": [
                {"PrefixListId": "pl-63a5400a"}
            ]
        },
        {
            "PrefixListIds": [],
            "FromPort": 22,
            "IpRanges": [
                {
                    "CidrIp": "203.0.113.0/24"
                }
            ],
            "ToPort": 22,
            "IpProtocol": "tcp",
```

```
            "UserIdGroupPairs": []
        }
    ]
}
```

AwsEc2Subnet

The `AwsEc2Subnet` object provides information about a subnet in Amazon EC2.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Subnet` object. To view descriptions of `AwsEc2Subnet` attributes, see AwsEc2SubnetDetails in the *AWS Security Hub API Reference*.

Example

```
AwsEc2Subnet: {
    "AssignIpv6AddressOnCreation": false,
    "AvailabilityZone": "us-west-2c",
    "AvailabilityZoneId": "usw2-az3",
    "AvailableIpAddressCount": 8185,
    "CidrBlock": "10.0.0.0/24",
    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false,
    "OwnerId": "123456789012",
    "State": "available",
    "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/subnet-d5436c93",
    "SubnetId": "subnet-d5436c93",
    "VpcId": "vpc-153ade70",
    "Ipv6CidrBlockAssociationSet": [{
        "AssociationId": "subnet-cidr-assoc-EXAMPLE",
        "Ipv6CidrBlock": "2001:DB8::/32",
        "CidrBlockState": "associated"
   }]
}
```

AwsEc2TransitGateway

The `AwsEc2TransitGateway` object provides details about an Amazon EC2 transit gateway that interconnects your virtual private clouds (VPCs) and on-premises networks.

The following is an example `AwsEc2TransitGateway` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsEc2TransitGateway` attributes, see AwsEc2TransitGatewayDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2TransitGateway": {
 "AmazonSideAsn": 65000,
 "AssociationDefaultRouteTableId": "tgw-rtb-099ba47cbbea837cc",
 "AutoAcceptSharedAttachments": "disable",
 "DefaultRouteTableAssociation": "enable",
 "DefaultRouteTablePropagation": "enable",
 "Description": "sample transit gateway",
 "DnsSupport": "enable",
 "Id": "tgw-042ae6bf7a5c126c3",
 "MulticastSupport": "disable",
 "PropagationDefaultRouteTableId": "tgw-rtb-099ba47cbbea837cc",
 "TransitGatewayCidrBlocks": ["10.0.0.0/16"],
 "VpnEcmpSupport": "enable"
}
```

AwsEc2Volume

The `AwsEc2Volume` object provides details about an Amazon EC2 volume.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Volume` object. To view descriptions of `AwsEc2Volume` attributes, see AwsEc2VolumeDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2Volume": {
    "Attachments": [
      {
        "AttachTime": "2017-10-17T14:47:11Z",
        "DeleteOnTermination": true,
        "InstanceId": "i-123abc456def789g",
        "Status": "attached"
      }
     ],
    "CreateTime": "2020-02-24T15:54:30Z",
    "Encrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
    "Size": 80,
    "SnapshotId": "",
    "Status": "available"
```

```
}
```

AwsEc2Vpc

The `AwsEc2Vpc` object provides details about an Amazon EC2 VPC.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Vpc` object. To view descriptions of `AwsEc2Vpc` attributes, see AwsEc2VpcDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2Vpc": {
    "CidrBlockAssociationSet": [
        {
            "AssociationId": "vpc-cidr-assoc-0dc4c852f52abda97",
            "CidrBlock": "192.0.2.0/24",
            "CidrBlockState": "associated"
        }
    ],
    "DhcpOptionsId": "dopt-4e42ce28",
    "Ipv6CidrBlockAssociationSet": [
        {
            "AssociationId": "vpc-cidr-assoc-0dc4c852f52abda97",
            "CidrBlockState": "associated",
            "Ipv6CidrBlock": "192.0.2.0/24"
       }

    ],
    "State": "available"
}
```

AwsEc2VpcEndpointService

The `AwsEc2VpcEndpointService` object contains details about the service configuration for a VPC endpoint service.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2VpcEndpointService` object. To view descriptions of `AwsEc2VpcEndpointService` attributes, see AwsEc2VpcEndpointServiceDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2VpcEndpointService": {
    "ServiceType": [
      {
        "ServiceType": "Interface"
      }
    ],
    "ServiceId": "vpce-svc-example1",
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example1",
    "ServiceState": "Available",
    "AvailabilityZones": [
      "us-east-1"
    ],
    "AcceptanceRequired": true,
    "ManagesVpcEndpoints": false,
    "NetworkLoadBalancerArns": [
      "arn:aws:elasticloadbalancing:us-east-1:444455556666:loadbalancer/net/my-network-
load-balancer/example1"
    ],
    "GatewayLoadBalancerArns": [],
    "BaseEndpointDnsNames": [
      "vpce-svc-04eec859668b51c34.us-east-1.vpce.amazonaws.com"
    ],
    "PrivateDnsName": "my-private-dns"
}
```

AwsEc2VpcPeeringConnection

The `AwsEc2VpcPeeringConnection` object provides details about the networking connection between two VPCs.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2VpcPeeringConnection` object. To view descriptions of `AwsEc2VpcPeeringConnection` attributes, see AwsEc2VpcPeeringConnectionDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2VpcPeeringConnection": {
 "AccepterVpcInfo": {
  "CidrBlock": "10.0.0.0/28",
  "CidrBlockSet": [{
   "CidrBlock": "10.0.0.0/28"
```

```
  }],
  "Ipv6CidrBlockSet": [{
   "Ipv6CidrBlock": "2002::1234:abcd:ffff:c0a8:101/64"
  }],
  "OwnerId": "012345678910",
  "PeeringOptions": {
   "AllowDnsResolutionFromRemoteVpc": true,
   "AllowEgressFromLocalClassicLinkToRemoteVpc": false,
   "AllowEgressFromLocalVpcToRemoteClassicLink": true
  },
  "Region": "us-west-2",
  "VpcId": "vpc-i123456"
 },
 "ExpirationTime": "2022-02-18T15:31:53.161Z",
 "RequesterVpcInfo": {
  "CidrBlock": "192.168.0.0/28",
  "CidrBlockSet": [{
   "CidrBlock": "192.168.0.0/28"
  }],
  "Ipv6CidrBlockSet": [{
   "Ipv6CidrBlock": "2002::1234:abcd:ffff:c0a8:101/64"
  }],
  "OwnerId": "012345678910",
  "PeeringOptions": {
   "AllowDnsResolutionFromRemoteVpc": true,
   "AllowEgressFromLocalClassicLinkToRemoteVpc": false,
   "AllowEgressFromLocalVpcToRemoteClassicLink": true
  },
  "Region": "us-west-2",
  "VpcId": "vpc-i123456"
 },
 "Status": {
  "Code": "initiating-request",
  "Message": "Active"
 },
 "VpcPeeringConnectionId": "pcx-1a2b3c4d"
}
```


AwsEcr resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsEcr` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsEcrContainerImage

The `AwsEcrContainerImage` object provides information about an Amazon ECR image.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcrContainerImage` object. To view descriptions of `AwsEcrContainerImage` attributes, see AwsEcrContainerImageDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEcrContainerImage": {
    "RegistryId": "123456789012",
    "RepositoryName": "repository-name",
    "Architecture": "amd64"
    "ImageDigest":
 "sha256:a568e5c7a953fbeaa2904ac83401f93e4a076972dc1bae527832f5349cd2fb10",
    "ImageTags": ["00000000-0000-0000-0000-000000000000"],
    "ImagePublishedAt": "2019-10-01T20:06:12Z"
}
```

AwsEcrRepository

The `AwsEcrRepository` object provides information about an Amazon Elastic Container Registry repository.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcrRepository` object. To view descriptions of `AwsEcrRepository` attributes, see AwsEcrRepositoryDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEcrRepository": {
    "LifecyclePolicy": {
        "RegistryId": "123456789012",
    },
    "RepositoryName": "sample-repo",
    "Arn": "arn:aws:ecr:us-west-2:111122223333:repository/sample-repo",
    "ImageScanningConfiguration": {
        "ScanOnPush": true
    },
```

```
    "ImageTagMutability": "IMMUTABLE"
}
```

AwsEcs resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsEcs` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsEcsCluster

The `AwsEcsCluster` object provides details about an Amazon Elastic Container Service cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsCluster` object. To view descriptions of `AwsEcsCluster` attributes, see AwsEcsClusterDetails in the *AWS Security Hub API Reference*.

Example

```
    "AwsEcsCluster": {
        "CapacityProviders": [],
        "ClusterSettings": [
            {
                "Name": "containerInsights",
                "Value": "enabled"
            }
        ],
        "Configuration": {
            "ExecuteCommandConfiguration": {
                "KmsKeyId": "kmsKeyId",
                "LogConfiguration": {
                    "CloudWatchEncryptionEnabled": true,
                    "CloudWatchLogGroupName": "cloudWatchLogGroupName",
                    "S3BucketName": "s3BucketName",
                    "S3EncryptionEnabled": true,
                    "S3KeyPrefix": "s3KeyPrefix"
                },
                "Logging": "DEFAULT"
            }
        }
        "DefaultCapacityProviderStrategy": [
```

```
            {
                "Base": 0,
                "CapacityProvider": "capacityProvider",
                "Weight": 1
            }
        ]
    }
```

AwsEcsContainer

The `AwsEcsContainer` object contains details about an Amazon ECS container.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsContainer` object. To view descriptions of `AwsEcsContainer` attributes, see AwsEcsContainerDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEcsContainer": {
    "Image": "1111111/
knotejs@sha256:356131c9fef111111111111115f4ed8de5f9dce4dc3bd34bg21846588a3",
    "MountPoints": [{
        "ContainerPath": "/mnt/etc",
        "SourceVolume": "vol-03909e9"
    }],
    "Name": "knote",
    "Privileged": true
}
```

AwsEcsService

The `AwsEcsService` object provides details about a service within an Amazon ECS cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsService` object. To view descriptions of `AwsEcsService` attributes, see AwsEcsServiceDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEcsService": {
    "CapacityProviderStrategy": [
        {
            "Base": 12,
```

```
            "CapacityProvider": "",
            "Weight": ""
        }
    ],
    "Cluster": "arn:aws:ecs:us-east-1:111122223333:cluster/example-ecs-cluster",
    "DeploymentConfiguration": {
        "DeploymentCircuitBreaker": {
            "Enable": false,
            "Rollback": false
        },
        "MaximumPercent": 200,
        "MinimumHealthyPercent": 100
    },
    "DeploymentController": "",
    "DesiredCount": 1,
    "EnableEcsManagedTags": false,
    "EnableExecuteCommand": false,
    "HealthCheckGracePeriodSeconds": 1,
    "LaunchType": "FARGATE",
    "LoadBalancers": [
        {
            "ContainerName": "",
            "ContainerPort": 23,
            "LoadBalancerName": "",
            "TargetGroupArn": ""
        }
    ],
    "Name": "sample-app-service",
    "NetworkConfiguration": {
        "AwsVpcConfiguration": {
            "Subnets": [
                "Subnet-example1",
                "Subnet-example2"
            ],
        "SecurityGroups": [
                "Sg-0ce48e9a6e5b457f5"
        ],
        "AssignPublicIp": "ENABLED"
        }
    },
    "PlacementConstraints": [
        {
            "Expression": "",
            "Type": ""
```

```
        }
    ],
    "PlacementStrategies": [
        {
            "Field": "",
            "Type": ""
        }
    ],
    "PlatformVersion": "LATEST",
    "PropagateTags": "",
    "Role": "arn:aws:iam::111122223333:role/aws-servicerole/ecs.amazonaws.com/
ServiceRoleForECS",
    "SchedulingStrategy": "REPLICA",
    "ServiceName": "sample-app-service",
    "ServiceArn": "arn:aws:ecs:us-east-1:111122223333:service/example-ecs-cluster/
sample-app-service",
    "ServiceRegistries": [
        {
            "ContainerName": "",
            "ContainerPort": 1212,
            "Port": 1221,
            "RegistryArn": ""
        }
    ],
    "TaskDefinition": "arn:aws:ecs:us-east-1:111122223333:task-definition/example-
taskdef:1"
}
```

AwsEcsTask

The `AwsEcsTask` object provides details about an Amazon ECS task.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsTask` object. To view descriptions of `AwsEcsTask` attributes, see AwsEcsTask in the *AWS Security Hub API Reference*.

Example

```
"AwsEcsTask": {
 "ClusterArn": "arn:aws:ecs:us-west-2:123456789012:task/MyCluster/1234567890123456789",
 "CreatedAt": "1557134011644",
 "Group": "service:fargate-service",
 "StartedAt": "1557134011644",
```

```
 "StartedBy": "ecs-svc/1234567890123456789",
 "TaskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-definition/sample-
fargate:2",
 "Version": 3,
 "Volumes": [{
  "Name": "string",
  "Host": {
   "SourcePath": "string"
  }
 }],
 "Containers": {
  "Image": "1111111/
knotejs@sha256:356131c9fef111111111111115f4ed8de5f9dce4dc3bd34bg21846588a3",
  "MountPoints": [{
   "ContainerPath": "/mnt/etc",
   "SourceVolume": "vol-03909e9"
  }],
  "Name": "knote",
  "Privileged": true
 }
}
```

AwsEcsTaskDefinition

The `AwsEcsTaskDefinition` object contains details about a task definition. A task definition describes the container and volume definitions of an Amazon Elastic Container Service task.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsTaskDefinition` object. To view descriptions of `AwsEcsTaskDefinition` attributes, see AwsEcsTaskDefinitionDetails in the *AWS Security Hub API Reference*.

Example

```
    "AwsEcsTaskDefinition": {
        "ContainerDefinitions": [
            {
                "Command": ['ruby', 'hi.rb'],
                "Cpu":128,
                "Essential": true,
                "HealthCheck": {
                    "Command": ["CMD-SHELL", "curl -f http://localhost/ || exit 1"],
                    "Interval": 10,
                    "Retries": 3,
```

```
                "StartPeriod": 5,
                "Timeout": 20
            },
            "Image": "tongueroo/sinatra:latest",
            "Interactive": true,
            "Links": [],
            "LogConfiguration": {
                "LogDriver": "awslogs",
                "Options": {
                    "awslogs-group": "/ecs/sinatra-hi",
                    "awslogs-region": "ap-southeast-1",
                    "awslogs-stream-prefix": "ecs"
                },
                "SecretOptions": []

            },
            "MemoryReservation": 128,
            "Name": "web",
            "PortMappings": [
                {
                    "ContainerPort": 4567,
                    "HostPort":4567,
                    "Protocol": "tcp"
                }
            ],
            "Privileged": true,
            "StartTimeout": 10,
            "StopTimeout": 100,
        }
    ],
    "Family": "sinatra-hi",
    "NetworkMode": "host",
    "RequiresCompatibilities": ["EC2"],
    "Status": "ACTIVE",
    "TaskRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
}
```

AwsEfs resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsEfs` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsEfsAccessPoint

The `AwsEfsAccessPoint` object provides details about files stored in Amazon Elastic File System.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEfsAccessPoint` object. To view descriptions of `AwsEfsAccessPoint` attributes, see AwsEfsAccessPointDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEfsAccessPoint": {
 "AccessPointId": "fsap-05c4c0e79ba0b118a",
 "Arn": "arn:aws:elasticfilesystem:us-east-1:863155670886:access-point/
fsap-05c4c0e79ba0b118a",
 "ClientToken": "AccessPointCompliant-ASk06ZZSXsEp",
 "FileSystemId": "fs-0f8137f731cb32146",
 "PosixUser": {
  "Gid": "1000",
  "SecondaryGids": ["0", "4294967295"],
  "Uid": "1234"
 },
 "RootDirectory": {
  "CreationInfo": {
   "OwnerGid": "1000",
   "OwnerUid": "1234",
   "Permissions": "777"
  },
  "Path": "/tmp/example"
 }
}
```

AwsEks resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsEks` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsEksCluster

The `AwsEksCluster` object provides details about an Amazon EKS cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEksCluster` object. To view descriptions of `AwsEksCluster` attributes, see AwsEksClusterDetails in the *AWS Security Hub API Reference*.

Example

```
{
  "AwsEksCluster": {
    "Name": "example",
    "Arn": "arn:aws:eks:us-west-2:222222222222:cluster/example",
    "CreatedAt": 1565804921.901,
    "Version": "1.12",
    "RoleArn": "arn:aws:iam::222222222222:role/example-cluster-
ServiceRole-1XWBQWYSFRE2Q",
    "ResourcesVpcConfig": {
      "EndpointPublicAccess": false,
      "SubnetIds": [
        "subnet-021345abcdef6789",
        "subnet-abcdef01234567890",
        "subnet-1234567890abcdef0"
      ],
      "SecurityGroupIds": [
        "sg-abcdef01234567890"
      ]
    },
    "Logging": {
      "ClusterLogging": [
        {
          "Types": [
            "api",
            "audit",
            "authenticator",
            "controllerManager",
            "scheduler"
          ],
          "Enabled": true
        }
      ]
    },
    "Status": "CREATING",
```

```
    "CertificateAuthorityData": {},
  }
}
```

AwsElasticBeanstalk resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsElasticBeanstalk` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsElasticBeanstalkEnvironment

The `AwsElasticBeanstalkEnvironment` object contains details about an AWS Elastic Beanstalk environment.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsElasticBeanstalkEnvironment` object. To view descriptions of `AwsElasticBeanstalkEnvironment` attributes, see AwsElasticBeanstalkEnvironmentDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsElasticBeanstalkEnvironment": {
    "ApplicationName": "MyApplication",
    "Cname": "myexampleapp-env.devo-2.elasticbeanstalk-internal.com",
    "DateCreated": "2021-04-30T01:38:01.090Z",
    "DateUpdated": "2021-04-30T01:38:01.090Z",
    "Description": "Example description of my awesome application",
    "EndpointUrl": "eb-dv-e-p-AWSEBLoa-abcdef01234567890-021345abcdef6789.us-
east-1.elb.amazonaws.com",
    "EnvironmentArn": "arn:aws:elasticbeanstalk:us-east-1:123456789012:environment/
MyApplication/myapplication-env",
    "EnvironmentId": "e-abcd1234",
    "EnvironmentLinks": [
        {
            "EnvironmentName": "myexampleapp-env",
            "LinkName": "myapplicationLink"
        }
    ],
    "EnvironmentName": "myapplication-env",
```

```
    "OptionSettings": [
        {
            "Namespace": "aws:elasticbeanstalk:command",
            "OptionName": "BatchSize",
            "Value": "100"
        },
        {
            "Namespace": "aws:elasticbeanstalk:command",
            "OptionName": "Timeout",
            "Value": "600"
        },
        {
            "Namespace": "aws:elasticbeanstalk:command",
            "OptionName": "BatchSizeType",
            "Value": "Percentage"
        },
        {
            "Namespace": "aws:elasticbeanstalk:command",
            "OptionName": "IgnoreHealthCheck",
            "Value": "false"
        },
        {
            "Namespace": "aws:elasticbeanstalk:application",
            "OptionName": "Application Healthcheck URL",
            "Value": "TCP:80"
        }
    ],
    "PlatformArn": "arn:aws:elasticbeanstalk:us-east-1::platform/Tomcat 8 with Java 8
 running on 64bit Amazon Linux/2.7.7",
    "SolutionStackName": "64bit Amazon Linux 2017.09 v2.7.7 running Tomcat 8 Java 8",
    "Status": "Ready",
    "Tier": {
        "Name": "WebServer"
       "Type": "Standard"
       "Version": "1.0"
    },
    "VersionLabel": "Sample Application"
}
```


AwsElasticSearch resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsElasticSearch` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsElasticSearchDomain

The `AwsElasticSearchDomain` object provides details about an Amazon OpenSearch Service domain.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsElasticSearchDomain` object. To view descriptions of `AwsElasticSearchDomain` attributes, see AwsElasticSearchDomainDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsElasticSearchDomain": {
    "AccessPolicies": "string",
    "DomainStatus": {
           "DomainId": "string",
           "DomainName": "string",
           "Endpoint": "string",
           "Endpoints": {
                  "string": "string"
           }
    },
    "DomainEndpointOptions": {
           "EnforceHTTPS": boolean,
           "TLSSecurityPolicy": "string"
    },
    "ElasticsearchClusterConfig": {
           "DedicatedMasterCount": number,
           "DedicatedMasterEnabled": boolean,
           "DedicatedMasterType": "string",
           "InstanceCount": number,
           "InstanceType": "string",
           "ZoneAwarenessConfig": {
                  "AvailabilityZoneCount": number
           },
           "ZoneAwarenessEnabled": boolean
    },
    "ElasticsearchVersion": "string",
    "EncryptionAtRestOptions": {
           "Enabled": boolean,
           "KmsKeyId": "string"
```

```
    },
    "LogPublishingOptions": {
           "AuditLogs": {
                  "CloudWatchLogsLogGroupArn": "string",
                  "Enabled": boolean
           },
           "IndexSlowLogs": {
                  "CloudWatchLogsLogGroupArn": "string",
                  "Enabled": boolean
           },
           "SearchSlowLogs": {
                  "CloudWatchLogsLogGroupArn": "string",
                  "Enabled": boolean
           }
    },
    "NodeToNodeEncryptionOptions": {
           "Enabled": boolean
    },
    "ServiceSoftwareOptions": {
           "AutomatedUpdateDate": "string",
           "Cancellable": boolean,
           "CurrentVersion": "string",
           "Description": "string",
           "NewVersion": "string",
           "UpdateAvailable": boolean,
           "UpdateStatus": "string"
    },
    "VPCOptions": {
           "AvailabilityZones": [
                 "string"
           ],
           "SecurityGroupIds": [
                 "string"
           ],
           "SubnetIds": [
                 "string"
           ],
          "VPCId": "string"
    }
}
```

AwsElb resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsElb` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsElbLoadBalancer

The `AwsElbLoadBalancer` object contains details about a Classic Load Balancer.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsElbLoadBalancer` object. To view descriptions of `AwsElbLoadBalancer` attributes, see AwsElbLoadBalancerDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsElbLoadBalancer": {
    "AvailabilityZones": ["us-west-2a"],
    "BackendServerDescriptions": [
         {
            "InstancePort": 80,
            "PolicyNames": ["doc-example-policy"]
        }
    ],
    "CanonicalHostedZoneName": "Z3DZXE0EXAMPLE",
    "CanonicalHostedZoneNameID": "my-load-balancer-444455556666.us-
west-2.elb.amazonaws.com",
    "CreatedTime": "2020-08-03T19:22:44.637Z",
    "DnsName": "my-load-balancer-444455556666.us-west-2.elb.amazonaws.com",
    "HealthCheck": {
        "HealthyThreshold": 2,
        "Interval": 30,
        "Target": "HTTP:80/png",
        "Timeout": 3,
        "UnhealthyThreshold": 2
    },
    "Instances": [
        {
            "InstanceId": "i-example"
        }
    ],
    "ListenerDescriptions": [
```

```
        {
            "Listener": {
                "InstancePort": 443,
                "InstanceProtocol": "HTTPS",
                "LoadBalancerPort": 443,
                "Protocol": "HTTPS",
                "SslCertificateId": "arn:aws:iam::444455556666:server-certificate/my-
server-cert"
            },
            "PolicyNames": ["ELBSecurityPolicy-TLS-1-2-2017-01"]
        }
    ],
    "LoadBalancerAttributes": {
        "AccessLog": {
            "EmitInterval": 60,
            "Enabled": true,
            "S3BucketName": "amzn-s3-demo-bucket",
            "S3BucketPrefix": "doc-example-prefix"
        },
        "ConnectionDraining": {
            "Enabled": false,
            "Timeout": 300
        },
        "ConnectionSettings": {
            "IdleTimeout": 30
        },
        "CrossZoneLoadBalancing": {
            "Enabled": true
        },
        "AdditionalAttributes": [{
            "Key": "elb.http.desyncmitigationmode",
            "Value": "strictest"
        }]

    },
    "LoadBalancerName": "example-load-balancer",
    "Policies": {
        "AppCookieStickinessPolicies": [
            {
                "CookieName": "",
                "PolicyName": ""
            }
        ],
        "LbCookieStickinessPolicies": [
```

```
            {
                "CookieExpirationPeriod": 60,
                "PolicyName": "my-example-cookie-policy"
            }
        ],
        "OtherPolicies": [
            "my-PublicKey-policy",
            "my-authentication-policy",
            "my-SSLNegotiation-policy",
            "my-ProxyProtocol-policy",
            "ELBSecurityPolicy-2015-03"
        ]
    },
    "Scheme": "internet-facing",
    "SecurityGroups": ["sg-example"],
    "SourceSecurityGroup": {
        "GroupName": "my-elb-example-group",
        "OwnerAlias": "444455556666"
    },
    "Subnets": ["subnet-example"],
    "VpcId": "vpc-a01106c2"
}
```

AwsElbv2LoadBalancer

The `AwsElbv2LoadBalancer` object provides information about a load balancer.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsElbv2LoadBalancer` object. To view descriptions of `AwsElbv2LoadBalancer` attributes, see AwsElbv2LoadBalancerDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsElbv2LoadBalancer": {
                        "AvailabilityZones": {
                            "SubnetId": "string",
                            "ZoneName": "string"
                        },
                        "CanonicalHostedZoneId": "string",
                        "CreatedTime": "string",
                        "DNSName": "string",
                        "IpAddressType": "string",
                        "LoadBalancerAttributes": [
```

```
                        {
                            "Key": "string",
                            "Value": "string"
                        }
                    ],
                    "Scheme": "string",
                    "SecurityGroups": [ "string" ],
                    "State": {
                        "Code": "string",
                        "Reason": "string"
                    },
                    "Type": "string",
                    "VpcId": "string"
                }
```

AwsEventBridge resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsEventBridge` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsEventSchemasRegistry

The `AwsEventSchemasRegistry` object provides information about an Amazon EventBridge schema registry. A schema defines the structure of events that are sent to EventBridge. Schema registries are containers that collect and logically group your schemas.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEventSchemasRegistry` object. To view descriptions of `AwsEventSchemasRegistry` attributes, see AwsEventSchemasRegistry in the *AWS Security Hub API Reference*.

Example

```
"AwsEventSchemasRegistry": {
    "Description": "This is an example event schema registry.",
    "RegistryArn": "arn:aws:schemas:us-east-1:123456789012:registry/schema-registry",
    "RegistryName": "schema-registry"
}
```

AwsEventsEndpoint

The `AwsEventsEndpoint` object provides information about an Amazon EventBridge global endpoint. The endpoint can improve your application's availability by making it Regional-fault tolerant.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEventsEndpoint` object. To view descriptions of `AwsEventsEndpoint` attributes, see AwsEventsEndpointDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEventsEndpoint": {
    "Arn": "arn:aws:events:us-east-1:123456789012:endpoint/my-endpoint",
    "Description": "This is a sample endpoint.",
    "EndpointId": "04k1exajoy.veo",
    "EndpointUrl": "https://04k1exajoy.veo.endpoint.events.amazonaws.com",
    "EventBuses": [
        {
            "EventBusArn": "arn:aws:events:us-east-1:123456789012:event-bus/default"
        },
        {
            "EventBusArn": "arn:aws:events:us-east-2:123456789012:event-bus/default"
        }
    ],
    "Name": "my-endpoint",
    "ReplicationConfig": {
        "State": "ENABLED"
    },
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/
Amazon_EventBridge_Invoke_Event_Bus_1258925394",
    "RoutingConfig": {
        "FailoverConfig": {
            "Primary": {
                "HealthCheck": "arn:aws:route53:::healthcheck/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
            },
            "Secondary": {
                "Route": "us-east-2"
            }
        }
    },
    "State": "ACTIVE"
```

```
}
```

AwsEventsEventbus

The `AwsEventsEventbus` object provides information about an Amazon EventBridge global endpoint. The endpoint can improve your application's availability by making it Regional-fault tolerant.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEventsEventbus` object. To view descriptions of `AwsEventsEventbus` attributes, see AwsEventsEventbusDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEventsEventbus":
    "Arn": "arn:aws:events:us-east-1:123456789012:event-bus/my-event-bus",
    "Name": "my-event-bus",
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"AllowAllAccountsFromOrganizationToPutEvents\",\"Effect\":\"Allow
\",\"Principal\":\"*\",\"Action\":\"events:PutEvents\",\"Resource\":
\"arn:aws:events:us-east-1:123456789012:event-bus/my-event-bus\",\"Condition
\":{\"StringEquals\":{\"aws:PrincipalOrgID\":\"o-ki7yjtkjv5\"}}},{\"Sid\":
\"AllowAccountToManageRulesTheyCreated\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
\"arn:aws:iam::123456789012:root\"},\"Action\":[\"events:PutRule\",\"events:PutTargets
\",\"events:DeleteRule\",\"events:RemoveTargets\",\"events:DisableRule
\",\"events:EnableRule\",\"events:TagResource\",\"events:UntagResource\",
\"events:DescribeRule\",\"events:ListTargetsByRule\",\"events:ListTagsForResource\"],
\"Resource\":\"arn:aws:events:us-east-1:123456789012:rule/my-event-bus\",\"Condition\":
{\"StringEqualsIfExists\":{\"events:creatorAccount\":\"123456789012\"}}}]}"
```

AwsGuardDuty resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsGuardDuty` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsGuardDutyDetector

The `AwsGuardDutyDetector` object provides information about an Amazon GuardDuty detector. A detector is an object that represents the GuardDuty service. A detector is required for GuardDuty to become operational.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsGuardDutyDetector` object. To view descriptions of `AwsGuardDutyDetector` attributes, see AwsGuardDutyDetector in the *AWS Security Hub API Reference*.

Example

```
"AwsGuardDutyDetector": {
    "FindingPublishingFrequency": "SIX_HOURS",
    "ServiceRole": "arn:aws:iam::123456789012:role/aws-service-role/
guardduty.amazonaws.com/AWSServiceRoleForAmazonGuardDuty",
    "Status": "ENABLED",
    "DataSources": {
        "CloudTrail": {
            "Status": "ENABLED"
        },
        "DnsLogs": {
            "Status": "ENABLED"
        },
        "FlowLogs": {
            "Status": "ENABLED"
        },
        "S3Logs": {
             "Status": "ENABLED"
         },
         "Kubernetes": {
             "AuditLogs": {
                "Status": "ENABLED"
             }
         },
         "MalwareProtection": {
             "ScanEc2InstanceWithFindings": {
                "EbsVolumes": {
                    "Status": "ENABLED"
                 }
             },
            "ServiceRole": "arn:aws:iam::123456789012:role/aws-service-role/malware-
protection.guardduty.amazonaws.com/AWSServiceRoleForAmazonGuardDutyMalwareProtection"
```

```
        }
    }
}
```

AwsIam resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsIam` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsIamAccessKey

The `AwsIamAccessKey` object contains details about an IAM access key that is related to a finding.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamAccessKey` object. To view descriptions of `AwsIamAccessKey` attributes, see AwsIamAccessKeyDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsIamAccessKey": {
                        "AccessKeyId": "string",
                        "AccountId": "string",
                        "CreatedAt": "string",
                        "PrincipalId": "string",
                        "PrincipalName": "string",
                        "PrincipalType": "string",
                        "SessionContext": {
                            "Attributes": {
                                "CreationDate": "string",
                                "MfaAuthenticated": boolean
                            },
                            "SessionIssuer": {
                                "AccountId": "string",
                                "Arn": "string",
                                "PrincipalId": "string",
                                "Type": "string",
                                "UserName": "string"
                            }
                        },
                        "Status": "string"
```

```
                    }
```

AwsIamGroup

The `AwsIamGroup` object contains details about an IAM group.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamGroup` object. To view descriptions of `AwsIamGroup` attributes, see AwsIamGroupDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsIamGroup": {
    "AttachedManagedPolicies": [
        {
            "PolicyArn": "arn:aws:iam::aws:policy/ExampleManagedAccess",
            "PolicyName": "ExampleManagedAccess",
        }
    ],
    "CreateDate": "2020-04-28T14:08:37.000Z",
    "GroupId": "AGPA4TPS3VLP7QEXAMPLE",
    "GroupName": "Example_User_Group",
    "GroupPolicyList": [
        {
            "PolicyName": "ExampleGroupPolicy"
        }
    ],
    "Path": "/"
}
```

AwsIamPolicy

The `AwsIamPolicy` object represents an IAM permissions policy.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamPolicy` object. To view descriptions of `AwsIamPolicy` attributes, see AwsIamPolicyDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsIamPolicy": {
    "AttachmentCount": 1,
    "CreateDate": "2017-09-14T08:17:29.000Z",
```

```
    "DefaultVersionId": "v1",
    "Description": "Example IAM policy",
    "IsAttachable": true,
    "Path": "/",
    "PermissionsBoundaryUsageCount": 5,
    "PolicyId": "ANPAJ2UCCR6DPCEXAMPLE",
    "PolicyName": "EXAMPLE-MANAGED-POLICY",
    "PolicyVersionList": [
        {
            "VersionId": "v1",
            "IsDefaultVersion": true,
            "CreateDate": "2017-09-14T08:17:29.000Z"
        }
    ],
    "UpdateDate": "2017-09-14T08:17:29.000Z"
}
```

AwsIamRole

The `AwsIamRole` object contains information about an IAM role, including all of the role's policies.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamRole` object. To view descriptions of `AwsIamRole` attributes, see AwsIamRoleDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsIamRole": {
    "AssumeRolePolicyDocument": "{'Version': '2012-10-17','Statement': [{'Effect':
 'Allow','Action': 'sts:AssumeRole'}]}",
    "AttachedManagedPolicies": [
        {
            "PolicyArn": "arn:aws:iam::aws:policy/ExamplePolicy1",
            "PolicyName": "Example policy 1"
        },
        {
            "PolicyArn": "arn:aws:iam::444455556666:policy/ExamplePolicy2",
            "PolicyName": "Example policy 2"
        }
        ],
        "CreateDate": "2020-03-14T07:19:14.000Z",
        "InstanceProfileList": [
            {
```

```
                "Arn": "arn:aws:iam::333333333333:ExampleProfile",
                "CreateDate": "2020-03-11T00:02:27Z",
                "InstanceProfileId": "AIPAIXEU4NUHUPEXAMPLE",
                "InstanceProfileName": "ExampleInstanceProfile",
                "Path": "/",
                "Roles": [
                    {
                       "Arn": "arn:aws:iam::444455556666:role/example-role",
                        "AssumeRolePolicyDocument": "",
                        "CreateDate": "2020-03-11T00:02:27Z",
                        "Path": "/",
                        "RoleId": "AROAJ52OTH4H7LEXAMPLE",
                        "RoleName": "example-role",
                    }
                ]
            }
        ],
        "MaxSessionDuration": 3600,
        "Path": "/",
        "PermissionsBoundary": {
            "PermissionsBoundaryArn": "arn:aws:iam::aws:policy/AdministratorAccess",
            "PermissionsBoundaryType": "PermissionsBoundaryPolicy"
        },
        "RoleId": "AROA4TPS3VLEXAMPLE",
        "RoleName": "BONESBootstrapHydra-OverbridgeOpsFunctionsLambda",
        "RolePolicyList": [
            {
                "PolicyName": "Example role policy"
            }
        ]
    }
```

AwsIamUser

The `AwsIamUser` object provides information about a user.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamUser` object. To view descriptions of `AwsIamUser` attributes, see AwsIamUserDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsIamUser": {
```

```
    "AttachedManagedPolicies": [
        {
            "PolicyName": "ExamplePolicy",
            "PolicyArn": "arn:aws:iam::aws:policy/ExampleAccess"
        }
    ],
    "CreateDate": "2018-01-26T23:50:05.000Z",
    "GroupList": [],
    "Path": "/",
    "PermissionsBoundary" : {
        "PermissionsBoundaryArn" : "arn:aws:iam::aws:policy/AdministratorAccess",
        "PermissionsBoundaryType" : "PermissionsBoundaryPolicy"
    },
    "UserId": "AIDACKCEVSQ6C2EXAMPLE",
    "UserName": "ExampleUser",
    "UserPolicyList": [
        {
            "PolicyName": "InstancePolicy"
        }
    ]
}
```

AwsKinesis resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsKinesis` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsKinesisStream

The `AwsKinesisStream` object provides details about Amazon Kinesis Data Streams.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsKinesisStream` object. To view descriptions of `AwsKinesisStream` attributes, see AwsKinesisStreamDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsKinesisStream": {
 "Name": "test-vir-kinesis-stream",
 "Arn": "arn:aws:kinesis:us-east-1:293279581038:stream/test-vir-kinesis-stream",
```

```
 "RetentionPeriodHours": 24,
 "ShardCount": 2,
 "StreamEncryption": {
  "EncryptionType": "KMS",
  "KeyId": "arn:aws:kms:us-east-1:293279581038:key/849cf029-4143-4c59-91f8-
ea76007247eb"
 }
}
```

AwsKms resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsKms` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsKmsKey

The `AwsKmsKey` object provides details about an AWS KMS key.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsKmsKey` object. To view descriptions of `AwsKmsKey` attributes, see AwsKmsKeyDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsKmsKey": {
                      "AWSAccountId": "string",
                      "CreationDate": "string",
                      "Description": "string",
                      "KeyId": "string",
                      "KeyManager": "string",
                      "KeyRotationStatus": boolean,
                      "KeyState": "string",
                      "Origin": "string"
                  }
```

AwsLambda

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsLambda` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsLambdaFunction

The `AwsLambdaFunction` object provides details about a Lambda function's configuration.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsLambdaFunction` object. To view descriptions of `AwsLambdaFunction` attributes, see AwsLambdaFunctionDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsLambdaFunction": {
    "Architectures": [
        "x86_64"
    ],
    "Code": {
        "S3Bucket": "amzn-s3-demo-bucket",
        "S3Key": "samplekey",
        "S3ObjectVersion": "2",
        "ZipFile": "myzip.zip"
    },
    "CodeSha256": "1111111111111abcdef",
    "DeadLetterConfig": {
        "TargetArn": "arn:aws:lambda:us-east-2:123456789012:queue:myqueue:2"
    },
    "Environment": {
        "Variables": {
            "Stage": "foobar"
         },
        "Error": {
            "ErrorCode": "Sample-error-code",
            "Message": "Caller principal is a manager."
         }
     },
    "FunctionName": "CheckOut",
    "Handler": "main.py:lambda_handler",
    "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/mykey",
    "LastModified": "2001-09-11T09:00:00Z",
    "Layers": {
        "Arn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer:3",
        "CodeSize": 169
```

```
    },
    "PackageType": "Zip",
    "RevisionId": "23",
    "Role": "arn:aws:iam::123456789012:role/Accounting-Role",
    "Runtime": "go1.7",
    "Timeout": 15,
    "TracingConfig": {
        "Mode": "Active"
    },
    "Version": "$LATEST$",
    "VpcConfig": {
        "SecurityGroupIds": ["sg-085912345678492fb", "sg-08591234567bdgdc"],
         "SubnetIds": ["subnet-071f712345678e7c8", "subnet-07fd123456788a036"]
    },
    "MasterArn": "arn:aws:lambda:us-east-2:123456789012:\$LATEST",
    "MemorySize": 2048
}
```

AwsLambdaLayerVersion

The `AwsLambdaLayerVersion` object provides details about a Lambda layer version.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsLambdaLayerVersion` object. To view descriptions of `AwsLambdaLayerVersion` attributes, see AwsLambdaLayerVersionDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsLambdaLayerVersion": {
    "Version": 2,
    "CompatibleRuntimes": [
        "java8"
    ],
    "CreatedDate": "2019-10-09T22:02:00.274+0000"
}
```

AwsMsk resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsMsk` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsMskCluster

The `AwsMskCluster` object provides information about an Amazon Managed Streaming for Apache Kafka (Amazon MSK) cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsMskCluster` object. To view descriptions of `AwsMskCluster` attributes, see AwsMskClusterDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsMskCluster": {
        "ClusterInfo": {
            "ClientAuthentication": {
                "Sasl": {
                    "Scram": {
                        "Enabled": true
                    },
                    "Iam": {
                        "Enabled": true
                    }
                },
                "Tls": {
                    "CertificateAuthorityArnList": [],
                    "Enabled": false
                },
                "Unauthenticated": {
                    "Enabled": false
                }
            },
            "ClusterName": "my-cluster",
            "CurrentVersion": "K2PWKAKR8XB7XF",
            "EncryptionInfo": {
                "EncryptionAtRest": {
                    "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
                },
                "EncryptionInTransit": {
                    "ClientBroker": "TLS",
                    "InCluster": true
                }
            },
            "EnhancedMonitoring": "PER_TOPIC_PER_BROKER",
```

```
            "NumberOfBrokerNodes": 3
        }
}
```

AwsNetworkFirewall resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsNetworkFirewall` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsNetworkFirewallFirewall

The `AwsNetworkFirewallFirewall` object contains details about an AWS Network Firewall firewall.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsNetworkFirewallFirewall` object. To view descriptions of `AwsNetworkFirewallFirewall` attributes, see AwsNetworkFirewallFirewallDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsNetworkFirewallFirewall": {
    "DeleteProtection": false,
    "FirewallArn": "arn:aws:network-firewall:us-east-1:024665936331:firewall/
testfirewall",
    "FirewallPolicyArn": "arn:aws:network-firewall:us-east-1:444455556666:firewall-
policy/InitialFirewall",
    "FirewallId": "dea7d8e9-ae38-4a8a-b022-672a830a99fa",
    "FirewallName": "testfirewall",
    "FirewallPolicyChangeProtection": false,
    "SubnetChangeProtection": false,
    "SubnetMappings": [
        {
            "SubnetId": "subnet-0183481095e588cdc"
        },
        {
            "SubnetId": "subnet-01f518fad1b1c90b0"
        }
    ],
```

```
    "VpcId": "vpc-40e83c38"
}
```

AwsNetworkFirewallFirewallPolicy

The `AwsNetworkFirewallFirewallPolicy` object provides details about a firewall policy. A firewall policy defines the behavior of a network firewall.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsNetworkFirewallFirewallPolicy` object. To view descriptions of `AwsNetworkFirewallFirewallPolicy` attributes, see AwsNetworkFirewallFirewallPolicyDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsNetworkFirewallFirewallPolicy": {
   "FirewallPolicy": {
    "StatefulRuleGroupReferences": [
        {
            "ResourceArn": "arn:aws:network-firewall:us-east-1:444455556666:stateful-
rulegroup/PatchesOnly"
        }
    ],
    "StatelessDefaultActions": [ "aws:forward_to_sfe" ],
    "StatelessFragmentDefaultActions": [ "aws:forward_to_sfe" ],
    "StatelessRuleGroupReferences": [
       {
          "Priority": 1,
          "ResourceArn": "arn:aws:network-firewall:us-east-1:444455556666:stateless-
rulegroup/Stateless-1"
       }
     ]
   },
   "FirewallPolicyArn": "arn:aws:network-firewall:us-east-1:444455556666:firewall-
policy/InitialFirewall",
   "FirewallPolicyId": "9ceeda22-6050-4048-a0ca-50ce47f0cc65",
   "FirewallPolicyName": "InitialFirewall",
   "Description": "Initial firewall"
}
```

AwsNetworkFirewallRuleGroup

The `AwsNetworkFirewallRuleGroup` object provides details about an AWS Network Firewall rule group. Rule groups are used to inspect and control network traffic. Stateless rule groups apply to individual packets. Stateful rule groups apply to packets in the context of their traffic flow.

Rule groups are referenced in firewall policies.

The following examples show the AWS Security Finding Format (ASFF) for the `AwsNetworkFirewallRuleGroup` object. To view descriptions of `AwsNetworkFirewallRuleGroup` attributes, see AwsNetworkFirewallRuleGroupDetails in the *AWS Security Hub API Reference*.

Example – stateless rule group

```
"AwsNetworkFirewallRuleGroup": {
    "Capacity": 600,
    "RuleGroupArn": "arn:aws:network-firewall:us-east-1:444455556666:stateless-
rulegroup/Stateless-1",
    "RuleGroupId": "fb13c4df-b6da-4c1e-91ec-84b7a5487493",
    "RuleGroupName": "Stateless-1"
    "Description": "Example of a stateless rule group",
    "Type": "STATELESS",
    "RuleGroup": {
        "RulesSource": {
            "StatelessRulesAndCustomActions": {
                "CustomActions": [],
                "StatelessRules": [
                    {
                        "Priority": 1,
                        "RuleDefinition": {
                            "Actions": [
                                "aws:pass"
                            ],
                            "MatchAttributes": {
                                "DestinationPorts": [
                                    {
                                        "FromPort": 443,
                                        "ToPort": 443
                                    }
                                ],
                                "Destinations": [
                                    {
```

```
                                    "AddressDefinition": "192.0.2.0/24"
                                }
                            ],
                            "Protocols": [
                                        6
                            ],
                            "SourcePorts": [
                                {
                                    "FromPort": 0,
                                    "ToPort": 65535
                                }
                            ],
                            "Sources": [
                                {
                                     "AddressDefinition": "198.51.100.0/24"
                                }
                            ]
                        }
                    }
                }
            ]
        }
    }
}
```

Example – stateful rule group

```
"AwsNetworkFirewallRuleGroup": {
    "Capacity": 100,
    "RuleGroupArn": "arn:aws:network-firewall:us-east-1:444455556666:stateful-
rulegroup/tupletest",
    "RuleGroupId": "38b71c12-da80-4643-a6c5-03337f8933e0",
    "RuleGroupName": "ExampleRuleGroup",
    "Description": "Example of a stateful rule group",
    "Type": "STATEFUL",
    "RuleGroup": {
        "RuleSource": {
            "StatefulRules": [
                {
                    "Action": "PASS",
                    "Header": {
                        "Destination": "Any",
```

```
                        "DestinationPort": "443",
                        "Direction": "ANY",
                        "Protocol": "TCP",
                        "Source": "Any",
                        "SourcePort": "Any"
                    },
                    "RuleOptions": [
                        {
                           "Keyword": "sid:1"
                        }
                    ]
                }
            ]
        }
    }
}
```

The following is a list of valid value examples for `AwsNetworkFirewallRuleGroup` attributes:

- `Action`

  Valid values: `PASS` | `DROP` | `ALERT`

- `Protocol`

  Valid values: `IP` | `TCP` | `UDP` | `ICMP` | `HTTP` | `FTP` | `TLS` | `SMB` | `DNS` | `DCERPC` | `SSH` | `SMTP` | `IMAP` | `MSN` | `KRB5` | `IKEV2` | `TFTP` | `NTP` | `DHCP`

- `Flags`

  Valid values: `FIN` | `SYN` | `RST` | `PSH` | `ACK` | `URG` | `ECE` | `CWR`

- `Masks`

  Valid values: `FIN` | `SYN` | `RST` | `PSH` | `ACK` | `URG` | `ECE` | `CWR`

AwsOpenSearchService resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsOpenSearchService` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsOpenSearchServiceDomain

The `AwsOpenSearchServiceDomain` object contains information about an Amazon OpenSearch Service domain.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsOpenSearchServiceDomain` object. To view descriptions of `AwsOpenSearchServiceDomain` attributes, see AwsOpenSearchServiceDomainDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsOpenSearchServiceDomain": {
    "AccessPolicies": "IAM_Id",
    "AdvancedSecurityOptions": {
        "Enabled": true,
        "InternalUserDatabaseEnabled": true,
        "MasterUserOptions": {
            "MasterUserArn": "arn:aws:iam::123456789012:user/third-master-use",
            "MasterUserName": "third-master-use",
            "MasterUserPassword": "some-password"
        }
    },
    "Arn": "arn:aws:Opensearch:us-east-1:111122223333:somedomain",
    "ClusterConfig": {
        "InstanceType": "c5.large.search",
        "InstanceCount": 1,
        "DedicatedMasterEnabled": true,
        "ZoneAwarenessEnabled": false,
        "ZoneAwarenessConfig": {
            "AvailabilityZoneCount": 2
        },
        "DedicatedMasterType": "c5.large.search",
        "DedicatedMasterCount": 3,
        "WarmEnabled": true,
        "WarmCount": 3,
        "WarmType": "ultrawarm1.large.search"
    },
    "DomainEndpoint": "https://es-2021-06-23t17-04-qowmgghud5vofgb5e4wmi.eu-
central-1.es.amazonaws.com",
    "DomainEndpointOptions": {
        "EnforceHTTPS": false,
        "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07",
```

```
        "CustomEndpointCertificateArn": "arn:aws:acm:us-
east-1:111122223333:certificate/bda1bff1-79c0-49d0-abe6-50a15a7477d4",
        "CustomEndpointEnabled": true,
        "CustomEndpoint": "example.com"
    },
    "DomainEndpoints": {
        "vpc": "vpc-endpoint-h2dsd34efgyghrtguk5gt6j2foh4.us-east-1.es.amazonaws.com"
    },
    "DomainName": "my-domain",
    "EncryptionAtRestOptions": {
        "Enabled": false,
        "KmsKeyId": "1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
    },
    "EngineVersion": "7.1",
    "Id": "123456789012",
    "LogPublishingOptions": {
        "IndexSlowLogs": {
            "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-
group:/aws/aes/domains/es-index-slow-logs",
            "Enabled": true
        },
        "SearchSlowLogs": {
            "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-
group:/aws/aes/domains/es-slow-logs",
            "Enabled": true
        },
        "AuditLogs": {
            "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-
group:/aws/aes/domains/es-slow-logs",
            "Enabled": true
        }
    },
    "NodeToNodeEncryptionOptions": {
        "Enabled": true
    },
    "ServiceSoftwareOptions": {
        "AutomatedUpdateDate": "2022-04-28T14:08:37.000Z",
        "Cancellable": false,
        "CurrentVersion": "R20210331",
        "Description": "There is no software update available for this domain.",
        "NewVersion": "OpenSearch_1.0",
        "UpdateAvailable": false,
        "UpdateStatus": "COMPLETED",
        "OptionalDeployment": false
```

```
    },
    "VpcOptions": {
        "SecurityGroupIds": [
            "sg-2a3a4a5a"
        ],
        "SubnetIds": [
            "subnet-1a2a3a4a"
        ],
    }
}
```

AwsRds resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsRds` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsRdsDbCluster

The `AwsRdsDbCluster` object provides details about an Amazon RDS database cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbCluster` object. To view descriptions of `AwsRdsDbCluster` attributes, see AwsRdsDbClusterDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsRdsDbCluster": {
    "ActivityStreamStatus": "stopped",
    "AllocatedStorage": 1,
    "AssociatedRoles": [
        {
        "RoleArn": "arn:aws:iam::777788889999:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
        "Status": "PENDING"
        }
    ],
    "AutoMinorVersionUpgrade": true,
    "AvailabilityZones": [
        "us-east-1a",
        "us-east-1c",
```

```
        "us-east-1e"
    ],
    "BackupRetentionPeriod": 1,
    "ClusterCreateTime": "2020-06-22T17:40:12.322Z",
    "CopyTagsToSnapshot": true,
    "CrossAccountClone": false,
    "CustomEndpoints": [],
    "DatabaseName": "Sample name",
    "DbClusterIdentifier": "database-3",
    "DbClusterMembers": [
        {
        "DbClusterParameterGroupStatus": "in-sync",
        "DbInstanceIdentifier": "database-3-instance-1",
        "IsClusterWriter": true,
        "PromotionTier": 1,
        }
    ],
    "DbClusterOptionGroupMemberships": [],
    "DbClusterParameterGroup": "cluster-parameter-group",
    "DbClusterResourceId": "cluster-example",
    "DbSubnetGroup": "subnet-group",
    "DeletionProtection": false,
    "DomainMemberships": [],
    "Status": "modifying",
    "EnabledCloudwatchLogsExports": [
        "audit",
        "error",
        "general",
        "slowquery"
    ],
    "Endpoint": "database-3.cluster-example.us-east-1.rds.amazonaws.com",
    "Engine": "aurora-mysql",
    "EngineMode": "provisioned",
    "EngineVersion": "5.7.mysql_aurora.2.03.4",
    "HostedZoneId": "ZONE1",
    "HttpEndpointEnabled": false,
    "IamDatabaseAuthenticationEnabled": false,
    "KmsKeyId": "arn:aws:kms:us-east-1:777788889999:key/key1",
    "MasterUsername": "admin",
    "MultiAz": false,
    "Port": 3306,
    "PreferredBackupWindow": "04:52-05:22",
    "PreferredMaintenanceWindow": "sun:09:32-sun:10:02",
    "ReaderEndpoint": "database-3.cluster-ro-example.us-east-1.rds.amazonaws.com",
```

```
    "ReadReplicaIdentifiers": [],
    "Status": "Modifying",
    "StorageEncrypted": true,
    "VpcSecurityGroups": [
        {
            "Status": "active",
            "VpcSecurityGroupId": "sg-example-1"
        }
    ],
}
```

AwsRdsDbClusterSnapshot

The `AwsRdsDbClusterSnapshot` object contains information about an Amazon RDS DB cluster snapshot.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbClusterSnapshot` object. To view descriptions of `AwsRdsDbClusterSnapshot` attributes, see AwsRdsDbClusterSnapshotDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsRdsDbClusterSnapshot": {
    "AllocatedStorage": 0,
    "AvailabilityZones": [
        "us-east-1a",
        "us-east-1d",
        "us-east-1e"
    ],
    "ClusterCreateTime": "2020-06-12T13:23:15.577Z",
    "DbClusterIdentifier": "database-2",
    "DbClusterSnapshotAttributes": [{
        "AttributeName": "restore",
        "AttributeValues": ["123456789012"]
    }],
    "DbClusterSnapshotIdentifier": "rds:database-2-2020-06-23-03-52",
    "Engine": "aurora",
    "EngineVersion": "5.6.10a",
    "IamDatabaseAuthenticationEnabled": false,
    "KmsKeyId": "arn:aws:kms:us-east-1:777788889999:key/key1",
    "LicenseModel": "aurora",
    "MasterUsername": "admin",
    "PercentProgress": 100,
```

```
    "Port": 0,
    "SnapshotCreateTime": "2020-06-22T17:40:12.322Z",
    "SnapshotType": "automated",
    "Status": "available",
    "StorageEncrypted": true,
    "VpcId": "vpc-faf7e380"
}
```

AwsRdsDbInstance

The `AwsRdsDbInstance` object provides details about an Amazon RDS DB instance.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbInstance` object. To view descriptions of `AwsRdsDbInstance` attributes, see AwsRdsDbInstanceDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsRdsDbInstance": {
    "AllocatedStorage": 20,
    "AssociatedRoles": [],
    "AutoMinorVersionUpgrade": true,
    "AvailabilityZone": "us-east-1d",
    "BackupRetentionPeriod": 7,
    "CaCertificateIdentifier": "certificate1",
    "CharacterSetName": "",
    "CopyTagsToSnapshot": true,
    "DbClusterIdentifier": "",
    "DbInstanceArn": "arn:aws:rds:us-east-1:111122223333:db:database-1",
    "DbInstanceClass": "db.t2.micro",
    "DbInstanceIdentifier": "database-1",
    "DbInstancePort": 0,
    "DbInstanceStatus": "available",
    "DbiResourceId": "db-EXAMPLE123",
    "DbName": "",
    "DbParameterGroups": [
        {
            "DbParameterGroupName": "default.mysql5.7",
            "ParameterApplyStatus": "in-sync"
        }
    ],
```

```
    "DbSecurityGroups": [],


    "DbSubnetGroup": {
        "DbSubnetGroupName": "my-group-123abc",
        "DbSubnetGroupDescription": "My subnet group",
        "VpcId": "vpc-example1",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-123abc",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1d"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-456def",
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetStatus": "Active"
            }
      ],
        "DbSubnetGroupArn": ""
    },
    "DeletionProtection": false,
    "DomainMemberships": [],
    "EnabledCloudWatchLogsExports": [],
    "Endpoint": {
        "address": "database-1.example.us-east-1.rds.amazonaws.com",
        "port": 3306,
        "hostedZoneId": "ZONEID1"
    },
    "Engine": "mysql",
    "EngineVersion": "5.7.22",
    "EnhancedMonitoringResourceArn": "arn:aws:logs:us-east-1:111122223333:log-
group:Example:log-stream:db-EXAMPLE1",
    "IamDatabaseAuthenticationEnabled": false,
    "InstanceCreateTime": "2020-06-22T17:40:12.322Z",
    "Iops": "",
    "KmsKeyId": "",
    "LatestRestorableTime": "2020-06-24T05:50:00.000Z",
    "LicenseModel": "general-public-license",
```

```
    "ListenerEndpoint": "",
    "MasterUsername": "admin",
    "MaxAllocatedStorage": 1000,
    "MonitoringInterval": 60,
    "MonitoringRoleArn": "arn:aws:iam::111122223333:role/rds-monitoring-role",
    "MultiAz": false,
    "OptionGroupMemberships": [
        {
            "OptionGroupName": "default:mysql-5-7",
            "Status": "in-sync"
        }
    ],
    "PreferredBackupWindow": "03:57-04:27",
    "PreferredMaintenanceWindow": "thu:10:13-thu:10:43",
    "PendingModifiedValues": {
        "DbInstanceClass": "",
        "AllocatedStorage": "",
        "MasterUserPassword": "",
        "Port": "",
        "BackupRetentionPeriod": "",
        "MultiAZ": "",
        "EngineVersion": "",
        "LicenseModel": "",
        "Iops": "",
        "DbInstanceIdentifier": "",
        "StorageType": "",
        "CaCertificateIdentifier": "",
        "DbSubnetGroupName": "",
        "PendingCloudWatchLogsExports": "",
        "ProcessorFeatures": []
    },
    "PerformanceInsightsEnabled": false,
    "PerformanceInsightsKmsKeyId": "",
    "PerformanceInsightsRetentionPeriod": "",
    "ProcessorFeatures": [],
    "PromotionTier": "",
    "PubliclyAccessible": false,
    "ReadReplicaDBClusterIdentifiers": [],
    "ReadReplicaDBInstanceIdentifiers": [],
    "ReadReplicaSourceDBInstanceIdentifier": "",
    "SecondaryAvailabilityZone": "",
    "StatusInfos": [],
    "StorageEncrypted": false,
    "StorageType": "gp2",
```

```
    "TdeCredentialArn": "",
    "Timezone": "",
    "VpcSecurityGroups": [
        {
            "VpcSecurityGroupId": "sg-example1",
            "Status": "active"
        }
    ]
}
```

AwsRdsDbSecurityGroup

The `AwsRdsDbSecurityGroup` object contains information about an Amazon Relational Database Service

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbSecurityGroup` object. To view descriptions of `AwsRdsDbSecurityGroup` attributes, see AwsRdsDbSecurityGroupDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsRdsDbSecurityGroup": {
    "DbSecurityGroupArn": "arn:aws:rds:us-west-1:111122223333:secgrp:default",
    "DbSecurityGroupDescription": "default",
    "DbSecurityGroupName": "mysecgroup",
    "Ec2SecurityGroups": [
        {
          "Ec2SecurityGroupuId": "myec2group",
          "Ec2SecurityGroupName": "default",
          "Ec2SecurityGroupOwnerId": "987654321021",
          "Status": "authorizing"
        }
    ],
    "IpRanges": [
        {
          "Cidrip": "0.0.0.0/0",
          "Status": "authorizing"
        }
    ],
    "OwnerId": "123456789012",
    "VpcId": "vpc-1234567f"
}
```

AwsRdsDbSnapshot

The `AwsRdsDbSnapshot` object contains details about an Amazon RDS DB cluster snapshot.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbSnapshot` object. To view descriptions of `AwsRdsDbSnapshot` attributes, see AwsRdsDbSnapshotDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsRdsDbSnapshot": {
    "DbSnapshotIdentifier": "rds:database-1-2020-06-22-17-41",
    "DbInstanceIdentifier": "database-1",
    "SnapshotCreateTime": "2020-06-22T17:41:29.967Z",
    "Engine": "mysql",
    "AllocatedStorage": 20,
    "Status": "available",
    "Port": 3306,
    "AvailabilityZone": "us-east-1d",
    "VpcId": "vpc-example1",
    "InstanceCreateTime": "2020-06-22T17:40:12.322Z",
    "MasterUsername": "admin",
    "EngineVersion": "5.7.22",
    "LicenseModel": "general-public-license",
    "SnapshotType": "automated",
    "Iops": null,
    "OptionGroupName": "default:mysql-5-7",
    "PercentProgress": 100,
    "SourceRegion": null,
    "SourceDbSnapshotIdentifier": "",
    "StorageType": "gp2",
    "TdeCredentialArn": "",
    "Encrypted": false,
    "KmsKeyId": "",
    "Timezone": "",
    "IamDatabaseAuthenticationEnabled": false,
    "ProcessorFeatures": [],
    "DbiResourceId": "db-resourceexample1"
}
```

AwsRdsEventSubscription

The `AwsRdsEventSubscription` contains details about an RDS event notification subscription. The subscription allows RDS to post events to an SNS topic.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsEventSubscription` object. To view descriptions of `AwsRdsEventSubscription` attributes, see AwsRdsEventSubscriptionDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsRdsEventSubscription": {
    "CustSubscriptionId": "myawsuser-secgrp",
    "CustomerAwsId": "111111111111",
    "Enabled": true,
    "EventCategoriesList": [
        "configuration change",
        "failure"
    ],
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:111111111111:es:my-instance-events",
    "SnsTopicArn": "arn:aws:sns:us-east-1:111111111111:myawsuser-RDS",
    "SourceIdsList": [
        "si-sample",
        "mysqldb-rr"
    ],
    "SourceType": "db-security-group",
    "Status": "creating",
    "SubscriptionCreationTime": "2021-06-27T01:38:01.090Z"
}
```

AwsRedshift resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsRedshift` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsRedshiftCluster

The `AwsRedshiftCluster` object contains details about an Amazon Redshift cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRedshiftCluster` object. To view descriptions of `AwsRedshiftCluster` attributes, see AwsRedshiftClusterDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsRedshiftCluster": {
    "AllowVersionUpgrade": true,
    "AutomatedSnapshotRetentionPeriod": 1,
    "AvailabilityZone": "us-west-2d",
    "ClusterAvailabilityStatus": "Unavailable",
    "ClusterCreateTime": "2020-08-03T19:22:44.637Z",
    "ClusterIdentifier": "redshift-cluster-1",
    "ClusterNodes": [
        {
            "NodeRole": "LEADER",
            "PrivateIPAddress": "192.0.2.108",
            "PublicIPAddress": "198.51.100.29"
        },
        {
            "NodeRole": "COMPUTE-0",
            "PrivateIPAddress": "192.0.2.22",
            "PublicIPAddress": "198.51.100.63"
        },
        {
             "NodeRole": "COMPUTE-1",
             "PrivateIPAddress": "192.0.2.224",
             "PublicIPAddress": "198.51.100.226"
        }
        ],
    "ClusterParameterGroups": [
        {
            "ClusterParameterStatusList": [
                {
                    "ParameterName": "max_concurrency_scaling_clusters",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
                {
                    "ParameterName": "enable_user_activity_logging",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
```

```
                {
                    "ParameterName": "auto_analyze",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
                {
                    "ParameterName": "query_group",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
                {
                    "ParameterName": "datestyle",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
                {
                    "ParameterName": "extra_float_digits",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
                {
                    "ParameterName": "search_path",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
                {
                    "ParameterName": "statement_timeout",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
                {
                    "ParameterName": "wlm_json_configuration",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
                {
                    "ParameterName": "require_ssl",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
                },
                {
                    "ParameterName": "use_fips_ssl",
                    "ParameterApplyStatus": "in-sync",
                    "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
```

```
                }
            ],
            "ParameterApplyStatus": "in-sync",
            "ParameterGroupName": "temp"
        }
    ],
    "ClusterPublicKey": "JalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY Amazon-Redshift",
    "ClusterRevisionNumber": 17498,
    "ClusterSecurityGroups": [
        {
            "ClusterSecurityGroupName": "default",
            "Status": "active"
        }
    ],
    "ClusterSnapshotCopyStatus": {
        "DestinationRegion": "us-west-2",
        "ManualSnapshotRetentionPeriod": -1,
        "RetentionPeriod": 1,
        "SnapshotCopyGrantName": "snapshotCopyGrantName"
    },
    "ClusterStatus": "available",
    "ClusterSubnetGroupName": "default",
    "ClusterVersion": "1.0",
    "DBName": "dev",
    "DeferredMaintenanceWindows": [
        {
            "DeferMaintenanceEndTime": "2020-10-07T20:34:01.000Z",
            "DeferMaintenanceIdentifier": "deferMaintenanceIdentifier",
            "DeferMaintenanceStartTime": "2020-09-07T20:34:01.000Z"
        }
     ],
    "ElasticIpStatus": {
        "ElasticIp": "203.0.113.29",
        "Status": "active"
    },
    "ElasticResizeNumberOfNodeOptions": "4",
    "Encrypted": false,
    "Endpoint": {
        "Address": "redshift-cluster-1.example.us-west-2.redshift.amazonaws.com",
        "Port": 5439
    },
    "EnhancedVpcRouting": false,
    "ExpectedNextSnapshotScheduleTime": "2020-10-13T20:34:01.000Z",
    "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
```

```
    "HsmStatus": {
        "HsmClientCertificateIdentifier": "hsmClientCertificateIdentifier",
        "HsmConfigurationIdentifier": "hsmConfigurationIdentifier",
        "Status": "applying"
    },
    "IamRoles": [
        {
             "ApplyStatus": "in-sync",
             "IamRoleArn": "arn:aws:iam::111122223333:role/RedshiftCopyUnload"
        }
    ],
    "KmsKeyId": "kmsKeyId",
    "LoggingStatus": {
        "BucketName": "amzn-s3-demo-bucket",
        "LastFailureMessage": "test message",
        "LastFailureTime": "2020-08-09T13:00:00.000Z",
        "LastSuccessfulDeliveryTime": "2020-08-08T13:00:00.000Z",
        "LoggingEnabled": true,
        "S3KeyPrefix": "/"
    },
    "MaintenanceTrackName": "current",
    "ManualSnapshotRetentionPeriod": -1,
    "MasterUsername": "awsuser",
    "NextMaintenanceWindowStartTime": "2020-08-09T13:00:00.000Z",
    "NodeType": "dc2.large",
    "NumberOfNodes": 2,
    "PendingActions": [],
    "PendingModifiedValues": {
        "AutomatedSnapshotRetentionPeriod": 0,
        "ClusterIdentifier": "clusterIdentifier",
        "ClusterType": "clusterType",
        "ClusterVersion": "clusterVersion",
        "EncryptionType": "None",
        "EnhancedVpcRouting": false,
        "MaintenanceTrackName": "maintenanceTrackName",
        "MasterUserPassword": "masterUserPassword",
        "NodeType": "dc2.large",
        "NumberOfNodes": 1,
        "PubliclyAccessible": true
    },
    "PreferredMaintenanceWindow": "sun:13:00-sun:13:30",
    "PubliclyAccessible": true,
    "ResizeInfo": {
        "AllowCancelResize": true,
```

```
        "ResizeType": "ClassicResize"
    },
    "RestoreStatus": {
        "CurrentRestoreRateInMegaBytesPerSecond": 15,
        "ElapsedTimeInSeconds": 120,
        "EstimatedTimeToCompletionInSeconds": 100,
        "ProgressInMegaBytes": 10,
        "SnapshotSizeInMegaBytes": 1500,
        "Status": "restoring"
    },
    "SnapshotScheduleIdentifier": "snapshotScheduleIdentifier",
    "SnapshotScheduleState": "ACTIVE",
     "VpcId": "vpc-example",
    "VpcSecurityGroups": [
        {
            "Status": "active",
            "VpcSecurityGroupId": "sg-example"
        }
    ]
}
```

AwsRoute53 resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsRoute53` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsRoute53HostedZone

The `AwsRoute53HostedZone` object provides information about an Amazon Route 53 hosted zone, including the four name servers assigned to the hosted zone. A hosted zone represents a collection of records that can be managed together, belonging to a single parent domain name.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRoute53HostedZone` object. To view descriptions of `AwsRoute53HostedZone` attributes, see AwsRoute53HostedZoneDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsRoute53HostedZone": {
```

```
    "HostedZone": {
        "Id": "Z06419652JEMGO9TA2XKL",
        "Name": "asff.testing",
        "Config": {
            "Comment": "This is an example comment."
        }
    },
    "NameServers": [
        "ns-470.awsdns-32.net",
        "ns-1220.awsdns-12.org",
        "ns-205.awsdns-13.com",
        "ns-1960.awsdns-51.co.uk"
    ],
    "QueryLoggingConfig": {
        "CloudWatchLogsLogGroupArn": {
            "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:123456789012:log-
group:asfftesting:*",
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "HostedZoneId": "Z00932193AF5H180PPNZD"
        }
    },
    "Vpcs": [
        {
            "Id": "vpc-05d7c6e36bc03ea76",
            "Region": "us-east-1"
        }
    ]
}
```

AwsS3 resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsS3` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsS3AccessPoint

`AwsS3AccessPoint` provides information about an Amazon S3 access point. S3 access points are named network endpoints that are attached to S3 buckets that you can use to perform S3 object operations.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsS3AccessPoint` object. To view descriptions of `AwsS3AccessPoint` attributes, see AwsS3AccessPointDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsS3AccessPoint": {
        "AccessPointArn": "arn:aws:s3:us-east-1:123456789012:accesspoint/asff-access-
point",
        "Alias": "asff-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias",
        "Bucket": "amzn-s3-demo-bucket",
        "BucketAccountId": "123456789012",
        "Name": "asff-access-point",
        "NetworkOrigin": "VPC",
        "PublicAccessBlockConfiguration": {
            "BlockPublicAcls": true,
            "BlockPublicPolicy": true,
            "IgnorePublicAcls": true,
            "RestrictPublicBuckets": true
        },
        "VpcConfiguration": {
            "VpcId": "vpc-1a2b3c4d5e6f1a2b3"
        }
}
```

AwsS3AccountPublicAccessBlock

`AwsS3AccountPublicAccessBlock` provides information about the Amazon S3 Public Access Block configuration for accounts.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsS3AccountPublicAccessBlock` object. To view descriptions of `AwsS3AccountPublicAccessBlock` attributes, see AwsS3AccountPublicAccessBlockDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsS3AccountPublicAccessBlock": {
    "BlockPublicAcls": true,
    "BlockPublicPolicy": true,
    "IgnorePublicAcls": false,
    "RestrictPublicBuckets": true
```

```
}
```

AwsS3Bucket

The `AwsS3Bucket` object provides details about an Amazon S3 bucket.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsS3Bucket` object. To view descriptions of `AwsS3Bucket` attributes, see AwsS3BucketDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsS3Bucket": {
    "AccessControlList": "{\"grantSet\":null,\"grantList\":[{\"grantee\":{\"id\":
\"4df55416215956920d9d056aa8b99803a294ea221222bb668b55a8c6bca81094\",\"displayName
\":null},\"permission\":\"FullControl\"},{\"grantee\":\"AllUsers\",\"permission\":
\"ReadAcp\"},{\"grantee\":\"AuthenticatedUsers\",\"permission\":\"ReadAcp\"}",,
    "BucketLifecycleConfiguration": {
       "Rules": [
           {
               "AbortIncompleteMultipartUpload": {
                   "DaysAfterInitiation": 5
               },
               "ExpirationDate": "2021-11-10T00:00:00.000Z",
               "ExpirationInDays": 365,
               "ExpiredObjectDeleteMarker": false,
               "Filter": {
                   "Predicate": {
                       "Operands": [
                           {
                               "Prefix": "tmp/",
                               "Type": "LifecyclePrefixPredicate"
                           },
                           {
                               "Tag": {
                                   "Key": "ArchiveAge",
                                   "Value": "9m"
                               },
                               "Type": "LifecycleTagPredicate"
                           }
                       ],
                       "Type": "LifecycleAndOperator"
                   }
```

```
            },
            "ID": "Move rotated logs to Glacier",
            "NoncurrentVersionExpirationInDays": -1,
            "NoncurrentVersionTransitions": [
                {
                    "Days": 2,
                    "StorageClass": "GLACIER"
                }
            ],
            "Prefix": "rotated/",
            "Status": "Enabled",
            "Transitions": [
                {
                    "Date": "2020-11-10T00:00:00.000Z",
                    "Days": 100,
                    "StorageClass": "GLACIER"
                }
            ]
        }
    ]
},
"BucketLoggingConfiguration": {
 "DestinationBucketName": "s3serversideloggingbucket-123456789012",
 "LogFilePrefix": "buckettestreadwrite23435/"
},
"BucketName": "amzn-s3-demo-bucket",
"BucketNotificationConfiguration": {
 "Configurations": [{
  "Destination": "arn:aws:lambda:us-east-1:123456789012:function:s3_public_write",
  "Events": [
   "s3:ObjectCreated:Put"
  ],
  "Filter": {
   "S3KeyFilter": {
    "FilterRules": [
    {
     "Name": "AffS3BucketNotificationConfigurationS3KeyFilterRuleName.PREFIX",
     "Value": "pre"
    },
    {
     "Name": "AffS3BucketNotificationConfigurationS3KeyFilterRuleName.SUFFIX",
     "Value": "suf"
    },
    ]
```

```
        }
       },
       "Type": "LambdaConfiguration"
      }]
     },
     "BucketVersioningConfiguration": {
      "IsMfaDeleteEnabled": true,
      "Status": "Off"
     },
     "BucketWebsiteConfiguration": {
      "ErrorDocument": "error.html",
      "IndexDocumentSuffix": "index.html",
      "RedirectAllRequestsTo": {
       "HostName": "example.com",
       "Protocol": "http"
      },
      "RoutingRules": [{
       "Condition": {
        "HttpErrorCodeReturnedEquals": "Redirected",
        "KeyPrefixEquals": "index"
          },
       "Redirect": {
        "HostName": "example.com",
        "HttpRedirectCode": "401",
        "Protocol": "HTTP",
        "ReplaceKeyPrefixWith": "string",
        "ReplaceKeyWith": "string"
       }
      }]
     },
     "CreatedAt": "2007-11-30T01:46:56.000Z",
     "ObjectLockConfiguration": {
      "ObjectLockEnabled": "Enabled",
      "Rule": {
       "DefaultRetention": {
        "Days": null,
        "Mode": "GOVERNANCE",
        "Years": 12
       },
      },
     },
     "OwnerId": "AIDACKCEVSQ6C2EXAMPLE",
     "OwnerName": "s3bucketowner",
     "PublicAccessBlockConfiguration": {
```

```
        "BlockPublicAcls": true,
        "BlockPublicPolicy": true,
        "IgnorePublicAcls": true,
        "RestrictPublicBuckets": true,
    },
    "ServerSideEncryptionConfiguration": {
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "AES256",
                    "KMSMasterKeyID": "12345678-abcd-abcd-abcd-123456789012"
                }
            }
        ]
     }
}
```

AwsS3Object

The `AwsS3Object` object provides information about an Amazon S3 object.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsS3Object` object. To view descriptions of `AwsS3Object` attributes, see AwsS3ObjectDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsS3Object": {
    "ContentType": "text/html",
    "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
    "LastModified": "2012-04-23T18:25:43.511Z",
    "ServerSideEncryption": "aws:kms",
    "SSEKMSKeyId": "arn:aws:kms:us-west-2:123456789012:key/4dff8393-e225-4793-
a9a0-608ec069e5a7",
    "VersionId": "ws31OurgOOjH_HHllIxPE35P.MELYaYh"
}
```

AwsSageMaker resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsSageMaker` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsSageMakerNotebookInstance

The `AwsSageMakerNotebookInstance` object provides information about a Amazon SageMaker AI notebook instance, which is a machine learning compute instance running the Jupyter Notebook App.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsSageMakerNotebookInstance` object. To view descriptions of `AwsSageMakerNotebookInstance` attributes, see AwsSageMakerNotebookInstanceDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsSageMakerNotebookInstance": {
    "DirectInternetAccess": "Disabled",
    "InstanceMetadataServiceConfiguration": {
     "MinimumInstanceMetadataServiceVersion": "1",
    },
    "InstanceType": "ml.t2.medium",
    "LastModifiedTime": "2022-09-09 22:48:32.012000+00:00",
    "NetworkInterfaceId": "eni-06c09ac2541a1bed3",
    "NotebookInstanceArn": "arn:aws:sagemaker:us-east-1:001098605940:notebook-instance/
sagemakernotebookinstancerootaccessdisabledcomplia-8myjcyofzixm",
    "NotebookInstanceName":
 "SagemakerNotebookInstanceRootAccessDisabledComplia-8MYjcyofZiXm",
    "NotebookInstanceStatus": "InService",
    "PlatformIdentifier": "notebook-al1-v1",
    "RoleArn": "arn:aws:iam::001098605940:role/sechub-SageMaker-1-scenar-
SageMakerCustomExecution-1R0X32HGC38IW",
    "RootAccess": "Disabled",
    "SecurityGroups": [
     "sg-06b347359ab068745"
    ],
    "SubnetId": "subnet-02c0deea5fa64578e",
    "Url":
 "sagemakernotebookinstancerootaccessdisabledcomplia-8myjcyofzixm.notebook.us-
east-1.sagemaker.aws",
    "VolumeSizeInGB": 5
}
```

AwsSecretsManager resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsSecretsManager` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsSecretsManagerSecret

The `AwsSecretsManagerSecret` object provides details about a Secrets Manager secret.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsSecretsManagerSecret` object. To view descriptions of `AwsSecretsManagerSecret` attributes, see AwsSecretsManagerSecretDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsSecretsManagerSecret": {
    "RotationRules": {
        "AutomaticallyAfterDays": 30
    },
    "RotationOccurredWithinFrequency": true,
    "KmsKeyId": "kmsKeyId",
    "RotationEnabled": true,
    "RotationLambdaArn": "arn:aws:lambda:us-
west-2:777788889999:function:MyTestRotationLambda",
    "Deleted": false,
    "Name": "MyTestDatabaseSecret",
    "Description": "My test database secret"
}
```

AwsSns resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsSns` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsSnsTopic

The `AwsSnsTopic` object contains details about an Amazon Simple Notification Service topic.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsSnsTopic` object. To view descriptions of `AwsSnsTopic` attributes, see AwsSnsTopicDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsSnsTopic": {
    "ApplicationSuccessFeedbackRoleArn": "arn:aws:iam::123456789012:role/
ApplicationSuccessFeedbackRoleArn",
    "FirehoseFailureFeedbackRoleArn": "arn:aws:iam::123456789012:role/
FirehoseFailureFeedbackRoleArn",
    "FirehoseSuccessFeedbackRoleArn": "arn:aws:iam::123456789012:role/
FirehoseSuccessFeedbackRoleArn",
    "HttpFailureFeedbackRoleArn": "arn:aws:iam::123456789012:role/
HttpFailureFeedbackRoleArn",
    "HttpSuccessFeedbackRoleArn": "arn:aws:iam::123456789012:role/
HttpSuccessFeedbackRoleArn",
    "KmsMasterKeyId": "alias/ExampleAlias",
    "Owner": "123456789012",
    "SqsFailureFeedbackRoleArn": "arn:aws:iam::123456789012:role/
SqsFailureFeedbackRoleArn",
    "SqsSuccessFeedbackRoleArn": "arn:aws:iam::123456789012:role/
SqsSuccessFeedbackRoleArn",
    "Subscription": {
         "Endpoint": "http://sampleendpoint.com",
         "Protocol": "http"
    },
    "TopicName": "SampleTopic"
}
```

AwsSqs resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsSqs` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsSqsQueue

The `AwsSqsQueue` object contains information about an Amazon Simple Queue Service queue.

The following example shows the AWS Security Finding Format (ASFF) for the AwsSqsQueue object. To view descriptions of AwsSqsQueue attributes, see AwsSqsQueueDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsSqsQueue": {
    "DeadLetterTargetArn": "arn:aws:sqs:us-west-2:123456789012:queue/target",
    "KmsDataKeyReusePeriodSeconds": 60,,
    "KmsMasterKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "QueueName": "sample-queue"
}
```

AwsSsm resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for AwsSsm resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsSsmPatchCompliance

The AwsSsmPatchCompliance object provides information about the state of a patch on an instance based on the patch baseline that was used to patch the instance.

The following example shows the AWS Security Finding Format (ASFF) for the AwsSsmPatchCompliance object. To view descriptions of AwsSsmPatchCompliance attributes, see AwsSsmPatchComplianceDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsSsmPatchCompliance": {
    "Patch": {
        "ComplianceSummary": {
            "ComplianceType": "Patch",
            "CompliantCriticalCount": 0,
            "CompliantHighCount": 0,
            "CompliantInformationalCount": 0,
            "CompliantLowCount": 0,
            "CompliantMediumCount": 0,
            "CompliantUnspecifiedCount": 461,
            "ExecutionType": "Command",
```

```
            "NonCompliantCriticalCount": 0,
            "NonCompliantHighCount": 0,
            "NonCompliantInformationalCount": 0,
            "NonCompliantLowCount": 0,
            "NonCompliantMediumCount": 0,
            "NonCompliantUnspecifiedCount": 0,
            "OverallSeverity": "UNSPECIFIED",
            "PatchBaselineId": "pb-0c5b2769ef7cbe587",
            "PatchGroup": "ExamplePatchGroup",
            "Status": "COMPLIANT"
        }
    }
}
```

AwsStepFunctions resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsStepFunctions` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsStepFunctionStateMachine

The `AwsStepFunctionStateMachine` object provides information about an AWS Step Functions state machine, which is a workflow consisting of a series of event-driven steps.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsStepFunctionStateMachine` object. To view descriptions of `AwsStepFunctionStateMachine` attributes, see AwsStepFunctionStateMachine in the *AWS Security Hub API Reference*.

Example

```
"AwsStepFunctionStateMachine": {
    "StateMachineArn": "arn:aws:states:us-
east-1:123456789012:stateMachine:StepFunctionsLogDisableNonCompliantResource-
fQLujTeXvwsb",
    "Name": "StepFunctionsLogDisableNonCompliantResource-fQLujTeXvwsb",
    "Status": "ACTIVE",
    "RoleArn": "arn:aws:iam::123456789012:role/teststepfunc-
StatesExecutionRole-1PNM71RVO1UKT",
    "Type": "STANDARD",
```

```
    "LoggingConfiguration": {
        "Level": "OFF",
        "IncludeExecutionData": false
    },
    "TracingConfiguration": {
        "Enabled": false
    }
}
```

AwsWaf resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsWaf` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsWafRateBasedRule

The `AwsWafRateBasedRule` object contains details about an AWS WAF rate-based rule for global resources. An AWS WAF rate-based rule provides settings to indicate when to allow, block, or count a request. Rate-based rules include the number of requests that arrive over a specified period of time.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafRateBasedRule` object. To view descriptions of `AwsWafRateBasedRule` attributes, see AwsWafRateBasedRuleDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRateBasedRule":{
    "MatchPredicates" : [{
        "DataId" : "391b7a7e-5f00-40d2-b114-3f27ceacbbb0",
        "Negated" : "True",
        "Type" : "IPMatch" ,
    }],
    "MetricName" : "MetricName",
    "Name" : "Test",
    "RateKey" : "IP",
    "RateLimit" : 235000,
    "RuleId" : "5dfb4085-f103-4ec6-b39a-d4a0dae5f47f"
}
```

AwsWafRegionalRateBasedRule

The `AwsWafRegionalRateBasedRule` object contains details about a rate-based rule for Regional resources. A rate-based rule provides settings to indicate when to allow, block, or count a request. Rate-based rules include the number of requests that arrive over a specified period of time.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafRegionalRateBasedRule` object. To view descriptions of `AwsWafRegionalRateBasedRule` attributes, see AwsWafRegionalRateBasedRuleDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRegionalRateBasedRule":{
    "MatchPredicates" : [{
        "DataId" : "391b7a7e-5f00-40d2-b114-3f27ceacbbb0",
        "Negated" : "True",
        "Type" : "IPMatch" ,
    }],
    "MetricName" : "MetricName",
    "Name" : "Test",
    "RateKey" : "IP",
    "RateLimit" : 235000,
    "RuleId" : "5dfb4085-f103-4ec6-b39a-d4a0dae5f47f"
}
```

AwsWafRegionalRule

The `AwsWafRegionalRule` object provides details about an AWS WAF Regional rule . This rule identifies the web requests that you want to allow, block, or count.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafRegionalRule` object. To view descriptions of `AwsWafRegionalRule` attributes, see AwsWafRegionalRuleDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRegionalRule": {
    "MetricName": "SampleWAF_Rule__Metric_1",
    "Name": "bb-waf-regional-rule-not-empty-conditions-compliant",
    "RuleId": "8f651760-24fa-40a6-a9ed-4b60f1de95fe",
```

```
    "PredicateList": [{
        "DataId": "127d9346-e607-4e93-9286-c1296fb5445a",
        "Negated": false,
        "Type": "GeoMatch"
    }]
}
```

AwsWafRegionalRuleGroup

The `AwsWafRegionalRuleGroup` object provides details about an AWS WAF Regional rule group. A rule group is a collection of predefined rules that you add to a web access control list (web ACL).

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafRegionalRuleGroup` object. To view descriptions of `AwsWafRegionalRuleGroup` attributes, see AwsWafRegionalRuleGroupDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRegionalRuleGroup": {
    "MetricName": "SampleWAF_Metric_1",
    "Name": "bb-WAFClassicRuleGroupWithRuleCompliant",
    "RuleGroupId": "2012ca6d-e66d-4d9b-b766-bfb03ad77cfb",
    "Rules": [{
        "Action": {
            "Type": "ALLOW"
        }
    }],
        "Priority": 1,
        "RuleId": "cdd225da-32cf-4773-8dc5-3bca3ed9c19c",
        "Type": "REGULAR"
}
```

AwsWafRegionalWebAcl

`AwsWafRegionalWebAcl` provides details about an AWS WAF Regional web access control list (web ACL). A web ACL contains the rules that identify the requests that you want to allow, block, or count.

The following is an example `AwsWafRegionalWebAcl` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayV2Stage` attributes, see AwsWafRegionalWebAclDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRegionalWebAcl": {
    "DefaultAction": "ALLOW",
    "MetricName" : "web-regional-webacl-metric-1",
    "Name": "WebACL_123",
    "RulesList": [
        {
            "Action": {
                "Type": "Block"
            },
            "Priority": 3,
            "RuleId": "24445857-852b-4d47-bd9c-61f05e4d223c",
            "Type": "REGULAR",
            "ExcludedRules": [
                {
                    "ExclusionType": "Exclusion",
                    "RuleId": "Rule_id_1"
                }
            ],
            "OverrideAction": {
                "Type": "OVERRIDE"
            }
        }
    ],
    "WebAclId": "443c76f4-2e72-4c89-a2ee-389d501c1f67"
}
```

AwsWafRule

`AwsWafRule` provides information about an AWS WAF rule. An AWS WAF rule identifies the web requests that you want to allow, block, or count.

The following is an example `AwsWafRule` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayV2Stage` attributes, see AwsWafRuleDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRule": {
    "MetricName": "AwsWafRule_Metric_1",
    "Name": "AwsWafRule_Name_1",
```

```
    "PredicateList": [{
        "DataId": "cdd225da-32cf-4773-1dc2-3bca3ed9c19c",
        "Negated": false,
        "Type": "GeoMatch"
    }],
    "RuleId": "8f651760-24fa-40a6-a9ed-4b60f1de953e"
}
```

AwsWafRuleGroup

`AwsWafRuleGroup` provides information about an AWS WAF rule group. An AWS WAF rule group is a collection of predefined rules that you add to a web access control list (web ACL).

The following is an example `AwsWafRuleGroup` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayV2Stage` attributes, see AwsWafRuleGroupDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRuleGroup": {
    "MetricName": "SampleWAF_Metric_1",
    "Name": "bb-WAFRuleGroupWithRuleCompliant",
    "RuleGroupId": "2012ca6d-e66d-4d9b-b766-bfb03ad77cfb",
    "Rules": [{
        "Action": {
            "Type": "ALLOW",
        },
        "Priority": 1,
        "RuleId": "cdd225da-32cf-4773-8dc5-3bca3ed9c19c",
        "Type": "REGULAR"
    }]
}
```

AwsWafv2RuleGroup

The `AwsWafv2RuleGroup` object provides details about an AWS WAFV2 rule group.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafv2RuleGroup` object. To view descriptions of `AwsWafv2RuleGroup` attributes, see AwsWafv2RuleGroupDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafv2RuleGroup": {
    "Arn": "arn:aws:wafv2:us-east-1:123456789012:global/rulegroup/wafv2rulegroupasff/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Capacity": 1000,
    "Description": "Resource for ASFF",
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "wafv2rulegroupasff",
    "Rules": [{
     "Action": {
     "Allow": {
      "CustomRequestHandling": {
       "InsertHeaders": [
        {
        "Name": "AllowActionHeader1Name",
        "Value": "AllowActionHeader1Value"
        },
        {
        "Name": "AllowActionHeader2Name",
        "Value": "AllowActionHeader2Value"
        }
       ]
      }
     },
     "Name": "RuleOne",
     "Priority": 1,
     "VisibilityConfig": {
      "CloudWatchMetricsEnabled": true,
      "MetricName": "rulegroupasff",
      "SampledRequestsEnabled": false
     }
    }],
    "VisibilityConfig": {
     "CloudWatchMetricsEnabled": true,
     "MetricName": "rulegroupasff",
     "SampledRequestsEnabled": false
    }
}
```

AwsWafWebAcl

The `AwsWafWebAcl` object provides details about an AWS WAF web ACL.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafWebAcl` object. To view descriptions of `AwsWafWebAcl` attributes, see AwsWafWebAclDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafWebAcl": {
    "DefaultAction": "ALLOW",
    "Name": "MyWafAcl",
    "Rules": [
        {
            "Action": {
                "Type": "ALLOW"
            },
            "ExcludedRules": [
                {
                    "RuleId": "5432a230-0113-5b83-bbb2-89375c5bfa98"
                }
            ],
            "OverrideAction": {
                "Type": "NONE"
            },
            "Priority": 1,
            "RuleId": "5432a230-0113-5b83-bbb2-89375c5bfa98",
            "Type": "REGULAR"
        }
    ],
    "WebAclId": "waf-1234567890"
}
```

AwsWafv2WebAcl

The `AwsWafv2WebAcl` object provides details about an AWS WAFV2 web ACL.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafv2WebAcl` object. To view descriptions of `AwsWafv2WebAcl` attributes, see AwsWafv2WebAclDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsWafv2WebAcl": {
    "Arn": "arn:aws:wafv2:us-east-1:123456789012:regional/webacl/WebACL-RoaD4QexqSxG/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
    "Capacity": 1326,
    "CaptchaConfig": {
     "ImmunityTimeProperty": {
      "ImmunityTime": 500
     }
    },
    "DefaultAction": {
     "Block": {}
    },
    "Description": "Web ACL for JsonBody testing",
    "ManagedbyFirewallManager": false,
    "Name": "WebACL-RoaD4QexqSxG",
    "Rules": [{
     "Action": {
      "RuleAction": {
       "Block": {}
      }
     },
     "Name": "TestJsonBodyRule",
     "Priority": 1,
     "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "JsonBodyMatchMetric"
     }
    }],
    "VisibilityConfig": {
     "SampledRequestsEnabled": true,
     "CloudWatchMetricsEnabled": true,
     "MetricName": "TestingJsonBodyMetric"
    }
}
```

AwsXray resources in ASFF

The following are examples of the AWS Security Finding Format (ASFF) syntax for `AwsXray` resources.

AWS Security Hub normalizes findings from various sources into ASFF. For background information about ASFF, see AWS Security Finding Format (ASFF).

AwsXrayEncryptionConfig

The `AwsXrayEncryptionConfig` object contains information about the encryption configuration for AWS X-Ray.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsXrayEncryptionConfig` object. To view descriptions of `AwsXrayEncryptionConfig` attributes, see AwsXrayEncryptionConfigDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsXRayEncryptionConfig":{
    "KeyId": "arn:aws:kms:us-east-2:222222222222:key/example-key",
    "Status": "UPDATING",
    "Type":"KMS"
}
```

CodeRepository object in ASFF

The `CodeRepository` object provides information about an external code repository that you connected to AWS resources and configured Amazon Inspector to scan for vulnerabilities.

The following example shows the AWS Security Finding Format (ASFF) syntax for the `CodeRepository` object. To view descriptions of `CodeRepository` attributes, see CodeRepositoryDetails in the *AWS Security Hub API Reference*. For background information about ASFF, see AWS Security Finding Format (ASFF).

Example

```
"CodeRepository": {
    "ProviderType": "GITLAB_SELF_MANAGED",
    "ProjectName": "projectName",
    "CodeSecurityIntegrationArn": "arn:aws:inspector2:us-
east-1:123456789012:codesecurity-integration/00000000-0000-0000-0000-000000000000"
}
```

Container object in ASFF

The following example shows the AWS Security Finding Format (ASFF) syntax for the `Container` object. To view descriptions of `Container` attributes, see ContainerDetails in the *AWS Security Hub API Reference*. For background information about ASFF, see AWS Security Finding Format (ASFF).

Example

```
"Container": {
    "ContainerRuntime": "docker",
    "ImageId": "image12",
    "ImageName": "1111111/
knotejs@sha256:372131c9fef111111111111115f4ed3ea5f9dce4dc3bd34ce21846588a3",
    "LaunchedAt": "2018-09-29T01:25:54Z",
    "Name": "knote",
    "Privileged": true,
    "VolumeMounts": [{
        "Name": "vol-03909e9",
        "MountPath": "/mnt/etc"
    }]
}
```

Other object in ASFF

In the AWS Security Finding Format (ASFF), the `Other` object specifies custom fields and values. For more information about ASFF, see AWS Security Finding Format (ASFF).

By using the `Other` object, you can specify custom fields and values for a resource. You can use the `Other` object for the following cases:

- The resource type does not have a corresponding `Details` object. To specify details for a resource, use the `Other` object.
- The `Details` object for the resource type does not include all the attributes that you want to specify. In this case, use the `Details` object for the resource type to specify available attributes. Use the `Other` object to specify attributes that are not in the type-specific `Details` object.
- The resource type is not one of the provided types. In this case, set `Resource.Type` to `Other` and use the `Other` object to specify the details.

Type: Map of up to 50 key-value pairs

Each key-value pair must meet the following requirements.

- The key must contain fewer than 128 characters.
- The value must contain fewer than 1,024 characters.

Viewing insights in Security Hub CSPM

In AWS Security Hub CSPM, an *insight* is a collection of related findings. An insight can identify a specific security area that requires attention and intervention. For example, an insight might point out EC2 instances that are the subject of findings that detect poor security practices. An insight brings together findings from across finding providers.

Each insight is defined by a group by statement and optional filters. The group by statement indicates how to group the matching findings, and identifies the type of item that the insight applies to. For example, if an insight is grouped by resource identifier, then the insight produces a list of resource identifiers. The optional filters identify the matching findings for the insight. For example, you might want to only see findings from specific providers or findings that are associated with specific types of resources.

Security Hub CSPM offers several built-in managed insights. You can't modify or delete managed insights. To track security issues that are unique to your AWS environment and usage, you can create custom insights.

The **Insights** page on the AWS Security Hub CSPM console displays the list of available insights.

By default, the list displays both managed and custom insights. To filter the insight list based on insight type, choose the insight type from the dropdown menu that is next to the filter field.

- To display all of the available insights, choose **All insights**. This is the default option.
- To display only managed insights, choose **Security Hub CSPM managed insights**.
- To display only custom insights, choose **Custom insights**.

You also can filter the insight list based on the insight's name. To do so, in the filter field, type the text to use to filter the list. The filter is not case sensitive. The filter looks for insights that contain the text anywhere in the insight name.

An insight only returns results if you have enabled integrations or standards that produce matching findings. For example, the managed insight **29. Top resources by counts of failed CIS checks** only returns results if you enable a version of the Center for Internet Security (CIS) AWS Foundations Benchmark standard.

Reviewing and acting on insights in Security Hub CSPM

For each insight, AWS Security Hub CSPM first determines the findings that match the filter criteria, and then uses the grouping attribute to group the matching findings.

From the **Insights** page on the console, you can view and take action on the results and findings.

If you enable cross-Region aggregation, the results for managed insights (when you're signed in to the aggregation Region) include findings from the aggregation Region and linked Regions. The results for custom insights, if the insight doesn't filter by Region, also include findings from the aggregation Region and linked Regions (when you're signed in to the aggregation Region). In other Regions, the insight results are only for that Region.

For information about configuring cross-Region aggregation, see the section called “Aggregating data across Regions”.

Viewing and taking action on insight results

The insight results consist of a grouped list of the results for the insight. For example, if the insight is grouped by resource identifiers, then the insight results are the list of resource identifiers. Each item in the results list indicates the number of matching findings for that item.

If the findings are grouped by resource identifier or resource type, the results include all of the resources in the matching findings. This includes resources that have a different type from the resource type specified in the filter criteria. For example, an insight identifies findings that are associated with S3 buckets. If a matching finding contains both an S3 bucket resource and an IAM access key resource, the insight results include both resources.

On the Security Hub CSPM console, the results list is sorted from most to fewest matching findings. Security Hub CSPM can only display 100 results. If there are more than 100 grouping values, you only see the first 100.

In addition to the results list, the insight results display a set of charts summarizing the number of matching findings for the following attributes.

- **Severity label** – Number of findings for each severity label
- **AWS account ID** – Top five account IDs for the matching findings
- **Resource type** – Top five resource types for the matching findings

- **Resource ID** – Top five resource IDs for the matching findings
- **Product name** - Top five finding providers for the matching findings

If you have configured custom actions, then you can send selected results to a custom action. The action must be associated with an Amazon CloudWatch rule for the `Security Hub Insight Results` event type. For more information, see the section called “Automated response and remediation”. If you have not configured custom actions, the **Actions** menu is disabled.

Security Hub CSPM console

To view and take action on insight results (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Insights**.
3. To display the list of insight results, choose the insight name.
4. Select the check box for each result to send to the custom action.
5. From the **Actions** menu, choose the custom action.

Security Hub CSPM API, AWS CLI

To view and take action on insight results (API, AWS CLI)

To view insight results, use the >GetInsightResults operation of the Security Hub CSPM API. If you use the AWS CLI, run the get-insight-results command.

To identify the insight to return results for, you need the insight ARN. To obtain the insight ARNs for custom insights, use the `GetInsights` API operation or the get-insight-results command.

The following example retrieves the results for the specified insight. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub get-insight-results --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

For information about how to create custom actions programmatically, see Using custom actions to send findings and insight results to EventBridge.

Viewing and taking action on insight result findings (console)

From an insight results list on the Security Hub CSPM console, you can display the list of findings for each result.

To display and take action on insight findings (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Insights**.
3. To display the list of insight results, choose the insight name.
4. To display the list of findings for an insight result, choose the item from the results list. The findings list shows the active findings for the selected insight result that have a workflow status of `NEW` or `NOTIFIED`.

From the findings list, you can perform the following actions:

- Filtering findings in Security Hub CSPM
- Reviewing finding details and history
- Setting the workflow status of findings in Security Hub CSPM
- Sending findings to a custom Security Hub CSPM action

Managed insights in Security Hub CSPM

AWS Security Hub CSPM provides several managed insights.

You can't edit or delete Security Hub CSPM managed insights. You can view and take action on the insight results and findings. You can also use a managed insight as the basis for a new custom insight.

As with all insights, a managed insight only returns results if you have enabled product integrations or security standards that can produce matching findings.

For insights that are grouped by resource identifier, the results include the identifiers of all of the resources in the matching findings. This includes resources that have a different type from the

resource type in the filter criteria. For example, insight 2 in the following list identifies findings that are associated with Amazon S3 buckets. If a matching finding contains both an S3 bucket resource and an IAM access key resource, the insight results include both resources.

Security Hub CSPM currently offers the following managed insights:

1. AWS resources with the most findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/1`

Grouped by: Resource identifier

Finding filters:

- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

2. S3 buckets with public write or read permissions

ARN: `arn:aws:securityhub:::insight/securityhub/default/10`

Grouped by: Resource identifier

Finding filters:

- Type starts with `Effects/Data Exposure`
- Resource type is `AwsS3Bucket`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

3. AMIs that are generating the most findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/3`

Grouped by: EC2 instance image ID

Finding filters:

- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

4. EC2 instances involved in known Tactics, Techniques, and Procedures (TTPs)

ARN: `arn:aws:securityhub:::insight/securityhub/default/14`

Grouped by: Resource ID

Finding filters:

- Type starts with `TTPs`
- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

5. AWS principals with suspicious access key activity

ARN: `arn:aws:securityhub:::insight/securityhub/default/9`

Grouped by: IAM access key principal name

Finding filters:

- Resource type is `AwsIamAccessKey`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

6. AWS resources instances that don't meet security standards / best practices

ARN: `arn:aws:securityhub:::insight/securityhub/default/6`

Grouped by: Resource ID

Finding filters:

- Type is `Software and Configuration Checks/Industry and Regulatory Standards/AWS Security Best Practices`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

7. AWS resources associated with potential data exfiltration

ARN: `arn:aws:securityhub:::insight/securityhub/default/7`

Grouped by:: Resource ID

Finding filters:

- Type starts with Effects/Data Exfiltration/

- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

8. AWS resources associated with unauthorized resource consumption

ARN: `arn:aws:securityhub:::insight/securityhub/default/8`

Grouped by: Resource ID

Finding filters:

- Type starts with `Effects/Resource Consumption`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

9. S3 buckets that don't meet security standards / best practice

ARN: `arn:aws:securityhub:::insight/securityhub/default/11`

Grouped by: Resource ID

Finding filters:

- Resource type is `AwsS3Bucket`
- Type is `Software and Configuration Checks/Industry and Regulatory Standards/AWS Security Best Practices`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

10. S3 buckets with sensitive data

ARN: `arn:aws:securityhub:::insight/securityhub/default/12`

Grouped by: Resource ID

Finding filters:

- Resource type is `AwsS3Bucket`
- Type starts with `Sensitive Data Identifications/`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

11. Credentials that may have leaked

ARN: `arn:aws:securityhub:::insight/securityhub/default/13`

Grouped by: Resource ID

Finding filters:

- Type starts with `Sensitive Data Identifications/Passwords/`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

12. EC2 instances that have missing security patches for important vulnerabilities

ARN: `arn:aws:securityhub:::insight/securityhub/default/16`

Grouped by: Resource ID

Finding filters:

- Type starts with `Software and Configuration Checks/Vulnerabilities/CVE`
- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

13. EC2 instances with general unusual behavior

ARN: `arn:aws:securityhub:::insight/securityhub/default/17`

Grouped by: Resource ID

Finding filters:

- Type starts with `Unusual Behaviors`
- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

14. EC2 instances that have ports accessible from the Internet

ARN: `arn:aws:securityhub:::insight/securityhub/default/18`

Grouped by: Resource ID

Finding filters:

- Type starts with `Software and Configuration Checks/AWS Security Best Practices/Network Reachability`
- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

15. EC2 instances that don't meet security standards / best practices

ARN: `arn:aws:securityhub:::insight/securityhub/default/19`

Grouped by: Resource ID

Finding filters:

- Type starts with one of the following:
  - `Software and Configuration Checks/Industry and Regulatory Standards/`
  - `Software and Configuration Checks/AWS Security Best Practices`
- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

16. EC2 instances that are open to the Internet

ARN: `arn:aws:securityhub:::insight/securityhub/default/21`

Grouped by: Resource ID

Finding filters:

- Type starts with `Software and Configuration Checks/AWS Security Best Practices/Network Reachability`
- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

17. EC2 instances associated with adversary reconnaissance

ARN: `arn:aws:securityhub:::insight/securityhub/default/22`

Grouped by: Resource ID

Finding filters:

- Type starts with TTPs/Discovery/Recon
- Resource type is AwsEc2Instance
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

18. AWS resources that are associated with malware

ARN: arn:aws:securityhub:::insight/securityhub/default/23

Grouped by: Resource ID

Finding filters:

- Type starts with one of the following:
  - Effects/Data Exfiltration/Trojan
  - TTPs/Initial Access/Trojan
  - TTPs/Command and Control/Backdoor
  - TTPs/Command and Control/Trojan
  - Software and Configuration Checks/Backdoor
  - Unusual Behaviors/VM/Backdoor
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

19. AWS resources associated with cryptocurrency issues

ARN: arn:aws:securityhub:::insight/securityhub/default/24

Grouped by: Resource ID

Finding filters:

- Type starts with one of the following:
  - Effects/Resource Consumption/Cryptocurrency
  - TTPs/Command and Control/CryptoCurrency
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

20. AWS resources with unauthorized access attempts

ARN: `arn:aws:securityhub:::insight/securityhub/default/25`

Grouped by: Resource ID

Finding filters:

- Type starts with one of the following:
  - `TTPs/Command and Control/UnauthorizedAccess`
  - `TTPs/Initial Access/UnauthorizedAccess`
  - `Effects/Data Exfiltration/UnauthorizedAccess`
  - `Unusual Behaviors/User/UnauthorizedAccess`
  - `Effects/Resource Consumption/UnauthorizedAccess`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

21. Threat Intel indicators with the most hits in the last week

ARN: `arn:aws:securityhub:::insight/securityhub/default/26`

Finding filters:

- Created within the last 7 days

22. Top accounts by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/27`

Grouped by: AWS account ID

Finding filters:

- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

23. Top products by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/28`

Grouped by: Product name

Finding filters:

- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

24. Severity by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/29`

Grouped by: Severity label

Finding filters:

- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

25. Top S3 buckets by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/30`

Grouped by: Resource ID

Finding filters:

- Resource type is `AwsS3Bucket`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

26. Top EC2 instances by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/31`

Grouped by: Resource ID

Finding filters:

- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

27. Top AMIs by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/32`

Grouped by: EC2 instance image ID

Finding filters:

- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

28. Top IAM users by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/33`

Grouped by: IAM access key ID

Finding filters:

- Resource type is `AwsIamAccessKey`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

29. Top resources by counts of failed CIS checks

ARN: `arn:aws:securityhub:::insight/securityhub/default/34`

Grouped by: Resource ID

Finding filters:

- Generator ID starts with `arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule`
- Updated in the last day
- Compliance status is `FAILED`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

30. Top integrations by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/35`

Grouped by: Product ARN

Finding filters:

- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

31. Resources with the most failed security checks

ARN: `arn:aws:securityhub:::insight/securityhub/default/36`

Grouped by: Resource ID

Finding filters:

- Updated in the last day
- Compliance status is `FAILED`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

32. IAM users with suspicious activity

ARN: `arn:aws:securityhub:::insight/securityhub/default/37`

Grouped by: IAM user

Finding filters:

- Resource type is `AwsIamUser`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

33. Resources with the most AWS Health findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/38`

Grouped by: Resource ID

Finding filters:

- `ProductName` equals `Health`

34. Resources with the most AWS Config findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/39`

Grouped by: Resource ID

Finding filters:

- `ProductName` equals `Config`

35. Applications with the most findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/40`

Grouped by: ResourceApplicationArn

Finding filters:

- `RecordState` equals `ACTIVE`
- `Workflow.Status` equals `NEW` or `NOTIFIED`

Understanding custom insights in Security Hub CSPM

In addition to AWS Security Hub CSPM managed insights, you can create custom insights in Security Hub CSPM to track issues that are specific to your environment. Custom insights help you track a curated subset of issues.

Here are some examples of custom insights that may be useful to set up:

- If you own an administrator account, you can set up a custom insight to track critical and high severity findings that are affecting member accounts.
- If you rely on a specific integrated AWS service, you can set up a custom insight to track critical and high severity findings from that service.
- If you rely on a third party integration, you can set up a custom insight to track critical and high severity findings from that integrated product.

You can create completely new custom insights, or start from an existing custom or managed insight.

Each insight can be configured with the following options:

- **Grouping attribute** – The grouping attribute determines which items are displayed in the insight results list. For example, if the grouping attribute is **Product name**, the insight results display the number of findings that are associated with each finding provider.
- **Optional filters** – The filters narrow down the matching findings for the insight.

  A finding is included in the insight results only if it matches all of the provided filters. For example, if the filters are "Product name is GuardDuty" and "Resource type is `AwsS3Bucket`", matching findings must match both of these criteria.

However, Security Hub CSPM applies boolean OR logic to filters that use the same attribute but different values. For example, if the filters are "Product name is GuardDuty" and "Product name is Amazon Inspector", a finding matches if it was generated by either Amazon GuardDuty or Amazon Inspector.

If you use the resource identifier or resource type as the grouping attribute, the insight results include all of the resources that are in the matching findings. The list is not limited to resources that match a resource type filter. For example, an insight identifies findings that are associated with S3 buckets, and groups those findings by resource identifier. A matching finding contains both an S3 bucket resource and an IAM access key resource. The insight results include both resources.

If you enable cross-region aggregation and then create a custom insight, the insight applies to matching findings in the aggregation Region and linked Regions. The exception is if your insight includes a Region filter.

Creating a custom insight

In AWS Security Hub CSPM, custom insights can be used to collect a specific set of findings and track issues that are unique to your environment. For background information about custom insights, see Understanding custom insights in Security Hub CSPM.

Choose your preferred method, and follow the steps to create a custom insight in Security Hub CSPM

Security Hub CSPM console

To create a custom insight (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Insights**.
3. Choose **Create insight**.
4. To select the grouping attribute for the insight:
    a. Choose the search box to display the filter options.
    b. Choose **Group by**.
    c. Select the attribute to use to group the findings that are associated with this insight.

d. Choose **Apply**.

5. Optionally, choose any additional filters to use for this insight. For each filter, define the filter criteria, and then choose **Apply**.
6. Choose **Create insight**.
7. Enter an **Insight name**, and then choose **Create insight**.

Security Hub CSPM API

To create a custom insight (API)

1. To create a custom insight, use the CreateInsight operation of the Security Hub CSPM API. If you use the AWS CLI, run the create-insight command.
2. Populate the `Name` parameter with a name for your custom insight.
3. Populate the `Filters` parameter to specify which findings to include in the insight.
4. Populate the `GroupByAttribute` parameter to specify which attribute is used to group the findings that are included in the insight.
5. Optionally, populate the `SortCriteria` parameter to sort the findings by a specific field.

The following example creates a custom insight that includes critical findings with the `AwsIamRole` resource type. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub create-insight --name "Critical role findings" --filters
 '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],
 "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' --group-by-
attribute "ResourceId"
```

PowerShell

To create a custom insight (PowerShell)

1. Use the `New-SHUBInsight` cmdlet.
2. Populate the `Name` parameter with a name for your custom insight.
3. Populate the `Filter` parameter to specify which findings to include in the insight.
4. Populate the `GroupByAttribute` parameter to specify which attribute is used to group the findings that are included in the insight.

If you've enabled cross-region aggregation and use this cmdlet from the aggregation Region, the insight applies to matching findings from the aggregation and linked Regions.

Example

```
$Filter = @{
    AwsAccountId = [Amazon.SecurityHub.Model.StringFilter]@{
        Comparison = "EQUALS"
        Value = "XXX"
    }
    ComplianceStatus = [Amazon.SecurityHub.Model.StringFilter]@{
        Comparison = "EQUALS"
        Value = 'FAILED'
    }
}
New-SHUBInsight -Filter $Filter -Name TestInsight -GroupByAttribute ResourceId
```

Creating a custom insight from a managed insight (console only)

You can't save changes to or delete a managed insight. However, you can use a managed insight as the basis for a custom insight. This is an option on the Security Hub CSPM console only.

To create a custom insight from a managed insight (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Insights**.
3. Choose the managed insight to work from.
4. Edit the insight configuration as needed.
   - To change the attribute used to group findings in the insight:
     a. To remove the existing grouping, choose the **X** next to the **Group by** setting.
     b. Choose the search box.
     c. Select the attribute to use for grouping.
     d. Choose **Apply**.
   - To remove a filter from the insight, choose the circled **X** next to the filter.
   - To add a filter to the insight:
     a. Choose the search box.

b. Select the attribute and value to use as a filter.

c. Choose **Apply**.

5. When your updates are complete, choose **Create insight**.
6. When prompted, enter an **Insight name**, and then choose **Create insight**.

Editing a custom insight

You can edit an existing custom insight to change the grouping value and filters. After you make the changes, you can save the updates to the original insight, or save the updated version as a new insight.

In AWS Security Hub CSPM, custom insights can be used to collect a specific set of findings and track issues that are unique to your environment. For background information about custom insights, see Understanding custom insights in Security Hub CSPM.

To edit a custom insight, choose your preferred method, and follow the instructions.

Security Hub CSPM console

To edit a custom insight (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Insights**.
3. Choose the custom insight to modify.
4. Edit the insight configuration as needed.
   - To change the attribute used to group findings in the insight:
     a. To remove the existing grouping, choose the **X** next to the **Group by** setting.
     b. Choose the search box.
     c. Select the attribute to use for grouping.
     d. Choose **Apply**.
   - To remove a filter from the insight, choose the circled **X** next to the filter.
   - To add a filter to the insight:
     a. Choose the search box.

   b. Select the attribute and value to use as a filter.
   c. Choose **Apply**.
5. When you complete the updates, choose **Save insight**.
6. When prompted, do one of the following:
   - To update the existing insight to reflect your changes, choose **Update *<Insight_Name>*** and then choose **Save insight**.
   - To create a new insight with the updates, choose **Save new insight**. Enter an **Insight name**, and then choose **Save insight**.

Security Hub CSPM API

To edit a custom insight (API)

1. Use the `UpdateInsight` operation of the Security Hub CSPM API. If you use the AWS CLI run the update-insight command.
2. To identify the custom insight that you want to update, provide the insight's Amazon Resource Name (ARN). To get the ARN of a custom insight, use the `GetInsights` operation or the get-insights command.
3. Update the `Name`, `Filters`, and `GroupByAttribute` parameters as needed.

The following example updates the specified insight. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub update-insight --insight-arn "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111" --filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value":
 "AwsIamRole"}], "SeverityLabel": [{"Comparison": "EQUALS", "Value": "HIGH"}]}' --
name "High severity role findings"
```

PowerShell

To edit a custom insight (PowerShell)

1. Use the `Update-SHUBInsight` cmdlet.
2. To identify the custom insight, provide the insight's Amazon Resource Name (ARN). To get the ARN of a custom insight, use the `Get-SHUBInsight` cmdlet.

3. Update the `Name`, `Filter`, and `GroupByAttribute` parameters as needed.

Example

```
$Filter = @{
    ResourceType = [Amazon.SecurityHub.Model.StringFilter]@{
        Comparison = "EQUALS"
        Value = "AwsIamRole"
    }
    SeverityLabel = [Amazon.SecurityHub.Model.StringFilter]@{
        Comparison = "EQUALS"
        Value = "HIGH"
    }
}

Update-SHUBInsight -InsightArn "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111" -Filter $Filter -Name "High severity role findings"
```

Deleting a custom insight

In AWS Security Hub CSPM, custom insights can be used to collect a specific set of findings and track issues that are unique to your environment. For background information about custom insights, see Understanding custom insights in Security Hub CSPM.

To delete a custom insight, choose your preferred method, and follow the instructions. You can't delete a managed insight.

Security Hub CSPM console

To delete a custom insight (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Insights**.
3. Locate the custom insight to delete.
4. For that insight, choose the more options icon (the three dots in the top-right corner of the card).
5. Choose **Delete**.

Security Hub CSPM API

To delete a custom insight (API)

1. Use the `DeleteInsight` operation of the Security Hub CSPM API. If you use the AWS CLI run the delete-insight command.
2. To identify the custom insight to delete, provide the insight's ARN. To get the ARN of a custom insight, use the `GetInsights` operation or get-insights command.

The following example deletes the specified insight. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub delete-insight --insight-arn "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
```

PowerShell

To delete a custom insight (PowerShell)

1. Use the `Remove-SHUBInsight` cmdlet.
2. To identify the custom insight, provide the insight's ARN. To get the ARN of a custom insight, use the `Get-SHUBInsight` cmdlet.

Example

```
-InsightArn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/custom/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Automatically modifying and acting on findings in Security Hub CSPM

AWS Security Hub CSPM has features that automatically modify and take action on findings based on your specifications.

Security Hub CSPM currently supports two types of automations:

- **Automation rules** – Automatically update and suppress findings in near real time based on criteria that you define.
- **Automated response and remediation** – Create custom Amazon EventBridge rules that define automatic actions to take against specific findings and insights.

Automation rules are helpful when you want to automatically update finding fields in the AWS Security Finding Format (ASFF). For example, you can use an automation rule to update the severity level or workflow status of findings from a specific third-party integrations. Using the automation rule eliminates the need to manually update the severity level or workflow status of each finding from this third-party product.

EventBridge rules are helpful when you want to take actions outside of Security Hub CSPM with regards to specific findings or send specific findings to third-party tools for remediation or additional investigation. The rules can be used to trigger supported actions, such as invoking an AWS Lambda function or notifying an Amazon Simple Notification Service (Amazon SNS) topic about a specific finding.

Automation rules take effect before EventBridge rules are applied. That is, automation rules are triggered and update a finding before EventBridge receives the finding. EventBridge rules then apply to the updated finding.

When setting up automations for security controls, we recommend filtering based on control ID rather than title or description. Whereas Security Hub CSPM occasionally updates control titles and descriptions, control IDs stay the same.

Topics

Understanding automation rules in Security Hub CSPM

You can use automation rules to automatically update findings in AWS Security Hub CSPM. As it ingests findings, Security Hub CSPM can apply a variety of rule actions, such as suppressing findings, changing their severity, and adding notes. Such rule actions modify findings that match your specified criteria.

Examples of use cases for automation rules include the following:

- Elevating a finding's severity to `CRITICAL` if the finding's resource ID refers to a business-critical resource.
- Elevating a finding's severity from `HIGH` to `CRITICAL` if the finding affects resources in specific production accounts.
- Assigning specific findings that have a severity of `INFORMATIONAL` a `SUPPRESSED` workflow status.

You can create and manage automation rules from a Security Hub CSPM administrator account only.

Rules apply to both new findings and updated findings. You can create a custom rule from scratch, or use a rule template provided by Security Hub CSPM. You can also start with a template and modify it as needed.

Defining rule criteria and rule actions

From a Security Hub CSPM administrator account, you can create an automation rule by defining one or more rule *criteria* and one or more rule *actions*. When a finding matches the defined criteria, Security Hub CSPM applies the rule actions to it. For more information about available criteria and actions, see Available rule criteria and rule actions.

Security Hub CSPM currently supports a maximum of 100 automation rules for each administrator account.

The Security Hub CSPM administrator account can also edit, view, and delete automation rules. A rule applies to matching findings in the administrator account and all of its member accounts. By providing member account IDs as rule criteria, Security Hub CSPM administrators can also use automation rules to update or suppress findings in specific member accounts.

An automation rule applies only in the AWS Region in which it's created. To apply a rule in multiple Regions, the administrator must create the rule in each Region. This can be done through the Security Hub CSPM console, Security Hub CSPM API, or AWS CloudFormation.You can also use a multi-Region deployment script.

Available rule criteria and rule actions

The following AWS Security Finding Format (ASFF) fields are currently supported as criteria for automation rules:

Rule criterion	Filter operators	Field type
AwsAccountId	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
AwsAccountName	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
CompanyName	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ComplianceAssociatedStandardsId	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ComplianceSecurityControlId	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ComplianceStatus	Is, Is Not	Select: [FAILED, NOT_AVAILABLE , PASSED, WARNING]
Confidence	Eq (equal-to), Gte (greater-than-equal), Lte (less-than-equal)	Number
CreatedAt	Start, End, DateRange	Date (formatted as 2022-12-01T21:47:39.269Z)

Rule criterion	Filter operators	Field type
Criticality	Eq (equal-to), Gte (greater-than-equal), Lte (less-than-equal)	Number
Description	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
FirstObservedAt	Start, End, DateRange	Date (formatted as 2022-12-01T21:47:39.269Z)
GeneratorId	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
Id	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
LastObservedAt	Start, End, DateRange	Date (formatted as 2022-12-01T21:47:39.269Z)
NoteText	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
NoteUpdatedAt	Start, End, DateRange	Date (formatted as 2022-12-01T21:47:39.269Z)

Rule criterion	Filter operators	Field type
NoteUpdatedBy	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ProductArn	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ProductName	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
RecordState	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
RelatedFindingsId	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
RelatedFindingsProductArn	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ResourceApplicationArn	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String

Rule criterion	Filter operators	Field type
ResourceApplicationName	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ResourceDetailsOther	CONTAINS, EQUALS, NOT_CONTAINS, NOT_EQUALS	Map
ResourceId	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ResourcePartition	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ResourceRegion	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ResourceTags	CONTAINS, EQUALS, NOT_CONTAINS, NOT_EQUALS	Map
ResourceType	Is, Is Not	Select (see Resources supported by ASFF)
SeverityLabel	Is, Is Not	Select: [CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL]

Rule criterion	Filter operators	Field type
SourceUrl	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
Title	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
Type	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
UpdatedAt	Start, End, DateRange	Date (formatted as 2022-12-01T21:47:39.269Z)
UserDefinedFields	CONTAINS, EQUALS, NOT_CONTAINS, NOT_EQUALS	Map
VerificationState	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
WorkflowStatus	Is, Is Not	Select: [NEW, NOTIFIED, RESOLVED, SUPPRESSED]

For criteria that are labeled as string fields, using different filter operators on the same field affects the evaluation logic. For more information, see StringFilter in the *AWS Security Hub CSPM API Reference*.

Each criterion supports a maximum number of values that can be used to filter matching findings. For the limits on each criterion, see AutomationRulesFindingFilters in the *AWS Security Hub CSPM API Reference*.

The following ASFF fields are currently supported as actions for automation rules:

- `Confidence`
- `Criticality`
- `Note`
- `RelatedFindings`
- `Severity`
- `Types`
- `UserDefinedFields`
- `VerificationState`
- `Workflow`

For more information about specific ASFF fields, see AWS Security Finding Format (ASFF) syntax.

> **Tip**
>
> If you want Security Hub CSPM to stop generating findings for a specific control, we recommend disabling the control instead of using an automation rule. When you disable a control, Security Hub CSPM stops running security checks on it and stops generating findings for it, so you won't incur charges for that control. We recommend using automation rules to change the values of specific ASFF fields for findings that match defined criteria. For more information about disabling controls, see Disabling controls in Security Hub CSPM.

Findings that automation rules evaluate

An automation rule evaluates new and updated findings that Security Hub CSPM generates or ingests through the BatchImportFindings operation *after* you create the rule. Security Hub CSPM updates control findings every 12-24 hours or when the associated resource changes state. For more information, see Schedule for running security checks.

Automation rules evaluate original, provider-supplied findings. Providers can supply new findings and update existing findings through the `BatchImportFindings` operation of the Security Hub CSPM API. Rules aren't triggered when you update finding fields after rule creation through the BatchUpdateFindings operation. If you create an automation rule and make a `BatchUpdateFindings` update that both affect the same finding field, the last update sets the value for that field. Take the following example:

1. You use `BatchUpdateFindings` to update the `Workflow.Status` field of a finding from `NEW` to `NOTIFIED`.
2. If you call `GetFindings`, the `Workflow.Status` field now has a value of `NOTIFIED`.
3. You create an automation rule that changes the `Workflow.Status` field of the finding from `NEW` to `SUPPRESSED` (recall that rules ignore updates made with `BatchUpdateFindings`).
4. The finding provider uses `BatchImportFindings` to update the finding and changes the `Workflow.Status` field to `NEW`.
5. If you call `GetFindings`, the `Workflow.Status` field now has a value of `SUPPRESSED` because the automation rule was applied, and the rule was the last action taken on the finding.

When you create or edit a rule on the Security Hub CSPM console, the console displays a beta of findings that match the rule criteria. Whereas automation rules evaluate original findings sent by the finding provider, the console beta reflects findings in their final state as they would be shown in a response to the GetFindings API operation (that is, after rule actions or other updates are applied to the finding).

How rule order works

When creating automation rules, you assign each rule an order. This determines the order in which Security Hub CSPM applies your automation rules, and becomes important when multiple rules relate to the same finding or finding field.

When multiple rule actions relate to the same finding or finding field, the rule with the highest numerical value for rule order applies last and has the ultimate effect.

When you create a rule in the Security Hub CSPM console, Security Hub CSPM automatically assigns rule order based on the order of rule creation. The most recently created rule has the lowest numerical value for rule order and therefore applies first. Security Hub CSPM applies subsequent rules in ascending order.

When you create a rule through the Security Hub CSPM API or AWS CLI, Security Hub CSPM applies the rule with the lowest numerical value for `RuleOrder` first. It then applies subsequent rules in ascending order. If multiple findings have the same `RuleOrder`, Security Hub CSPM applies a rule with an earlier value for the `UpdatedAt` field first (that is, the rule which was most recently edited applies last).

You can modify rule order at any time.

Example of rule order:

Rule A (rule order is 1):

- Rule A criteria
  - `ProductName` = `Security Hub CSPM`
  - `Resources.Type` is `S3 Bucket`
  - `Compliance.Status` = `FAILED`
  - `RecordState` is `NEW`
  - `Workflow.Status` = `ACTIVE`
- Rule A actions
  - Update `Confidence` to 95
  - Update `Severity` to `CRITICAL`

Rule B (rule order is 2):

- Rule B criteria
  - `AwsAccountId` = `123456789012`
- Rule B actions
  - Update `Severity` to `INFORMATIONAL`

Rule A actions apply first to Security Hub CSPM findings that match Rule A criteria. Next, Rule B actions apply to Security Hub CSPM findings with the specified account ID. In this example, since Rule B applies last, the end value of `Severity` in findings from the specified account ID is `INFORMATIONAL`. Based on the Rule A action, the end value of `Confidence` in matched findings is 95.

Creating automation rules

An automation rule can be used to automatically update findings in AWS Security Hub CSPM. You can create a custom automation rule from scratch or, on the Security Hub CSPM console, use a pre-populated rule template. For background information about how automation rules work, see Understanding automation rules in Security Hub CSPM.

You can only create one automation rule at a time. To create multiple automation rules, follow the console procedures multiple times, or call the API or command multiple times with your desired parameters.

You must create an automation rule in each Region and account in which you want the rule to apply to findings.

When you create an automation rule in the Security Hub CSPM console, Security Hub CSPM shows you a beta of the findings to which your rule applies. The beta is currently not supported if your rule criteria include a CONTAINS or NOT_CONTAINS filter. You can choose these filters for map and string field types.

> **Important**
>
> AWS recommends that you don't include personally identifying, confidential, or sensitive information in your rule name, description, or other fields.

Creating a custom automation rule

Choose your preferred method, and complete the following steps to create a custom automation rule.

Console

To create a custom automation rule (console)

1. Using the credentials of the Security Hub CSPM administrator, open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Automations**.
3. Choose **Create rule**. For **Rule Type**, choose **Create custom rule**.
4. In the **Rule** section, provide a unique rule name and a description for your rule.

5. For **Criteria**, use the **Key**, **Operator**, and **Value** drop down menus to specify your rule criteria. You must specify at least one rule criterion.

   If supported for your selected criteria, the console shows you a beta of findings that match your criteria.
6. For **Automated action**, use the drop down menus to specify which finding fields to update when findings match your rule criteria. You must specify at least one rule action.
7. For **Rule status**, choose whether you want the rule to be **Enabled** or **Disabled** after it's created.
8. (Optional) Expand the **Additional settings** section. Select **Ignore subsequent rules for findings that match these criteria** if you want this rule to be the last rule applied to findings that match the rule criteria.
9. (Optional) For **Tags**, add tags as key-value pairs to help you easily identify the rule.
10. Choose **Create rule**.

API

To create a custom automation rule (API)

1. Run `CreateAutomationRule` from the Security Hub CSPM administrator account. This API creates a rule with a specific Amazon Resource Name (ARN).
2. Provide a name and description for the rule.
3. Set the `IsTerminal` parameter to `true` if you want this rule to be the last rule applied to findings that match the rule criteria.
4. For the `RuleOrder` parameter, provide the order of the rule. Security Hub CSPM applies rules with a lower numerical value for this parameter first.
5. For the `RuleStatus` parameter, specify if you want Security Hub CSPM to enable and start applying the rule to findings after creation. If no value is specified, the default value is `ENABLED`. A value of `DISABLED` means that the rule is paused after creation.
6. For the `Criteria` parameter, provide the criteria that you want Security Hub CSPM to use to filter your findings. The rule action will apply to findings that match the criteria. For a list of supported criteria, see Available rule criteria and rule actions.
7. For the `Actions` parameter, provide the actions that you want Security Hub CSPM to take when there's a match between a finding and your defined criteria. For a list of supported actions, see Available rule criteria and rule actions.

The following example AWS CLI command creates an automation rule the updates the workflow status and note of matching findings. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub create-automation-rule \
--actions '[{
 "Type": "FINDING_FIELDS_UPDATE",
 "FindingFieldsUpdate": {
 "Severity": {
 "Label": "HIGH"
 },
 "Note": {
 "Text": "Known issue that is a risk. Updated by automation rules",
 "UpdatedBy": "sechub-automation"
 }
 }
 }]' \
--criteria '{
 "SeverityLabel": [{
 "Value": "INFORMATIONAL",
 "Comparison": "EQUALS"
 }]
 }' \
--description "A sample rule" \
--no-is-terminal \
--rule-name "sample rule" \
--rule-order 1 \
--rule-status "ENABLED" \
--region us-east-1
```

Creating an automation rule from a template (console only)

Rule templates reflect common use cases for automation rules. Currently, only the Security Hub CSPM console supports rule templates. Complete the following steps to create an automation rule from a template in the console.

To create an automation rule from a template (console)

1. Using the credentials of the Security Hub CSPM administrator, open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Automations**.

3. Choose **Create rule**. For **Rule Type**, choose **Create a rule from template**.
4. Select a rule template from the drop down menu.
5. (Optional) If necessary for your use case, modify the **Rule**, **Criteria**, and **Automated action** sections. You must specify at least one rule criterion and one rule action.

   If supported for your selected criteria, the console shows you a beta of findings that match your criteria.
6. For **Rule status**, choose whether you want the rule to be **Enabled** or **Disabled** after it's created.
7. (Optional) Expand the **Additional settings** section. Select **Ignore subsequent rules for findings that match these criteria** if you want this rule to be the last rule applied to findings that match the rule criteria.
8. (Optional) For **Tags**, add tags as key-value pairs to help you easily identify the rule.
9. Choose **Create rule**.

Viewing automation rules

An automation rule can be used to automatically update findings in AWS Security Hub CSPM. For background information about how automation rules work, see Understanding automation rules in Security Hub CSPM.

Choose your preferred method, and follow the steps to view your existing automation rules and the details of each rule.

To view a history of how automation rules have changed your findings, see Reviewing finding details and history in Security Hub CSPM.

Console

To view automation rules (console)

1. Using the credentials of the Security Hub CSPM administrator, open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Automations**.
3. Choose a rule name. Alternatively, select a rule.
4. Choose **Actions** and **View**.

API

To view automation rules (API)

1. To view the automation rules for your account, run `ListAutomationRules` from the Security Hub CSPM administrator account. This API returns the rule ARNs and other metadata for your rules. No input parameters are required for this API, but you can optionally provide `MaxResults` to limit the number of results and `NextToken` as a pagination parameter. The initial value of `NextToken` should be `NULL`.
2. For additional rule details, including the criteria and actions for a rule, run `BatchGetAutomationRules` from the Security Hub CSPM administrator account. Provide the ARNs of the automation rules that you want details for.

   The following example retrieves details for the specified automation rules. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub batch-get-automation-rules \
--automation-rules-arns '["arn:aws:securityhub:us-
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
 "arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"]' \
--region us-east-1
```

Editing automation rules

An automation rule can be used to automatically update findings in AWS Security Hub CSPM. For background information about how automation rules work, see Understanding automation rules in Security Hub CSPM.

After creating an automation rule, the delegated Security Hub CSPM administrator can edit the rule. When you edit an automation rule, the changes apply to new and updated findings that Security Hub CSPM generates or ingests after the rule edit.

Choose your preferred method, and follow the steps to edit the contents of an automation rule. You can edit one or more rules with a single request. For instructions on editing rule order, see Editing automation rule order.

Console

To edit automation rules (console)

1. Using the credentials of the Security Hub CSPM administrator, open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Automations**.
3. Select the rule that you want to edit. Choose **Action** and **Edit**.
4. Change the rule as desired, and choose **Save changes**.

API

To edit automation rules (API)

1. Run `BatchUpdateAutomationRules` from the Security Hub CSPM administrator account.
2. For the `RuleArn` parameter, provide the ARN of the rule(s) that you want to edit.
3. Provide the new values for the parameters that you want to edit. You can edit any parameter except `RuleArn`.

The following example updates the specified automation rule. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub batch-update-automation-rules \
--update-automation-rules-request-items '[
    {
      "Actions": [{
        "Type": "FINDING_FIELDS_UPDATE",
        "FindingFieldsUpdate": {
          "Note": {
            "Text": "Known issue that is a risk",
            "UpdatedBy": "sechub-automation"
          },
          "Workflow": {
            "Status": "NEW"
          }
        }
      }],
      "Criteria": {
```

```
        "SeverityLabel": [{
         "Value": "LOW",
         "Comparison": "EQUALS"
        }]
      },
      "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-
rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "RuleOrder": 14,
      "RuleStatus": "DISABLED",
    }
  ]' \
--region us-east-1
```

Editing automation rule order

An automation rule can be used to automatically update findings in AWS Security Hub CSPM. For background information about how automation rules work, see Understanding automation rules in Security Hub CSPM.

After creating an automation rule, the delegated Security Hub CSPM administrator can edit the rule.

If you want to keep the rule criteria and actions the same, but change the order in which Security Hub CSPM applies an automation rule, you can edit just the rule order. Choose your preferred method, and follow the steps to edit rule order.

For instructions on editing the criteria or actions of an automation rule, see Editing automation rules.

Console

To edit automation rule order (console)

1. Using the credentials of the Security Hub CSPM administrator, open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Automations**.
3. Select the rule whose order you want to change. Choose **Edit priority**.
4. Choose **Move up** to increase the rule's priority by one unit. Choose **Move down** to decrease the rule priority's by one unit. Choose **Move to top** to assign the rule an order of **1** (this gives the rule precedence over other existing rules).

> **Note**
>
> When you create a rule in the Security Hub CSPM console, Security Hub CSPM automatically assigns rule order based on the order of rule creation. The most recently created rule has the lowest numerical value for rule order and therefore applies first.

API

To edit automation rule order (API)

1. Use the `BatchUpdateAutomationRules` operation from the Security Hub CSPM administrator account.
2. For the `RuleArn` parameter, provide the ARN of the rule(s) whose order you want to edit.
3. Modify the value of the `RuleOrder` field.

> **Note**
>
> If multiple rules have the same `RuleOrder`, Security Hub CSPM applies a rule with an earlier value for the `UpdatedAt` field first (that is, the rule which was most recently edited applies last).

Deleting or disabling automation rules

An automation rule can be used to automatically update findings in AWS Security Hub CSPM. For background information about how automation rules work, see Understanding automation rules in Security Hub CSPM.

When you delete an automation rule, Security Hub CSPM removes it from your account and no longer applies the rule to findings. As an alternative to deletion, you can *disable* a rule. This retains the rule for future use, but Security Hub CSPM won't apply the rule to any matching findings until you enable it.

Choose your preferred method, and follow the steps to delete an automation rule. You can delete one or more rules in a single request.

Console

To delete or disable automation rules (console)

1. Using the credentials of the Security Hub CSPM administrator, open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Automations**.
3. Select the rule(s) that you want to delete. Choose **Action** and **Delete** (to retain a rule, but disable it temporarily, choose **Disable**).
4. Confirm your choice, and choose **Delete**.

API

To delete or disable automation rules (API)

1. Use the `BatchDeleteAutomationRules` operation from the Security Hub CSPM administrator account.
2. For the `AutomationRulesArns` parameter, provide the ARN of the rule(s) that you want to delete (to retain a rule, but disable it temporarily, provide `DISABLED` for the `RuleStatus` parameter).

The following example deletes the specified automation rule. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub batch-delete-automation-rules \
--automation-rules-arns '["arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]' \
--region us-east-1
```

Examples of automation rules

This section provides examples of automation rules for common Security Hub CSPM use cases. These examples correspond to rule templates that are available on the Security Hub CSPM console.

Elevate severity to Critical when specific resource such as an S3 bucket is at risk

In this example, the rule criteria are matched when the `ResourceId` in a finding is a specific Amazon Simple Storage Service (Amazon S3) bucket. The rule action is to change the severity of matched findings to `CRITICAL`. You can modify this template to apply to other resources.

Example API request:

```
{
    "IsTerminal": true,
    "RuleName": "Elevate severity of findings that relate to important resources",
    "RuleOrder": 1,
    "RuleStatus": "ENABLED",
    "Description": "Elevate finding severity to CRITICAL when specific resource such as
 an S3 bucket is at risk",
    "Criteria": {
        "ProductName": [{
            "Value": "Security Hub CSPM",
            "Comparison": "EQUALS"
        }],
        "ComplianceStatus": [{
            "Value": "FAILED",
            "Comparison": "EQUALS"
        }],
        "RecordState": [{
            "Value": "ACTIVE",
            "Comparison": "EQUALS"
        }],
        "WorkflowStatus": [{
            "Value": "NEW",
            "Comparison": "EQUALS"
        }],
        "ResourceId": [{
            "Value": "arn:aws:s3:::amzn-s3-demo-bucket/developers/design_info.doc",
            "Comparison": "EQUALS"
        }]
    },
    "Actions": [{
        "Type": "FINDING_FIELDS_UPDATE",
        "FindingFieldsUpdate": {
            "Severity": {
                "Label": "CRITICAL"
            },
```

```
            "Note": {
                "Text": "This is a critical resource. Please review ASAP.",
                "UpdatedBy": "sechub-automation"
            }
        }
    }]
}
```

Example CLI command:

```
$
aws securityhub create-automation-rule \
--is-terminal \
--rule-name "Elevate severity of findings that relate to important resources" \
--rule-order 1 \
--rule-status "ENABLED" \

--description "Elevate finding severity to CRITICAL when specific resource such as an
 S3 bucket is at risk" \
--criteria '{
"ProductName": [{
"Value": "Security Hub CSPM",
"Comparison": "EQUALS"
}],
"ComplianceStatus": [{
"Value": "FAILED",
"Comparison": "EQUALS"
}],
"RecordState": [{
"Value": "ACTIVE",
"Comparison": "EQUALS"
}],
"WorkflowStatus": [{
"Value": "NEW",
"Comparison": "EQUALS"
}],
"ResourceId": [{
"Value": "arn:aws:s3:::amzn-s3-demo-bucket/developers/design_info.doc",
"Comparison": "EQUALS"
}]
}' \
--actions '[{
"Type": "FINDING_FIELDS_UPDATE",
```

```
"FindingFieldsUpdate": {
"Severity": {
"Label": "CRITICAL"
},
"Note": {
"Text": "This is a critical resource. Please review ASAP.",
"UpdatedBy": "sechub-automation"
}
}
}]' \
--region us-east-1
```

Elevate severity of findings that relate to resources in production accounts

In this example, the rule criteria are matched when a HIGH severity finding is generated in specific production accounts. The rule action is to change the severity of matched findings to CRITICAL.

Example API request:

```
{
    "IsTerminal": false,
    "RuleName": "Elevate severity for production accounts",
    "RuleOrder": 1,
    "RuleStatus": "ENABLED",
    "Description": "Elevate finding severity from HIGH to CRITICAL for findings that
 relate to resources in specific production accounts",
    "Criteria": {
        "ProductName": [{
            "Value": "Security Hub CSPM",
            "Comparison": "EQUALS"
        }],
        "ComplianceStatus": [{
            "Value": "FAILED",
            "Comparison": "EQUALS"
        }],
        "RecordState": [{
            "Value": "ACTIVE",
            "Comparison": "EQUALS"
        }],
        "WorkflowStatus": [{
            "Value": "NEW",
            "Comparison": "EQUALS"
        }],
```

```
        "SeverityLabel": [{
            "Value": "HIGH",
            "Comparison": "EQUALS"
        }],
        "AwsAccountId": [
        {
            "Value": "111122223333",
            "Comparison": "EQUALS"
        },
        {
            "Value": "123456789012",
            "Comparison": "EQUALS"
        }]
    },
    "Actions": [{
        "Type": "FINDING_FIELDS_UPDATE",
        "FindingFieldsUpdate": {
            "Severity": {
                "Label": "CRITICAL"
            },
            "Note": {
                "Text": "A resource in production accounts is at risk. Please review
 ASAP.",
                "UpdatedBy": "sechub-automation"
            }
        }
    }]
}
```

Example CLI command:

```
aws securityhub create-automation-rule \
--no-is-terminal \
--rule-name "Elevate severity of findings that relate to resources in production
 accounts" \
--rule-order 1 \
--rule-status "ENABLED" \
--description "Elevate finding severity from HIGH to CRITICAL for findings that relate
 to resources in specific production accounts" \
--criteria '{
"ProductName": [{
"Value": "Security Hub CSPM",
```

```
"Comparison": "EQUALS"
}],
"ComplianceStatus": [{
"Value": "FAILED",
"Comparison": "EQUALS"
}],
"RecordState": [{
"Value": "ACTIVE",
"Comparison": "EQUALS"
}],
"SeverityLabel": [{
"Value": "HIGH",
"Comparison": "EQUALS"
}],
"AwsAccountId": [
{
"Value": "111122223333",
"Comparison": "EQUALS"
},
{
"Value": "123456789012",
"Comparison": "EQUALS"
}]
}' \
--actions '[{
"Type": "FINDING_FIELDS_UPDATE",
"FindingFieldsUpdate": {
"Severity": {
"Label": "CRITICAL"
},
"Note": {
"Text": "A resource in production accounts is at risk. Please review ASAP.",
"UpdatedBy": "sechub-automation"
}
}
}]' \
--region us-east-1
```

Suppress informational findings

In this example, the rule criteria are matched for `INFORMATIONAL` severity findings sent to Security Hub CSPM from Amazon GuardDuty. The rule action is to change the workflow status of matched findings to SUPPRESSED.

Example API request:

```
{
    "IsTerminal": false,
    "RuleName": "Suppress informational findings",
    "RuleOrder": 1,
    "RuleStatus": "ENABLED",
    "Description": "Suppress GuardDuty findings with INFORMATIONAL severity",
    "Criteria": {
        "ProductName": [{
            "Value": "GuardDuty",
            "Comparison": "EQUALS"
        }],
        "RecordState": [{
            "Value": "ACTIVE",
            "Comparison": "EQUALS"
        }],
        "WorkflowStatus": [{
            "Value": "NEW",
            "Comparison": "EQUALS"
        }],
        "SeverityLabel": [{
            "Value": "INFORMATIONAL",
            "Comparison": "EQUALS"
        }]
    },
    "Actions": [{
        "Type": "FINDING_FIELDS_UPDATE",
        "FindingFieldsUpdate": {
            "Workflow": {
                "Status": "SUPPRESSED"
            },
            "Note": {
                "Text": "Automatically suppress GuardDuty findings with INFORMATIONAL
 severity",
                "UpdatedBy": "sechub-automation"
            }
        }
    }]
}
```

Example CLI command:

```
aws securityhub create-automation-rule \
--no-is-terminal \
--rule-name "Suppress informational findings" \
--rule-order 1 \
--rule-status "ENABLED" \
--description "Suppress GuardDuty findings with INFORMATIONAL severity" \
--criteria '{
"ProductName": [{
"Value": "GuardDuty",
"Comparison": "EQUALS"
}],
"ComplianceStatus": [{
"Value": "FAILED",
"Comparison": "EQUALS"
}],
"RecordState": [{
"Value": "ACTIVE",
"Comparison": "EQUALS"
}],
"WorkflowStatus": [{
"Value": "NEW",
"Comparison": "EQUALS"
}],
"SeverityLabel": [{
"Value": "INFORMATIONAL",
"Comparison": "EQUALS"
}]
}' \
--actions '[{
"Type": "FINDING_FIELDS_UPDATE",
"FindingFieldsUpdate": {
"Workflow": {
"Status": "SUPPRESSED"
},
"Note": {
"Text": "Automatically suppress GuardDuty findings with INFORMATIONAL severity",
"UpdatedBy": "sechub-automation"
}
}
}]' \
--region us-east-1
```

Using EventBridge for automated response and remediation

By creating rules in Amazon EventBridge, you can respond automatically to AWS Security Hub CSPM findings. Security Hub CSPM sends findings as *events* to EventBridge in near-real time. You can write simple rules to indicate which events you are interested in and what automated actions to take when an event matches a rule. The actions that can be automatically triggered include the following:

- Invoking an AWS Lambda function
- Invoking the Amazon EC2 run command
- Relaying the event to Amazon Kinesis Data Streams
- Activating an AWS Step Functions state machine
- Notifying an Amazon SNS topic or an Amazon SQS queue
- Sending a finding to a third-party ticketing, chat, SIEM, or incident response and management tool

Security Hub CSPM automatically sends all new findings and all updates to existing findings to EventBridge as EventBridge events. You can also create custom actions that allow you to send selected findings and insight results to EventBridge.

You then configure EventBridge rules to respond to each type of event.

For more information about using EventBridge, see the *Amazon EventBridge User Guide*.

> **Note**
>
> As a best practice, make sure that the permissions granted to your users to access EventBridge use least-privilege AWS Identity and Access Management (IAM) policies that grant only the required permissions.
> For more information, see Identity and access management in Amazon EventBridge.

A set of templates for cross-account automated response and remediation is also available in AWS Solutions. The templates leverage EventBridge event rules and Lambda functions. You deploy the solution using AWS CloudFormation and AWS Systems Manager. The solution can create fully automated response and remediation actions. It can also use Security Hub CSPM custom actions to

create user-triggered response and remediation actions. For details on how to configure and use the solution, see the Automated Security Response on AWS solution page.

Topics

Security Hub CSPM event types in EventBridge

Security Hub CSPM uses the following Amazon EventBridge event types to integrate with EventBridge.

On the EventBridge dashboard for Security Hub CSPM, **All Events** includes all of these event types.

All findings (Security Hub Findings - Imported)

Security Hub CSPM automatically sends all new findings and all updates to existing findings to EventBridge as **Security Hub Findings - Imported** events. Each **Security Hub Findings - Imported** event contains a single finding.

Every `BatchImportFindings` and `BatchUpdateFindings` request triggers a **Security Hub Findings - Imported** event.

For administrator accounts, the event feed in EventBridge includes events for findings from both their account and from their member accounts.

In an aggregation Region, the event feed includes events for findings from the aggregation Region and the linked Regions. Cross-Region findings are included in the event feed in near real time. For information on how to configure finding aggregation, see the section called “Aggregating data across Regions”.

You can define rules in EventBridge that automatically route findings to a remediation workflow, third-party tool, or other supported EventBridge target. The rules can include filters that only apply the rule if the finding has specific attribute values.

You use this method to automatically send all findings, or all findings that have specific characteristics, to a response or remediation workflow.

See the section called "Configuring an EventBridge rule".

Findings for custom actions (Security Hub Findings - Custom Action)

Security Hub CSPM also sends findings that are associated with custom actions to EventBridge as **Security Hub Findings - Custom Action** events.

This is useful for analysts working with the Security Hub CSPM console who want to send a specific finding, or a small set of findings, to a response or remediation workflow. You can select a custom action for up to 20 findings at a time. Each finding is sent to EventBridge as a separate EventBridge event.

When you create a custom action, you assign it a custom action ID. You can use this ID to create an EventBridge rule that takes a specified action after receiving a finding that is associated with that custom action ID.

See the section called "Configuring and using custom actions".

For example, you can create a custom action in Security Hub CSPM called `send_to_ticketing`. Then in EventBridge, you create a rule that is triggered when EventBridge receives a finding that includes the `send_to_ticketing` custom action ID. The rule includes logic to send the finding to your ticketing system. You can then select findings within Security Hub CSPM and use the custom action in Security Hub CSPM to manually send findings to your ticketing system.

For examples of how to send Security Hub CSPM findings to EventBridge for further processing, see How to Integrate AWS Security Hub CSPM Custom Actions with PagerDuty and How to Enable Custom Actions in AWS Security Hub CSPM on the AWS Partner Network (APN) Blog.

Insight results for custom actions (Security Hub Insight Results)

You can also use custom actions to send sets of insight results to EventBridge as **Security Hub Insight Results** events. Insight results are the resources that match an insight. Note that when you send insight results to EventBridge, you are not sending the findings to EventBridge. You are only sending the resource identifiers that are associated with the insight results. You can send up to 100 resource identifiers at a time.

Similar to custom actions for findings, you first create the custom action in Security Hub CSPM, and then create a rule in EventBridge.

See the section called "Configuring and using custom actions".

For example, suppose you see a particular insight result of interest that you want to share with a colleague. In that case, you can use a custom action to send that insight result to the colleague through a chat or ticketing system.

EventBridge event formats for Security Hub CSPM

The **Security Hub Findings - Imported**, **Security Findings - Custom Action**, and **Security Hub Insight Results** event types use the following event formats.

The event format is the format that is used when Security Hub CSPM sends an event to EventBridge.

Security Hub Findings - Imported

Security Hub Findings - Imported events that are sent from Security Hub CSPM to EventBridge use the following format.

```
{
   "version":"0",
   "id":"CWE-event-id",
   "detail-type":"Security Hub Findings - Imported",
   "source":"aws.securityhub",
   "account":"111122223333",
   "time":"2019-04-11T21:52:17Z",
   "region":"us-west-2",
   "resources":[
      "arn:aws:securityhub:us-west-2::product/aws/macie/arn:aws:macie:us-
west-2:111122223333:integtest/trigger/6294d71b927c41cbab915159a8f326a3/alert/
f2893b211841"
   ],
   "detail":{
      "findings": [{
         <finding content>
       }]
   }
}
```

<finding content> is the content, in JSON format, of the finding that is sent by the event. Each event sends a single finding.

For a complete list of finding attributes, see AWS Security Finding Format (ASFF).

For information about how to configure EventBridge rules that are triggered by these events, see the section called "Configuring an EventBridge rule".

Security Hub Findings - Custom Action

Security Hub Findings - Custom Action events that are sent from Security Hub CSPM to EventBridge use the following format. Each finding is sent in a separate event.

```
{
  "version": "0",
  "id": "1a1111a1-b22b-3c33-444d-5555e5ee5555",
  "detail-type": "Security Hub Findings - Custom Action",
  "source": "aws.securityhub",
  "account": "111122223333",
  "time": "2019-04-11T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:securityhub:us-west-1:111122223333:action/custom/custom-action-name"
  ],
  "detail": {
    "actionName":"custom-action-name",
    "actionDescription": "description of the action",
    "findings": [
      {
        <finding content>
      }
    ]
  }
}
```

<finding content> is the content, in JSON format, of the finding that is sent by the event. Each event sends a single finding.

For a complete list of finding attributes, see AWS Security Finding Format (ASFF).

For information about how to configure EventBridge rules that are triggered by these events, see the section called "Configuring and using custom actions".

Security Hub Insight Results

Security Hub Insight Results events that are sent from Security Hub CSPM to EventBridge use the following format.

```
{
  "version": "0",
  "id": "1a1111a1-b22b-3c33-444d-5555e5ee5555",
  "detail-type": "Security Hub Insight Results",
  "source": "aws.securityhub",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
      "arn:aws:securityhub:us-west-1:111122223333::product/aws/macie:us-
west-1:222233334444:test/trigger/1ec9cf700ef6be062b19584e0b7d84ec/alert/f2893b211841"
  ],
  "detail": {
    "actionName":"name of the action",
    "actionDescription":"description of the action",
    "insightArn":"ARN of the insight",
    "insightName":"Name of the insight",
    "resultType":"ResourceAwsIamAccessKeyUserName",
    "number of results":"number of results, max of 100",
    "insightResults": [
        {"result 1": 5},
        {"result 2": 6}
    ]
  }
}
```

For information about how to create an EventBridge rule that is triggered by these events, see the section called “Configuring and using custom actions”.

Configuring an EventBridge rule for Security Hub CSPM findings

You can create a rule in Amazon EventBridge that defines an action to take when a **Security Hub Findings - Imported** event is received. **Security Hub Findings - Imported** events are triggered by updates from both the `BatchImportFindings` and `BatchUpdateFindings` operations.

Each rule contains an event pattern, which identifies the events that trigger the rule. The event pattern always contains the event source (`aws.securityhub`) and the event type (**Security Hub Findings - Imported**). The event pattern can also specify filters to identify the findings that the rule applies to.

The event rule then identifies the rule targets. The targets are the actions to take when EventBridge receives a **Security Hub Findings - Imported** event and the finding matches the filters.

The instructions provided here use the EventBridge console. When you use the console, EventBridge automatically creates the required resource-based policy that enables EventBridge to write to Amazon CloudWatch Logs.

You can also use the `PutRule` operation of the EventBridge API. However, if you use the EventBridge API, then you must create the resource-based policy. For information about the required policy, see CloudWatch Logs permissions in the *Amazon EventBridge User Guide*.

Format of the event pattern

The format of the event pattern for **Security Hub Findings - Imported** events is as follows:

```
{
  "source": [
    "aws.securityhub"
  ],
  "detail-type": [
    "Security Hub Findings - Imported"
  ],
  "detail": {
    "findings": {
      <attribute filter values>
    }
  }
}
```

- `source` identifies Security Hub CSPM as the service that generates the event.
- `detail-type` identifies the type of event.
- `detail` is optional and provides the filter values for the event pattern. If the event pattern does not contain a `detail` field, then all findings trigger the rule.

You can filter the findings based on any finding attribute. For each attribute, you provide a comma-separated array of one or more values.

```
"<attribute name>": [ "<value1>", "<value2>"]
```

If you provide more than one value for an attribute, then those values are joined by OR. A finding matches the filter for an individual attribute if the finding has any of the listed values. For example, if you provide both INFORMATIONAL and LOW as values for Severity.Label, then the finding matches if it has a severity label of either INFORMATIONAL or LOW.

The attributes are joined by AND. A finding matches if it matches the filter criteria for all of the provided attributes.

When you provide an attribute value, it must reflect the location of that attribute within the AWS Security Finding Format (ASFF) structure.

Tip

When filtering control findings, we recommend using the SecurityControlId or SecurityControlArn ASFF fields as filters, rather than Title or Description. The latter fields can change occasionally, whereas the control ID and ARN are static identifiers.

In the following example, the event pattern provides filter values for ProductArn and Severity.Label, so a finding matches if it is generated by Amazon Inspector and it has a severity label of either INFORMATIONAL or LOW.

```
{
    "source": [
        "aws.securityhub"
     ],
    "detail-type": [
        "Security Hub Findings - Imported"
    ],
    "detail": {
        "findings": {
            "ProductArn": ["arn:aws:securityhub:us-east-1::product/aws/inspector"],
            "Severity": {
                "Label": ["INFORMATIONAL", "LOW"]
            }
        }
    }
}
```

Creating an event rule

You can use a predefined event pattern or a custom event pattern to create a rule in EventBridge. If you select a predefined pattern, EventBridge automatically fills in `source` and `detail-type`. EventBridge also provides fields to specify filter values for the following finding attributes:

- `AwsAccountId`
- `Compliance.Status`
- `Criticality`
- `ProductArn`
- `RecordState`
- `ResourceId`
- `ResourceType`
- `Severity.Label`
- `Types`
- `Workflow.Status`

To create an EventBridge rule (console)

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. Using the following values, create an EventBridge rule that monitors finding events:

   - For **Rule type**, choose **Rule with an event pattern**.
   - Choose how to build the event pattern.

To build the event pattern with...	Do this...	
A template	In the **Event pattern** section, choose the following options: • For **Event source**, choose **AWS services**. • For **AWS service**, choose **Security Hub**.	

To build the event pattern with...	Do this...	
	• For **Event type**, choose **Security Hub Findings - Imported**. • (Optional) To make the rule more specific, add filter values. For example, to limit the rule to findings with active record states, for **Specific Record state(s)**, choose **Active**.	

To build the event pattern with...	Do this...
A custom event pattern (Use a custom pattern if you want to filter findings based on attributes that do not appear in the EventBridge console.)	• In the **Event pattern** section, choose **Custom patterns (JSON editor)**, and then paste the following event pattern into the text area: `{ "source": [ "aws.securityhub" ], "detail-type": [ "Security Hub Findings - Imported" ], "detail": { "findings": { "<attribute name>": [ "<value1>", "<value2>"] } } }` • Update the event pattern to include the attribute and attribute values that you want to use as a filter. For example, to apply the rule to findings that have a verification state of `TRUE_POSITIVE` ,

To build the event pattern with...	Do this...	
	use the following pattern example: `{ "source": [ "aws.securityhub" ], "detail-type": [ "Security Hub Findings - Imported" ], "detail": { "findings": { "VerificationState": ["TRUE_POSITIVE"] } } }`	

- For **Target types**, choose **AWS service**, and for **Select a target**, choose a target such as an Amazon SNS topic or AWS Lambda function. The target is triggered when an event is received that matches the event pattern defined in the rule.

For details about creating rules, see Creating Amazon EventBridge rules that react to events in the *Amazon EventBridge User Guide*.

Using custom actions to send findings and insight results to EventBridge

To use AWS Security Hub CSPM custom actions to send findings or insight results to Amazon EventBridge, you first create the custom action in Security Hub CSPM. Then, you can define rules in EventBridge that apply to your custom actions.

You can create up to 50 custom actions.

If you enable cross-Region aggregation, and manage findings from the aggregation Region, then create custom actions in the aggregation Region.

The rule in EventBridge uses the Amazon Resource Name (ARN) from the custom action.

Creating a custom action

When you create a custom action in AWS Security Hub CSPM, you specify its name, description, and a unique identifier.

A custom action specifies which actions to take when an EventBridge event matches an EventBridge rule. Security Hub CSPM sends each finding to EventBridge as an event.

Choose your preferred method, and follow the steps to create a custom action.

Console

To create a custom action in Security Hub CSPM (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Settings** and then choose **Custom actions**.
3. Choose **Create custom action**.
4. Provide a **Name**, **Description**, and **Custom action ID** for the action.

   The **Name** must be fewer than 20 characters.

   The **Custom action ID** must be unique for each AWS account.
5. Choose **Create custom action**.
6. Make a note of the **Custom action ARN**. You need to use the ARN when you create a rule to associate with this action in EventBridge.

API

To create a custom action (API)

Use the CreateActionTarget operation. If you're using the AWS CLI, run the create-action-target command.

The following example creates a custom action to send findings to a remediation tool. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub create-action-target --name "Send to remediation" --description
 "Action to send the finding for remediation tracking" --id "Remediation"
```

Defining a rule in EventBridge

To trigger a custom action in Amazon EventBridge, you must create a corresponding rule in EventBridge. The rule definition includes the Amazon Resource Name (ARN) of the custom action.

The event pattern for a **Security Hub Findings - Custom Action** event has the following format:

```
{
  "source": [
    "aws.securityhub"
  ],
  "detail-type": [
    "Security Hub Findings - Custom Action"
  ],
  "resources": [ "<custom action ARN>" ]
}
```

The event pattern for a **Security Hub Insight Results** event has the following format:

```
{
  "source": [
    "aws.securityhub"
  ],
  "detail-type": [
    "Security Hub Insight Results"
  ],
  "resources": [ "<custom action ARN>" ]
}
```

In both patterns, *<custom action ARN>* is the ARN of a custom action. You can configure a rule that applies to more than one custom action.

The instructions provided here are for the EventBridge console. When you use the console, EventBridge automatically creates the required resource-based policy that enables EventBridge to write to CloudWatch Logs.

You can also use the `PutRule` API operation of the EventBridge API. However, if you use the EventBridge API, then you must create the resource-based policy. For details on the required policy, see CloudWatch Logs permissions in the *Amazon EventBridge User Guide*.

To define a rule in EventBridge (EventBridge console)

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. In the navigation pane, choose **Rules**.
3. Choose **Create rule**.
4. Enter a name and description for the rule.
5. For **Event bus**, choose the event bus that you want to associate with this rule. If you want this rule to match events that come from your account, select **default**. When an AWS service in your account emits an event, it always goes to your account's default event bus.
6. For **Rule type**, choose **Rule with an event pattern**.
7. Choose **Next**.
8. For **Event source**, choose **AWS events**.
9. For **Event pattern**, choose **Event pattern form**.
10. For **Event source**, choose **AWS services**.
11. For **AWS service**, choose **Security Hub**.
12. For **Event type**, do one of the following:
    - To create a rule to apply when you send findings to a custom action, choose **Security Hub Findings - Custom Action**.
    - To create a rule to apply when you send insight results to a custom action, choose **Security Hub Insight Results**.
13. Choose **Specific custom action ARNs**, add a custom action ARN.

    If the rule applies to multiple custom actions, choose **Add** to add more custom action ARNs.
14. Choose **Next**.
15. Under **Select targets**, choose and configure the target to invoke when this rule is matched.

16. Choose **Next**.
17. (Optional) Enter one or more tags for the rule. For more information, see Amazon EventBridge tags in the *Amazon EventBridge User Guide*.
18. Choose **Next**.
19. Review the details of the rule and choose **Create rule**.

    When you perform a custom action on findings or insight results in your account, events are generated in EventBridge.

Selecting a custom action for findings and insight results

After you create AWS Security Hub CSPM custom actions and Amazon EventBridge rules, you can send findings and insight results to EventBridge for automatic management and processing.

Events are sent to EventBridge only in the account in which they are viewed. If you view a finding using an administrator account, the event is sent to EventBridge in the administrator account.

For AWS API calls to be effective, the implementations of target code must switch roles into member accounts. This also means that the role you switch into must be deployed to each member where action is needed.

To send findings to EventBridge (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Display a list of findings:
   - From **Findings**, you can view findings from all of the enabled product integrations and controls.
   - From **Security standards**, you can navigate to a list of findings generated from a specific control. For more information, see Reviewing the details of controls in Security Hub CSPM.
   - From **Integrations**, you can navigate to a list of findings generated by an enabled integration. For more information, see Viewing findings from a Security Hub CSPM integration.
   - From **Insights**, you can navigate to a list of findings for an insight result. For more information, see Reviewing and acting on insights in Security Hub CSPM.
3. Select the findings to send to EventBridge. You can select up to 20 findings at a time.
4. From **Actions**, choose the custom action that aligns with the EventBridge rule to apply.

Security Hub CSPM sends a separate **Security Hub Findings - Custom Action** event for each finding.

To send insight results to EventBridge (console)

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Insights**.
3. On the **Insights** page, choose the insight that includes the results to send to EventBridge.
4. Select the insight results to send to EventBridge. You can select up to 20 results at a time.
5. From **Actions**, choose the custom action that aligns with the EventBridge rule to apply.

Working with the dashboard in Security Hub CSPM

On the Security Hub CSPM console, the **Summary** dashboard shows a summary of your risks, attack sequences, and security coverage. This dashboard helps you identify risks and attack sequences based on severity and account coverage for different security capabilities. Each time you open the dashboard, it refreshes automatically. Note, however, that security scores and control statuses refresh every 24 hours.

You can customize the **Summary** dashboard by adding and removing different security widgets from it. You can also specify filter criteria to retrieve and display particular types of data. If you customize the dashboard, Security Hub saves your customization settings. If other users of your account customize the dashboard, their changes are saved independently from your customization settings.

If you configured cross-Region aggregation in Security Hub CSPM, the **Summary** dashboard shows your aggregated data. If your account is the delegated administrator account for an organization, the data includes findings for your account and your member accounts. If your account is a member account or a standalone account, the data includes findings only for your account.

Topics

Available widgets for the Summary dashboard

The **Summary** dashboard includes widgets that reflect the modern cloud security threat landscape, guided by the security operations and experiences of AWS customers. Some widgets are shown by default while others are not. You can customize your view of the dashboard by adding or removing widgets.

To add a widget, choose **Add widget** at the top of the dashboard. You can then browse the list of available widgets or enter the title of a widget in the search bar. When you find the widget to add, drag it to the location where you want it to appear on the dashboard. For more information, see Customizing the dashboard.

Widgets shown by default

By default, the **Summary** dashboard includes the following widgets.

Top threat sequences

Displays the highest severity threat sequences. Threat sequence findings, known as *attack sequence findings* in Amazon GuardDuty, correlate multiple events to identify potential threats to your AWS environment. Threat sequences may include in-progress or recent attack behaviors (within a 24-hour time window) in your environment, which may in turn lead to further compromise. You must have GuardDuty and GuardDuty S3 Protection enabled to receive threat sequence findings in Security Hub CSPM.

Top risks

Displays a summary of the top risks in your environment. The top of the widget shows you the count of risks at each severity level. You can choose a severity level to go to the **Risks** page with risks filtered to the selected severity level. Risks that have the most occurrences in your environment appear first. This widget helps you prioritize which risks to mitigate.

Security coverage

Summarizes the extent of your security coverage, based on coverage control findings. Coverage controls check whether a specific AWS service and its capabilities are enabled (for example, [Macie.1] Amazon Macie should be enabled). This widget helps you ensure that you have PASSED findings for coverage controls. The Security Hub CSPM console provides links from this widget to help you enable missing security capabilities. We recommend using central configuration to enable missing security capabilities across multiple AWS accounts and AWS Regions. For more information, see Understanding central configuration in Security Hub CSPM.

Security standards

Displays your most recent summary security score and the security score for each Security Hub CSPM standard. Security scores, which range from 0–100 percent, represent the proportion of passed controls relative to all of your enabled controls. For more information about these scores, see Method of calculating security scores. This widget helps you understand your overall security posture.

Assets with the most findings

Provides an overview of the resources, accounts, and applications that have the most findings. The list is sorted in descending order by the number of findings. In the widget, each tab shows the top six items in that category, grouped by severity and resource type. If you choose a number in the **Total findings** column, Security Hub CSPM opens a page that shows the findings for the asset. This widget helps you quickly identify which of your core assets have potential security threats.

Findings by Region

Shows the total number of findings, grouped by severity, in each AWS Region in which Security Hub CSPM is enabled. This widget helps you identify security issues that potentially affect particular Regions. If you open the dashboard in your aggregation Region, this widget helps you monitor potential security issues in each linked Region.

Most common threat types

Provides a breakdown of the 10 most common types of threats in your AWS environment. This includes threats such as escalation of privileges, use of exposed credentials, or communication with malicious IP addresses.

To view this data, Amazon GuardDuty must be enabled. If it is, choose a threat type in this widget to open the GuardDuty console and review findings related to this threat. This widget helps you evaluate potential threats in the context of other security issues.

Software vulnerabilities with exploits

Provides a summary of software vulnerabilities that exist in your AWS environment and have known exploits. You can also review a breakdown of vulnerabilities that do and don't have fixes available.

To view this data, Amazon Inspector must be enabled. If it is, choose a statistic in this widget to open the Amazon Inspector console and review more details about the vulnerability. This widget helps you evaluate software vulnerabilities in the context of other security issues.

New findings over time

Shows trends in the number of new daily findings during the past 90 days. You can break down the data by severity or by provider for additional context. This widget helps you understand if finding volume spiked or dropped at specific times during the past 90 days.

Resources with the most findings

Provides a summary of the resources that have generated the most findings, broken down by the following resource types: Amazon Simple Storage Service (Amazon S3) buckets, Amazon Elastic Compute Cloud (Amazon EC2) instances, and AWS Lambda functions.

In the widget, each tab focuses on one of the preceding resource types, listing the 10 resource instances that generated the most findings. To review the findings for a specific resource, choose the resource instance. This widget helps you triage security findings that are associated with common AWS resources.

Widgets hidden by default

The following widgets are also available for the **Summary** dashboard, but they are hidden by default.

AMIs with the most findings

Provides a list of the 10 Amazon Machine Images (AMIs) that have generated the most findings. This data is available only if Amazon EC2 is enabled for your account. It helps you identify which AMIs pose potential security risks.

IAM principals with the most findings

Provides a list of the 10 AWS Identity and Access Management (IAM) users that have generated the most findings. This widget helps you perform administrative and billing tasks. It shows you which users contribute to Security Hub CSPM usage the most.

Accounts with the most findings (by severity)

Shows a graph of the 10 accounts that have generated the most findings, grouped by severity. This widget helps you determine which accounts to focus analysis and remediation efforts on.

Accounts with the most findings (by resource type)

Shows a graph of the 10 accounts that have generated the most findings, grouped by resource type. This widget helps you determine which accounts and resource types to prioritize for analysis and remediation.

Insights

Lists five Security Hub CSPM managed insights and the number of findings that they generated. Insights identify a specific security area that requires attention.

Latest findings from AWS integrations

Shows the number of findings that you received in Security Hub CSPM from integrated AWS services. It also shows when you most recently received findings from each integrated service. This widget provides consolidated findings data from multiple AWS services. To drill down, choose an integrated service. Security Hub CSPM then opens the console for that service.

Filtering the Summary dashboard in Security Hub CSPM

You can curate the **Summary** dashboard on the Security Hub CSPM console so that it includes only the security data that's most relevant to you. For example, if you're a member of an application team, you might create a dedicated view for a critical application in your production environment. If you're a member of a security team, you might create a dedicated view that helps you focus on high-severity findings.

To create these curated views, enter filter criteria in the filter box above the dashboard. If you apply filter criteria, the criteria apply to all the data and widgets on the dashboard, except the data in the **Insights** and **Security standards** widgets. For a list of available widgets on the dashboard, see Available widgets for the Summary dashboard.

You can filter the data by using the following fields:

- Account name
- Account ID
- Application ARN
- Application name
- Product name (for an AWS service or third-party product that sends findings to Security Hub CSPM)

- Record state
- Region
- Resource tag
- Severity
- Workflow status

By default, dashboard data is filtered using the following criteria: `Workflow.Status` is `NOTIFIED` or `NEW`, and `RecordState` is `ACTIVE`. These criteria appear above the dashboard, below the filter box. To remove these criteria, choose **X** in the filter token for the criteria that you want to remove.

If you apply filter criteria that you want to use again, you can save it as a *filter set*. A filter set is a set of filter criteria that you create and save to reapply when you review data on the **Summary** dashboard. You can create and save a filter set that uses any of the available fields except the following fields: Application ARN, application name, and resource tag.

Creating and saving filter sets

Follow these steps to create and save a filter set.

To create and save a filter set

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Summary**.
3. In the filter box above the **Summary** dashboard, enter the filter criteria for the filter set.
4. On the **Clear filters** menu, choose **Save new filter set**.
5. In the **Save filter set** dialog box, enter a name for the filter set.
6. (Optional) To use the filter set by default each time you open the **Summary** page, select the option to set it as the default view.
7. Choose **Save**.

To switch between filter sets that you've created and saved, use the **Choose a filter set** menu above the **Summary** dashboard. When you select a filter set, Security Hub CSPM applies the criteria of the filter set to the data on the dashboard.

Updating or deleting filter sets

Follow these steps to update or delete an existing filter set. If you delete a filter set that is currently set as your default view of the **Summary** dashboard, your default view is reset to the default Security Hub CSPM view.

To update or delete a filter set

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Summary**.
3. In the **Choose a filter set** menu above the **Summary** page, choose the filter set.
4. On the **Clear filters** menu, do one of the following:
   - To update the filter set, choose **Update current filter set**. Then, enter your changes in the dialog box that appears.
   - To delete the filter set choose **Delete current filter set**. Then, choose **Delete** in the dialog box that appears.

Customizing the Summary dashboard in Security Hub CSPM

You can customize the **Summary** dashboard on the Security Hub CSPM console in several ways. For example, you can add and remove widgets from the dashboard. You can also rearrange and resize widgets on the dashboard. For a list of available widgets and a description of each one, see Available widgets for the Summary dashboard.

If you customize the dashboard, Security Hub CSPM applies your changes immediately and saves your new dashboard settings. Your changes apply to your view of the dashboard in all AWS Regions and browsers.

To customize the Summary dashboard

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Summary**.
3. Do any of the following:
   - To add a widget, choose **Add widgets** at the upper-right corner of the page. In the search bar, enter the title of the widget to add. Then, drag the widget to the location that you want.

- To remove a widget, choose the three dots in the upper-right corner of the widget.
- To move a widget, choose the handle at the upper-left corner of the widget, and then drag the widget to the location that you want.
- To change the size of a widget, choose the resize handle at the lower-right corner of the widget. Drag the widget's edge until the widget is your preferred size.

To subsequently restore the original settings, choose **Reset to default layout** at the top of the page.

Regional limits for Security Hub CSPM

Some AWS Security Hub CSPM features are available in only certain AWS Regions. The following sections specify these Regional limits. For a complete list of all the Regions where Security Hub CSPM is currently available, see AWS Security Hub endpoints and quotas in the *AWS General Reference*.

Cross-Region aggregation restrictions

In AWS GovCloud (US) Regions, cross-Region aggregation is available for findings, finding updates, and insights across AWS GovCloud (US) Regions only. Specifically, you can aggregate findings, finding updates, and insights only between the AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions.

In the China Regions, cross-Region aggregation is available for findings, finding updates, and insights across the China Regions only. Specifically, you can aggregate findings, finding updates, and insights only between the China (Beijing) and China (Ningxia) Regions.

You can't use a Region that's disabled by default as your aggregation Region. For a list of Regions that are disabled by default, see Enable or disable AWS Regions in your account in the *AWS Account Management Reference Guide*.

Availability of integrations by Region

Some integrations aren't available in all AWS Regions. On the Security Hub CSPM console, an integration doesn't appear on the **Integrations** page if it isn't available in the Region that you're currently signed in to.

Integrations supported in the China (Beijing) and China (Ningxia) Regions

In the China (Beijing) and China (Ningxia) Regions, Security Hub CSPM supports only the following integrations with AWS services:

- AWS Firewall Manager
- Amazon GuardDuty
- AWS Identity and Access Management Access Analyzer
- Amazon Inspector
- AWS IoT Device Defender
- AWS Systems Manager Explorer
- AWS Systems Manager OpsCenter
- AWS Systems Manager Patch Manager

In the China (Beijing) and China (Ningxia) Regions, Security Hub CSPM supports only the following third-party integrations:

- Cloud Custodian
- FireEye Helix
- Helecloud
- IBM QRadar
- PagerDuty
- Palo Alto Networks Cortex XSOAR
- Palo Alto Networks VM-Series
- Prowler
- RSA Archer
- Splunk Enterprise
- Splunk Phantom
- ThreatModeler

Integrations supported in the AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions

In the AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions, Security Hub CSPM supports only the following integrations with AWS services:

- AWS Config
- Amazon Detective
- AWS Firewall Manager
- Amazon GuardDuty
- AWS Health
- IAM Access Analyzer
- Amazon Inspector
- AWS IoT Device Defender

In the AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions, Security Hub CSPM supports only the following third-party integrations:

- Atlassian Jira Service Management
- Atlassian Jira Service Management Cloud
- Atlassian OpsGenie
- Caveonix Cloud
- Cloud Custodian
- Cloud Storage Security Antivirus for Amazon S3
- CrowdStrike Falcon
- FireEye Helix
- Forcepoint CASB
- Forcepoint DLP
- Forcepoint NGFW
- Fugue
- Kion
- MicroFocus ArcSight
- NETSCOUT Cyber Investigator

- PagerDuty
- Palo Alto Networks – Prisma Cloud Compute
- Palo Alto Networks – Prisma Cloud Enterprise
- Palo Alto Networks – VM-Series (available only in AWS GovCloud (US-West))
- Prowler
- Rackspace Technology – Cloud Native Security
- Rapid7 InsightConnect
- RSA Archer
- SecureCloudDb
- ServiceNow ITSM
- Slack
- ThreatModeler
- Vectra AI Cognito Detect

Availability of standards by Region

The AWS Control Tower service-managed standard is available only in AWS Regions that AWS Control Tower supports, including AWS GovCloud (US) Regions. For a list of Regions that AWS Control Tower currently supports, see How AWS Regions Work With AWS Control Tower in the *AWS Control Tower User Guide*.

The AWS Resource Tagging standard isn't available in the Asia Pacific (Taipei) Region.

Other security standards are available in all the Regions where Security Hub CSPM is currently available.

Availability of controls by Region

Some Security Hub CSPM controls aren't available in all AWS Regions. For a list of controls that aren't available in each Region, see Regional limits on Security Hub CSPM controls.

On the Security Hub CSPM console, a control doesn't appear in the list of controls if it isn't available in the Region that you're currently signed in to. The exception is an aggregation Region. If you set an aggregation Region and sign in to that Region, the console shows controls that are available in the aggregation Region or one or more linked Regions.

Regional limits on Security Hub CSPM controls

Some AWS Security Hub CSPM controls aren't available in all AWS Regions. This page specifies which controls aren't available in specific Regions.

On the Security Hub CSPM console, a control doesn't appear in the list of controls if it isn't available in the Region that you're currently signed in to. The exception is an aggregation Region. If you set an aggregation Region and sign in to that Region, the console shows controls that are available in the aggregation Region or one or more linked Regions.

AWS Regions

- US East (N. Virginia)
- US East (Ohio)
- US West (N. California)
- US West (Oregon)
- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacific (Jakarta)
- Asia Pacific (Malaysia)
- Asia Pacific (Melbourne)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Taipei)
- Asia Pacific (Thailand)
- Asia Pacific (Tokyo)
- Canada (Central)
- Canada West (Calgary)
- China (Beijing)
- China (Ningxia)

- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Milan)
- Europe (Paris)
- Europe (Spain)
- Europe (Stockholm)
- Europe (Zurich)
- Israel (Tel Aviv)
- Mexico (Central)
- Middle East (Bahrain)
- Middle East (UAE)
- South America (São Paulo)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

US East (N. Virginia)

The following controls are not supported in the US East (N. Virginia) Region.

- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled

US East (Ohio)

The following controls are not supported in the US East (Ohio) Region.

- [AppSync.1] AWS AppSync API caches should be encrypted at rest

- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged

- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [RDS.31] RDS DB security groups should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

US West (N. California)

The following controls are not supported in the US West (N. California) Region.

- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins

- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged

- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

US West (Oregon)

The following controls are not supported in the US West (Oregon) Region.

- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured

- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Africa (Cape Town)

The following controls are not supported in the Africa (Cape Town) Region.

- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged

- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit

- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [ECR.4] ECR public repositories should be tagged
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ES.3] Elasticsearch domains should encrypt data sent between nodes
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged

- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [RDS.1] RDS snapshot should be private
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username

- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Asia Pacific (Hong Kong)

The following controls are not supported in the Asia Pacific (Hong Kong) Region.

- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins

- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged

- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled

- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Asia Pacific (Hyderabad)

The following controls are not supported in the Asia Pacific (Hyderabad) Region.

- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests

- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit

- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.175] EC2 launch templates should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.4] ECR public repositories should be tagged
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group

- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.2] GuardDuty filters should be tagged
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled

- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled

- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.4] Amazon MQ brokers should be tagged
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed

- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [SQS.1] Amazon SQS queues should be encrypted at rest

- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Asia Pacific (Jakarta)

The following controls are not supported in the Asia Pacific (Jakarta) Region.

- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Backup.1] AWS Backup recovery points should be encrypted at rest

- [Backup.4] AWS Backup report plans should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [CodeBuild.3] CodeBuild S3 logs should be encrypted
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged

- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [ECR.4] ECR public repositories should be tagged
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS

- [EFS.2] Amazon EFS volumes should be in backup plans
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [GuardDuty.2] GuardDuty filters should be tagged
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged

- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [Opensearch.5] OpenSearch domains should have audit logging enabled

- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Asia Pacific (Malaysia)

The following controls are not supported in the Asia Pacific (Malaysia) Region.

- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits

- [Account.1] Security contact information should be provided for an AWS account
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.4] AWS AppSync GraphQL APIs should be tagged
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Athena.2] Athena data catalogs should be tagged
- [Athena.3] Athena workgroups should be tagged
- [Athena.4] Athena workgroups should have logging enabled
- [AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones
- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones
- [AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.2] AWS Backup recovery points should be tagged
- [Backup.3] AWS Backup vaults should be tagged
- [Backup.4] AWS Backup report plans should be tagged

- [Backup.5] AWS Backup backup plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.2] Batch scheduling policies should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFormation.2] CloudFormation stacks should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket
- [CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys
- [CloudWatch.17] CloudWatch alarm actions should be activated
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [CodeBuild.3] CodeBuild S3 logs should be encrypted

- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeBuild.7] CodeBuild report group exports should be encrypted at rest
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DataFirehose.1] Firehose delivery streams should be encrypted at rest
- [DataSync.1] DataSync tasks should have logging enabled
- [DataSync.2] DataSync tasks should be tagged
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit

- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [DynamoDB.6] DynamoDB tables should have deletion protection enabled
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.33] EC2 transit gateway attachments should be tagged
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.52] EC2 transit gateways should be tagged
- [EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports
- [EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports
- [EC2.55] VPCs should be configured with an interface endpoint for ECR API
- [EC2.56] VPCs should be configured with an interface endpoint for Docker Registry
- [EC2.57] VPCs should be configured with an interface endpoint for Systems Manager
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.171] EC2 VPN connections should have logging enabled

- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.174] EC2 DHCP option sets should be tagged
- [EC2.175] EC2 launch templates should be tagged
- [EC2.176] EC2 prefix lists should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.178] EC2 traffic mirror filters should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECR.4] ECR public repositories should be tagged
- [ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys
- [ECS.3] ECS task definitions should not share the host's process namespace
- [ECS.4] ECS containers should run as non-privileged
- [ECS.5] ECS containers should be limited to read-only access to root filesystems
- [ECS.8] Secrets should not be passed as container environment variables
- [ECS.9] ECS task definitions should have a logging configuration
- [ECS.10] ECS Fargate services should run on the latest Fargate platform version
- [ECS.12] ECS clusters should use Container Insights
- [ECS.16] ECS task sets should not automatically assign public IP addresses
- [ECS.17] ECS task definitions should not use host network mode
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EFS.5] EFS access points should be tagged
- [EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch
- [EFS.7] EFS file systems should have automatic backups enabled

- [EFS.8] EFS file systems should be encrypted at rest
- [EKS.2] EKS clusters should run on a supported Kubernetes version
- [EKS.3] EKS clusters should use encrypted Kubernetes secrets
- [EKS.6] EKS clusters should be tagged
- [EKS.7] EKS identity provider configurations should be tagged
- [EKS.8] EKS clusters should have audit logging enabled
- [ELB.10] Classic Load Balancer should span multiple Availability Zones
- [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [EMR.2] Amazon EMR block public access setting should be enabled
- [EMR.3] Amazon EMR security configurations should be encrypted at rest

- [EMR.4] Amazon EMR security configurations should be encrypted in transit
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [ES.9] Elasticsearch domains should be tagged
- [EventBridge.2] EventBridge event buses should be tagged
- [EventBridge.3] EventBridge custom event buses should have a resource-based policy attached
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment
- [FSx.4] FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment
- [FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.1] AWS Glue jobs should be tagged
- [Glue.3] AWS Glue machine learning transforms should be encrypted at rest
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.2] GuardDuty filters should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [GuardDuty.4] GuardDuty detectors should be tagged
- [GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled
- [GuardDuty.6] GuardDuty Lambda Protection should be enabled
- [GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled
- [GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled
- [GuardDuty.9] GuardDuty RDS Protection should be enabled
- [GuardDuty.10] GuardDuty S3 Protection should be enabled
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled

- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.9] MFA should be enabled for the root user
- [IAM.10] Password policies for IAM users should have strong configurations
- [IAM.11] Ensure IAM password policy requires at least one uppercase letter
- [IAM.12] Ensure IAM password policy requires at least one lowercase letter
- [IAM.13] Ensure IAM password policy requires at least one symbol
- [IAM.14] Ensure IAM password policy requires at least one number
- [IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
- [IAM.16] Ensure IAM password policy prevents password reuse
- [IAM.17] Ensure IAM password policy expires passwords within 90 days or less
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.23] IAM Access Analyzer analyzers should be tagged
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled

- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [Kinesis.2] Kinesis streams should be tagged
- [Kinesis.3] Kinesis streams should have an adequate data retention period

- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [KMS.5] KMS keys should not be publicly accessible
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.4] Amazon MQ brokers should be tagged
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones
- [NetworkFirewall.2] Network Firewall logging should be enabled

- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets
- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [PCA.2] AWS Private CA certificate authorities should be tagged
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.18] RDS instances should be deployed in a VPC
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.27] RDS DB clusters should be encrypted at rest
- [RDS.31] RDS DB security groups should be tagged
- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled

- [RDS.36] RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [RDS.38] RDS for PostgreSQL DB instances should be encrypted in transit
- [RDS.39] RDS for MySQL DB instances should be encrypted in transit
- [RDS.40] RDS for SQL Server DB instances should publish logs to CloudWatch Logs
- [RDS.41] RDS for SQL Server DB instances should be encrypted in transit
- [RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs
- [RDS.44] RDS for MariaDB DB instances should be encrypted in transit
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.1] Amazon Redshift clusters should prohibit public access
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.11] Redshift clusters should be tagged
- [Redshift.13] Redshift cluster snapshots should be tagged
- [Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins
- [Redshift.16] Redshift cluster subnet groups should have subnets from multiple Availability Zones
- [Redshift.17] Redshift cluster parameter groups should be tagged
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username

- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.7] S3 general purpose buckets should use cross-Region replication
- [S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.12] ACLs should not be used to manage user access to S3 general purpose buckets
- [S3.13] S3 general purpose buckets should have Lifecycle configurations
- [S3.19] S3 access points should have block public access settings enabled
- [S3.20] S3 general purpose buckets should have MFA delete enabled
- [S3.22] S3 general purpose buckets should log object-level write events
- [S3.23] S3 general purpose buckets should log object-level read events
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SageMaker.8] SageMaker notebook instances should run on supported platforms
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SNS.4] SNS topic access policies should not allow public access
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access

- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.4] SSM documents should not be public
- [SSM.5] SSM documents should be tagged
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [StepFunctions.2] Step Functions activities should be tagged
- [Transfer.1] AWS Transfer Family workflows should be tagged
- [Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [Transfer.5] Transfer Family certificates should be tagged
- [Transfer.6] Transfer Family connectors should be tagged
- [Transfer.7] Transfer Family profiles should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.2] AWS WAF Classic Regional rules should have at least one condition
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Asia Pacific (Melbourne)

The following controls are not supported in the Asia Pacific (Melbourne) Region.

- [APIGateway.8] API Gateway routes should specify an authorization type

- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged

- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports
- [EC2.22] Unused Amazon EC2 security groups should be removed

- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.175] EC2 launch templates should be tagged
- [ECR.4] ECR public repositories should be tagged
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group

- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.2] GuardDuty filters should be tagged
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.10] Password policies for IAM users should have strong configurations
- [IAM.11] Ensure IAM password policy requires at least one uppercase letter
- [IAM.12] Ensure IAM password policy requires at least one lowercase letter
- [IAM.13] Ensure IAM password policy requires at least one symbol
- [IAM.14] Ensure IAM password policy requires at least one number
- [IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
- [IAM.16] Ensure IAM password policy prevents password reuse
- [IAM.17] Ensure IAM password policy expires passwords within 90 days or less

- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged

- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled

- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [RDS.1] RDS snapshot should be private
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.31] RDS DB security groups should be tagged
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.5] SageMaker models should have network isolation enabled

- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SageMaker.8] SageMaker notebook instances should run on supported platforms
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.4] SSM documents should not be public
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Asia Pacific (Mumbai)

The following controls are not supported in the Asia Pacific (Mumbai) Region.

- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled

- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [RDS.31] RDS DB security groups should be tagged
- [Route53.1] Route 53 health checks should be tagged

- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Asia Pacific (Osaka)

The following controls are not supported in the Asia Pacific (Osaka) Region.

- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged

- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.55] VPCs should be configured with an interface endpoint for ECR API
- [EC2.56] VPCs should be configured with an interface endpoint for Docker Registry
- [EC2.57] VPCs should be configured with an interface endpoint for Systems Manager
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts

- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination
- [ELB.4] Application Load Balancer should be configured to drop invalid http headers
- [ELB.6] Application, Gateway, and Network Load Balancers should have deletion protection enabled
- [ELB.8] Classic Load Balancers with SSL listeners should use a predefined security policy that has strong AWS Configuration
- [ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.4] IAM root user access key should not exist
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged

- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [RDS.31] RDS DB security groups should be tagged
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL

- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Asia Pacific (Seoul)

The following controls are not supported in the Asia Pacific (Seoul) Region.

- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit

- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.4] ECR public repositories should be tagged
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged

- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [RDS.31] RDS DB security groups should be tagged
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Asia Pacific (Singapore)

The following controls are not supported in the Asia Pacific (Singapore) Region.

- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured

- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule

- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Asia Pacific (Sydney)

The following controls are not supported in the Asia Pacific (Sydney) Region.

- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries

- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Asia Pacific (Taipei)

The following controls are not supported in the Asia Pacific (Taipei) Region.

- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [ACM.3] ACM certificates should be tagged
- [Account.1] Security contact information should be provided for an AWS account
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.1] API Gateway REST and WebSocket API execution logging should be enabled
- [APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication
- [APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled
- [APIGateway.4] API Gateway should be associated with a WAF Web ACL
- [APIGateway.5] API Gateway REST API cache data should be encrypted at rest
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged

- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.4] AWS AppSync GraphQL APIs should be tagged
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Athena.2] Athena data catalogs should be tagged
- [Athena.3] Athena workgroups should be tagged
- [Athena.4] Athena workgroups should have logging enabled
- [AutoScaling.1] Auto Scaling groups associated with a load balancer should use ELB health checks
- [AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones
- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones
- [AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates
- [AutoScaling.10] EC2 Auto Scaling groups should be tagged
- [Autoscaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.2] AWS Backup recovery points should be tagged
- [Backup.3] AWS Backup vaults should be tagged
- [Backup.4] AWS Backup report plans should be tagged
- [Backup.5] AWS Backup backup plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.2] Batch scheduling policies should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFormation.2] CloudFormation stacks should be tagged

- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket
- [CloudTrail.9] CloudTrail trails should be tagged
- [CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys
- [CloudWatch.17] CloudWatch alarm actions should be activated
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [CodeBuild.3] CodeBuild S3 logs should be encrypted
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeBuild.7] CodeBuild report group exports should be encrypted at rest
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication

- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DataFirehose.1] Firehose delivery streams should be encrypted at rest
- [DataSync.1] DataSync tasks should have logging enabled
- [DataSync.2] DataSync tasks should be tagged
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.1] Database Migration Service replication instances should not be public
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [DynamoDB.5] DynamoDB tables should be tagged
- [DynamoDB.6] DynamoDB tables should have deletion protection enabled

- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.10] Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service
- [EC2.19] Security groups should not allow unrestricted access to ports with high risk
- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.33] EC2 transit gateway attachments should be tagged
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.35] EC2 network interfaces should be tagged
- [EC2.36] EC2 customer gateways should be tagged
- [EC2.37] EC2 Elastic IP addresses should be tagged
- [EC2.38] EC2 instances should be tagged
- [EC2.39] EC2 internet gateways should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.41] EC2 network ACLs should be tagged
- [EC2.42] EC2 route tables should be tagged
- [EC2.43] EC2 security groups should be tagged
- [EC2.44] EC2 subnets should be tagged
- [EC2.45] EC2 volumes should be tagged
- [EC2.46] Amazon VPCs should be tagged
- [EC2.47] Amazon VPC endpoint services should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged
- [EC2.49] Amazon VPC peering connections should be tagged
- [EC2.50] EC2 VPN gateways should be tagged
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled

- [EC2.52] EC2 transit gateways should be tagged
- [EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports
- [EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports
- [EC2.55] VPCs should be configured with an interface endpoint for ECR API
- [EC2.56] VPCs should be configured with an interface endpoint for Docker Registry
- [EC2.57] VPCs should be configured with an interface endpoint for Systems Manager
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.171] EC2 VPN connections should have logging enabled
- [EC2.172] EC2 VPC Block Public Access settings should block internet gateway traffic
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.174] EC2 DHCP option sets should be tagged
- [EC2.175] EC2 launch templates should be tagged
- [EC2.176] EC2 prefix lists should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.178] EC2 traffic mirror filters should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECR.4] ECR public repositories should be tagged
- [ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys
- [ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions
- [ECS.2] ECS services should not have public IP addresses assigned to them automatically

- [ECS.3] ECS task definitions should not share the host's process namespace
- [ECS.4] ECS containers should run as non-privileged
- [ECS.5] ECS containers should be limited to read-only access to root filesystems
- [ECS.8] Secrets should not be passed as container environment variables
- [ECS.9] ECS task definitions should have a logging configuration
- [ECS.10] ECS Fargate services should run on the latest Fargate platform version
- [ECS.12] ECS clusters should use Container Insights
- [ECS.13] ECS services should be tagged
- [ECS.14] ECS clusters should be tagged
- [ECS.15] ECS task definitions should be tagged
- [ECS.16] ECS task sets should not automatically assign public IP addresses
- [ECS.17] ECS task definitions should not use host network mode
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EFS.5] EFS access points should be tagged
- [EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch
- [EFS.7] EFS file systems should have automatic backups enabled
- [EFS.8] EFS file systems should be encrypted at rest
- [EKS.1] EKS cluster endpoints should not be publicly accessible
- [EKS.2] EKS clusters should run on a supported Kubernetes version
- [EKS.3] EKS clusters should use encrypted Kubernetes secrets
- [EKS.6] EKS clusters should be tagged
- [EKS.7] EKS identity provider configurations should be tagged
- [EKS.8] EKS clusters should have audit logging enabled
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination
- [ELB.7] Classic Load Balancers should have connection draining enabled

- [ELB.8] Classic Load Balancers with SSL listeners should use a predefined security policy that has strong AWS Configuration
- [ELB.10] Classic Load Balancer should span multiple Availability Zones
- [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [EMR.2] Amazon EMR block public access setting should be enabled
- [EMR.3] Amazon EMR security configurations should be encrypted at rest
- [EMR.4] Amazon EMR security configurations should be encrypted in transit
- [ES.1] Elasticsearch domains should have encryption at-rest enabled
- [ES.2] Elasticsearch domains should not be publicly accessible

- [ES.3] Elasticsearch domains should encrypt data sent between nodes
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [ES.5] Elasticsearch domains should have audit logging enabled
- [ES.6] Elasticsearch domains should have at least three data nodes
- [ES.7] Elasticsearch domains should be configured with at least three dedicated master nodes
- [ES.8] Connections to Elasticsearch domains should be encrypted using the latest TLS security policy
- [ES.9] Elasticsearch domains should be tagged
- [EventBridge.2] EventBridge event buses should be tagged
- [EventBridge.3] EventBridge custom event buses should have a resource-based policy attached
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment
- [FSx.4] FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment
- [FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.1] AWS Glue jobs should be tagged
- [Glue.3] AWS Glue machine learning transforms should be encrypted at rest
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.1] GuardDuty should be enabled
- [GuardDuty.2] GuardDuty filters should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [GuardDuty.4] GuardDuty detectors should be tagged
- [GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled
- [GuardDuty.6] GuardDuty Lambda Protection should be enabled

- [GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled
- [GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled
- [GuardDuty.9] GuardDuty RDS Protection should be enabled
- [GuardDuty.10] GuardDuty S3 Protection should be enabled
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled
- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.9] MFA should be enabled for the root user
- [IAM.10] Password policies for IAM users should have strong configurations
- [IAM.11] Ensure IAM password policy requires at least one uppercase letter
- [IAM.12] Ensure IAM password policy requires at least one lowercase letter
- [IAM.13] Ensure IAM password policy requires at least one symbol
- [IAM.14] Ensure IAM password policy requires at least one number
- [IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
- [IAM.16] Ensure IAM password policy prevents password reuse
- [IAM.17] Ensure IAM password policy expires passwords within 90 days or less
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.23] IAM Access Analyzer analyzers should be tagged
- [IAM.24] IAM roles should be tagged

- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged

- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [Kinesis.2] Kinesis streams should be tagged
- [Kinesis.3] Kinesis streams should have an adequate data retention period
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [KMS.3] AWS KMS keys should not be deleted unintentionally
- [KMS.5] KMS keys should not be publicly accessible
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Lambda.6] Lambda functions should be tagged
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.4] Amazon MQ brokers should be tagged
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled

- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones
- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets
- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.7] Network Firewall firewalls should be tagged
- [NetworkFirewall.8] Network Firewall firewall policies should be tagged
- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [PCA.1] AWS Private CA root certificate authority should be disabled

- [PCA.2] AWS Private CA certificate authorities should be tagged
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.16] Aurora DB clusters should be configured to copy tags to DB snapshots
- [RDS.17] RDS DB instances should be configured to copy tags to snapshots
- [RDS.18] RDS instances should be deployed in a VPC
- [RDS.19] Existing RDS event notification subscriptions should be configured for critical cluster events
- [RDS.20] Existing RDS event notification subscriptions should be configured for critical database instance events
- [RDS.21] An RDS event notifications subscription should be configured for critical database parameter group events
- [RDS.22] An RDS event notifications subscription should be configured for critical database security group events
- [RDS.23] RDS instances should not use a database engine default port
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.27] RDS DB clusters should be encrypted at rest
- [RDS.28] RDS DB clusters should be tagged
- [RDS.29] RDS DB cluster snapshots should be tagged
- [RDS.30] RDS DB instances should be tagged
- [RDS.31] RDS DB security groups should be tagged
- [RDS.32] RDS DB snapshots should be tagged
- [RDS.33] RDS DB subnet groups should be tagged
- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.36] RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [RDS.38] RDS for PostgreSQL DB instances should be encrypted in transit
- [RDS.39] RDS for MySQL DB instances should be encrypted in transit
- [RDS.40] RDS for SQL Server DB instances should publish logs to CloudWatch Logs

- [RDS.41] RDS for SQL Server DB instances should be encrypted in transit
- [RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs
- [RDS.44] RDS for MariaDB DB instances should be encrypted in transit
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.1] Amazon Redshift clusters should prohibit public access
- [Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Redshift.4] Amazon Redshift clusters should have audit logging enabled
- [Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled
- [Redshift.7] Redshift clusters should use enhanced VPC routing
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.11] Redshift clusters should be tagged
- [Redshift.12] Redshift event notification subscriptions should be tagged
- [Redshift.13] Redshift cluster snapshots should be tagged
- [Redshift.14] Redshift cluster subnet groups should be tagged
- [Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins
- [Redshift.16] Redshift cluster subnet groups should have subnets from multiple Availability Zones
- [Redshift.17] Redshift cluster parameter groups should be tagged
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username

- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.7] S3 general purpose buckets should use cross-Region replication
- [S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.12] ACLs should not be used to manage user access to S3 general purpose buckets
- [S3.13] S3 general purpose buckets should have Lifecycle configurations
- [S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys
- [S3.19] S3 access points should have block public access settings enabled
- [S3.20] S3 general purpose buckets should have MFA delete enabled
- [S3.22] S3 general purpose buckets should log object-level write events
- [S3.23] S3 general purpose buckets should log object-level read events
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SageMaker.8] SageMaker notebook instances should run on supported platforms
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [SecretsManager.1] Secrets Manager secrets should have automatic rotation enabled
- [SecretsManager.2] Secrets Manager secrets configured with automatic rotation should rotate successfully
- [SecretsManager.3] Remove unused Secrets Manager secrets

- [SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days
- [SecretsManager.5] Secrets Manager secrets should be tagged
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SNS.3] SNS topics should be tagged
- [SNS.4] SNS topic access policies should not allow public access
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.1] Amazon EC2 instances should be managed by AWS Systems Manager
- [SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.4] SSM documents should not be public
- [SSM.5] SSM documents should be tagged
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [StepFunctions.2] Step Functions activities should be tagged
- [Transfer.1] AWS Transfer Family workflows should be tagged
- [Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [Transfer.5] Transfer Family certificates should be tagged
- [Transfer.6] Transfer Family connectors should be tagged
- [Transfer.7] Transfer Family profiles should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.2] AWS WAF Classic Regional rules should have at least one condition
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group
- [WAF.6] AWS WAF Classic global rules should have at least one condition

- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WAF.11] AWS WAF web ACL logging should be enabled
- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Asia Pacific (Thailand)

The following controls are not supported in the Asia Pacific (Thailand) Region.

- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [Account.1] Security contact information should be provided for an AWS account
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.4] AWS AppSync GraphQL APIs should be tagged
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys

- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Athena.2] Athena data catalogs should be tagged
- [Athena.3] Athena workgroups should be tagged
- [Athena.4] Athena workgroups should have logging enabled
- [AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones
- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones
- [AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.2] AWS Backup recovery points should be tagged
- [Backup.3] AWS Backup vaults should be tagged
- [Backup.4] AWS Backup report plans should be tagged
- [Backup.5] AWS Backup backup plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.2] Batch scheduling policies should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFormation.2] CloudFormation stacks should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins

- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket
- [CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys
- [CloudWatch.17] CloudWatch alarm actions should be activated
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [CodeBuild.3] CodeBuild S3 logs should be encrypted
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeBuild.7] CodeBuild report group exports should be encrypted at rest
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DataFirehose.1] Firehose delivery streams should be encrypted at rest
- [DataSync.1] DataSync tasks should have logging enabled
- [DataSync.2] DataSync tasks should be tagged
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.1] Database Migration Service replication instances should not be public
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged

- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [DynamoDB.6] DynamoDB tables should have deletion protection enabled
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.33] EC2 transit gateway attachments should be tagged
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged

- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.52] EC2 transit gateways should be tagged
- [EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports
- [EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports
- [EC2.55] VPCs should be configured with an interface endpoint for ECR API
- [EC2.56] VPCs should be configured with an interface endpoint for Docker Registry
- [EC2.57] VPCs should be configured with an interface endpoint for Systems Manager
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.171] EC2 VPN connections should have logging enabled
- [EC2.172] EC2 VPC Block Public Access settings should block internet gateway traffic
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.174] EC2 DHCP option sets should be tagged
- [EC2.175] EC2 launch templates should be tagged
- [EC2.176] EC2 prefix lists should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.178] EC2 traffic mirror filters should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECR.4] ECR public repositories should be tagged
- [ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys
- [ECS.3] ECS task definitions should not share the host's process namespace

- [ECS.4] ECS containers should run as non-privileged
- [ECS.5] ECS containers should be limited to read-only access to root filesystems
- [ECS.8] Secrets should not be passed as container environment variables
- [ECS.9] ECS task definitions should have a logging configuration
- [ECS.10] ECS Fargate services should run on the latest Fargate platform version
- [ECS.12] ECS clusters should use Container Insights
- [ECS.16] ECS task sets should not automatically assign public IP addresses
- [ECS.17] ECS task definitions should not use host network mode
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EFS.5] EFS access points should be tagged
- [EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch
- [EFS.7] EFS file systems should have automatic backups enabled
- [EFS.8] EFS file systems should be encrypted at rest
- [EKS.2] EKS clusters should run on a supported Kubernetes version
- [EKS.3] EKS clusters should use encrypted Kubernetes secrets
- [EKS.6] EKS clusters should be tagged
- [EKS.7] EKS identity provider configurations should be tagged
- [EKS.8] EKS clusters should have audit logging enabled
- [ELB.10] Classic Load Balancer should span multiple Availability Zones
- [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies

- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [EMR.2] Amazon EMR block public access setting should be enabled
- [EMR.3] Amazon EMR security configurations should be encrypted at rest
- [EMR.4] Amazon EMR security configurations should be encrypted in transit
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [ES.9] Elasticsearch domains should be tagged
- [EventBridge.2] EventBridge event buses should be tagged
- [EventBridge.3] EventBridge custom event buses should have a resource-based policy attached
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment
- [FSx.4] FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment

- [FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.1] AWS Glue jobs should be tagged
- [Glue.3] AWS Glue machine learning transforms should be encrypted at rest
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.1] GuardDuty should be enabled
- [GuardDuty.2] GuardDuty filters should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [GuardDuty.4] GuardDuty detectors should be tagged
- [GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled
- [GuardDuty.6] GuardDuty Lambda Protection should be enabled
- [GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled
- [GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled
- [GuardDuty.9] GuardDuty RDS Protection should be enabled
- [GuardDuty.10] GuardDuty S3 Protection should be enabled
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled
- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.9] MFA should be enabled for the root user
- [IAM.10] Password policies for IAM users should have strong configurations
- [IAM.11] Ensure IAM password policy requires at least one uppercase letter
- [IAM.12] Ensure IAM password policy requires at least one lowercase letter

- [IAM.13] Ensure IAM password policy requires at least one symbol
- [IAM.14] Ensure IAM password policy requires at least one number
- [IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
- [IAM.16] Ensure IAM password policy prevents password reuse
- [IAM.17] Ensure IAM password policy expires passwords within 90 days or less
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.23] IAM Access Analyzer analyzers should be tagged
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged

- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [Kinesis.2] Kinesis streams should be tagged
- [Kinesis.3] Kinesis streams should have an adequate data retention period
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [KMS.5] KMS keys should not be publicly accessible
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.4] Amazon MQ brokers should be tagged
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode

- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones
- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets
- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled

- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [PCA.2] AWS Private CA certificate authorities should be tagged
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.18] RDS instances should be deployed in a VPC
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.27] RDS DB clusters should be encrypted at rest
- [RDS.31] RDS DB security groups should be tagged
- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.36] RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [RDS.38] RDS for PostgreSQL DB instances should be encrypted in transit
- [RDS.39] RDS for MySQL DB instances should be encrypted in transit
- [RDS.40] RDS for SQL Server DB instances should publish logs to CloudWatch Logs
- [RDS.41] RDS for SQL Server DB instances should be encrypted in transit
- [RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs
- [RDS.44] RDS for MariaDB DB instances should be encrypted in transit
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.1] Amazon Redshift clusters should prohibit public access
- [Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled

- [Redshift.7] Redshift clusters should use enhanced VPC routing
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.11] Redshift clusters should be tagged
- [Redshift.12] Redshift event notification subscriptions should be tagged
- [Redshift.13] Redshift cluster snapshots should be tagged
- [Redshift.14] Redshift cluster subnet groups should be tagged
- [Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins
- [Redshift.16] Redshift cluster subnet groups should have subnets from multiple Availability Zones
- [Redshift.17] Redshift cluster parameter groups should be tagged
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.7] S3 general purpose buckets should use cross-Region replication
- [S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.12] ACLs should not be used to manage user access to S3 general purpose buckets
- [S3.13] S3 general purpose buckets should have Lifecycle configurations

- [S3.19] S3 access points should have block public access settings enabled
- [S3.20] S3 general purpose buckets should have MFA delete enabled
- [S3.22] S3 general purpose buckets should log object-level write events
- [S3.23] S3 general purpose buckets should log object-level read events
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SageMaker.8] SageMaker notebook instances should run on supported platforms
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SNS.4] SNS topic access policies should not allow public access
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.4] SSM documents should not be public
- [SSM.5] SSM documents should be tagged
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [StepFunctions.2] Step Functions activities should be tagged
- [Transfer.1] AWS Transfer Family workflows should be tagged

- [Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [Transfer.5] Transfer Family certificates should be tagged
- [Transfer.6] Transfer Family connectors should be tagged
- [Transfer.7] Transfer Family profiles should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.2] AWS WAF Classic Regional rules should have at least one condition
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WAF.11] AWS WAF web ACL logging should be enabled
- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Asia Pacific (Tokyo)

The following controls are not supported in the Asia Pacific (Tokyo) Region.

- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests

- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Canada (Central)

The following controls are not supported in the Canada (Central) Region.

- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest

- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled

- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Kinesis.3] Kinesis streams should have an adequate data retention period
- [RDS.31] RDS DB security groups should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Canada West (Calgary)

The following controls are not supported in the Canada West (Calgary) Region.

- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [Account.1] Security contact information should be provided for an AWS account
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged

- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.4] AWS AppSync GraphQL APIs should be tagged
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Athena.4] Athena workgroups should have logging enabled
- [AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones
- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones
- [AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled

- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket
- [CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys
- [CloudWatch.17] CloudWatch alarm actions should be activated
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [CodeBuild.3] CodeBuild S3 logs should be encrypted
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeBuild.7] CodeBuild report group exports should be encrypted at rest
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DataFirehose.1] Firehose delivery streams should be encrypted at rest
- [DataSync.1] DataSync tasks should have logging enabled
- [Detective.1] Detective behavior graphs should be tagged

- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [DynamoDB.6] DynamoDB tables should have deletion protection enabled
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.33] EC2 transit gateway attachments should be tagged

- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports
- [EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports
- [EC2.55] VPCs should be configured with an interface endpoint for ECR API
- [EC2.56] VPCs should be configured with an interface endpoint for Docker Registry
- [EC2.57] VPCs should be configured with an interface endpoint for Systems Manager
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.171] EC2 VPN connections should have logging enabled
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.175] EC2 launch templates should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECR.4] ECR public repositories should be tagged
- [ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys
- [ECS.3] ECS task definitions should not share the host's process namespace
- [ECS.4] ECS containers should run as non-privileged
- [ECS.5] ECS containers should be limited to read-only access to root filesystems

- [ECS.8] Secrets should not be passed as container environment variables
- [ECS.9] ECS task definitions should have a logging configuration
- [ECS.10] ECS Fargate services should run on the latest Fargate platform version
- [ECS.12] ECS clusters should use Container Insights
- [ECS.16] ECS task sets should not automatically assign public IP addresses
- [ECS.17] ECS task definitions should not use host network mode
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch
- [EFS.7] EFS file systems should have automatic backups enabled
- [EFS.8] EFS file systems should be encrypted at rest
- [EKS.2] EKS clusters should run on a supported Kubernetes version
- [EKS.3] EKS clusters should use encrypted Kubernetes secrets
- [EKS.6] EKS clusters should be tagged
- [EKS.7] EKS identity provider configurations should be tagged
- [EKS.8] EKS clusters should have audit logging enabled
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.10] Classic Load Balancer should span multiple Availability Zones
- [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit

- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [EMR.2] Amazon EMR block public access setting should be enabled
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.3] EventBridge custom event buses should have a resource-based policy attached
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment
- [FSx.4] FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment
- [FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.3] AWS Glue machine learning transforms should be encrypted at rest
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.2] GuardDuty filters should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged

- [GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled
- [GuardDuty.6] GuardDuty Lambda Protection should be enabled
- [GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled
- [GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled
- [GuardDuty.9] GuardDuty RDS Protection should be enabled
- [GuardDuty.10] GuardDuty S3 Protection should be enabled
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled
- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.9] MFA should be enabled for the root user
- [IAM.10] Password policies for IAM users should have strong configurations
- [IAM.11] Ensure IAM password policy requires at least one uppercase letter
- [IAM.12] Ensure IAM password policy requires at least one lowercase letter
- [IAM.13] Ensure IAM password policy requires at least one symbol
- [IAM.14] Ensure IAM password policy requires at least one number
- [IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
- [IAM.16] Ensure IAM password policy prevents password reuse
- [IAM.17] Ensure IAM password policy expires passwords within 90 days or less
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed

- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged

- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [Kinesis.2] Kinesis streams should be tagged
- [Kinesis.3] Kinesis streams should have an adequate data retention period
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [KMS.5] KMS keys should not be publicly accessible
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.4] Amazon MQ brokers should be tagged
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled

- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones
- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets
- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.18] RDS instances should be deployed in a VPC
- [RDS.24] RDS Database clusters should use a custom administrator username

- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.27] RDS DB clusters should be encrypted at rest
- [RDS.31] RDS DB security groups should be tagged
- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.36] RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [RDS.38] RDS for PostgreSQL DB instances should be encrypted in transit
- [RDS.39] RDS for MySQL DB instances should be encrypted in transit
- [RDS.40] RDS for SQL Server DB instances should publish logs to CloudWatch Logs
- [RDS.41] RDS for SQL Server DB instances should be encrypted in transit
- [RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs
- [RDS.44] RDS for MariaDB DB instances should be encrypted in transit
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins
- [Redshift.16] Redshift cluster subnet groups should have subnets from multiple Availability Zones
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys

- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.7] S3 general purpose buckets should use cross-Region replication
- [S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.12] ACLs should not be used to manage user access to S3 general purpose buckets
- [S3.13] S3 general purpose buckets should have Lifecycle configurations
- [S3.19] S3 access points should have block public access settings enabled
- [S3.20] S3 general purpose buckets should have MFA delete enabled
- [S3.22] S3 general purpose buckets should log object-level write events
- [S3.23] S3 general purpose buckets should log object-level read events
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SageMaker.8] SageMaker notebook instances should run on supported platforms
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SNS.4] SNS topic access policies should not allow public access
- [SQS.1] Amazon SQS queues should be encrypted at rest

- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.4] SSM documents should not be public
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.2] AWS WAF Classic Regional rules should have at least one condition
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

China (Beijing)

The following controls are not supported in the China (Beijing) Region.

- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [Account.2] AWS accounts should be part of an AWS Organizations organization

- [APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [AutoScaling.10] EC2 Auto Scaling groups should be tagged
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.2] Batch scheduling policies should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins

- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DataFirehose.1] Firehose delivery streams should be encrypted at rest
- [DataSync.2] DataSync tasks should be tagged
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up

- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.33] EC2 transit gateway attachments should be tagged
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.35] EC2 network interfaces should be tagged
- [EC2.36] EC2 customer gateways should be tagged
- [EC2.42] EC2 route tables should be tagged
- [EC2.46] Amazon VPCs should be tagged
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports
- [EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.171] EC2 VPN connections should have logging enabled
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.174] EC2 DHCP option sets should be tagged
- [EC2.175] EC2 launch templates should be tagged
- [EC2.176] EC2 prefix lists should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.178] EC2 traffic mirror filters should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.4] ECR public repositories should be tagged

- [EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch
- [EKS.3] EKS clusters should use encrypted Kubernetes secrets
- [EKS.6] EKS clusters should be tagged
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.2] Amazon EMR block public access setting should be enabled
- [EMR.3] Amazon EMR security configurations should be encrypted at rest
- [EMR.4] Amazon EMR security configurations should be encrypted in transit
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [GuardDuty.4] GuardDuty detectors should be tagged
- [GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled
- [GuardDuty.6] GuardDuty Lambda Protection should be enabled
- [GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled
- [GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled
- [GuardDuty.9] GuardDuty RDS Protection should be enabled

- [GuardDuty.10] GuardDuty S3 Protection should be enabled
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled
- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.9] MFA should be enabled for the root user
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.23] IAM Access Analyzer analyzers should be tagged
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged

- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones
- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets
- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.7] Network Firewall firewalls should be tagged

- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [PCA.2] AWS Private CA certificate authorities should be tagged
- [RDS.7] RDS clusters should have deletion protection enabled
- [RDS.12] IAM authentication should be configured for RDS clusters
- [RDS.13] RDS automatic minor version upgrades should be enabled
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.16] Aurora DB clusters should be configured to copy tags to DB snapshots
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.27] RDS DB clusters should be encrypted at rest
- [RDS.28] RDS DB clusters should be tagged
- [RDS.31] RDS DB security groups should be tagged
- [RDS.32] RDS DB snapshots should be tagged
- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs

- [RDS.44] RDS for MariaDB DB instances should be encrypted in transit
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins
- [Redshift.17] Redshift cluster parameter groups should be tagged
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.22] S3 general purpose buckets should log object-level write events
- [S3.23] S3 general purpose buckets should log object-level read events
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SSM.5] SSM documents should be tagged
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection
- [Transfer.4] Transfer Family agreements should be tagged
- [Transfer.5] Transfer Family certificates should be tagged
- [Transfer.6] Transfer Family connectors should be tagged

- [Transfer.7] Transfer Family profiles should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

China (Ningxia)

The following controls are not supported in the China (Ningxia) Region.

- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [AutoScaling.10] EC2 Auto Scaling groups should be tagged
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged

- [Batch.1] Batch job queues should be tagged
- [Batch.2] Batch scheduling policies should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DataFirehose.1] Firehose delivery streams should be encrypted at rest
- [DataSync.2] DataSync tasks should be tagged

- [Detective.1] Detective behavior graphs should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.33] EC2 transit gateway attachments should be tagged
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.35] EC2 network interfaces should be tagged
- [EC2.36] EC2 customer gateways should be tagged
- [EC2.42] EC2 route tables should be tagged
- [EC2.46] Amazon VPCs should be tagged
- [EC2.50] EC2 VPN gateways should be tagged
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.171] EC2 VPN connections should have logging enabled
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes

- [EC2.174] EC2 DHCP option sets should be tagged
- [EC2.175] EC2 launch templates should be tagged
- [EC2.176] EC2 prefix lists should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.178] EC2 traffic mirror filters should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.4] ECR public repositories should be tagged
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch
- [EKS.3] EKS clusters should use encrypted Kubernetes secrets
- [EKS.6] EKS clusters should be tagged
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.2] Amazon EMR block public access setting should be enabled
- [EMR.3] Amazon EMR security configurations should be encrypted at rest
- [EMR.4] Amazon EMR security configurations should be encrypted in transit
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes

- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.3] AWS Glue machine learning transforms should be encrypted at rest
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [GuardDuty.4] GuardDuty detectors should be tagged
- [GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled
- [GuardDuty.6] GuardDuty Lambda Protection should be enabled
- [GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled
- [GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled
- [GuardDuty.9] GuardDuty RDS Protection should be enabled
- [GuardDuty.10] GuardDuty S3 Protection should be enabled
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled
- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.9] MFA should be enabled for the root user
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.23] IAM Access Analyzer analyzers should be tagged
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged

- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Lambda.1] Lambda function policies should prohibit public access
- [Lambda.2] Lambda functions should use supported runtimes
- [Lambda.3] Lambda functions should be in a VPC
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Lambda.6] Lambda functions should be tagged
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones

- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets
- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.7] Network Firewall firewalls should be tagged
- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [PCA.2] AWS Private CA certificate authorities should be tagged
- [RDS.13] RDS automatic minor version upgrades should be enabled
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.28] RDS DB clusters should be tagged
- [RDS.31] RDS DB security groups should be tagged
- [RDS.32] RDS DB snapshots should be tagged

- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs
- [RDS.44] RDS for MariaDB DB instances should be encrypted in transit
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins
- [Redshift.17] Redshift cluster parameter groups should be tagged
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SSM.5] SSM documents should be tagged
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [StepFunctions.2] Step Functions activities should be tagged
- [Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection
- [Transfer.4] Transfer Family agreements should be tagged
- [Transfer.5] Transfer Family certificates should be tagged
- [Transfer.6] Transfer Family connectors should be tagged

- [Transfer.7] Transfer Family profiles should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Europe (Frankfurt)

The following controls are not supported in the Europe (Frankfurt) Region.

- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged

- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [RDS.31] RDS DB security groups should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Europe (Ireland)

The following controls are not supported in the Europe (Ireland) Region.

- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy

- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [ECR.4] ECR public repositories should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Europe (London)

The following controls are not supported in the Europe (London) Region.

- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control

- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [RDS.31] RDS DB security groups should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled

- [S3.25] S3 directory buckets should have lifecycle configurations
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

Europe (Milan)

The following controls are not supported in the Europe (Milan) Region.

- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged

- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [ECR.4] ECR public repositories should be tagged
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged

- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots

- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [RDS.1] RDS snapshot should be private
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Europe (Paris)

The following controls are not supported in the Europe (Paris) Region.

- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed

- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [RDS.31] RDS DB security groups should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Europe (Spain)

The following controls are not supported in the Europe (Spain) Region.

- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged

- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.1] Amazon EBS snapshots should not be publicly restorable

- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.175] EC2 launch templates should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.4] ECR public repositories should be tagged
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled

- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.2] GuardDuty filters should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged

- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged

- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [Lambda.1] Lambda function policies should prohibit public access
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.4] Amazon MQ brokers should be tagged
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled

- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [RDS.1] RDS snapshot should be private
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT

- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Europe (Stockholm)

The following controls are not supported in the Europe (Stockholm) Region.

- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins

- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged

- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Europe (Zurich)

The following controls are not supported in the Europe (Zurich) Region.

- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged

- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication

- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts

- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.175] EC2 launch templates should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.4] ECR public repositories should be tagged
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged

- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.2] GuardDuty filters should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged

- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.4] Amazon MQ brokers should be tagged
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Neptune.1] Neptune DB clusters should be encrypted at rest

- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [RDS.1] RDS snapshot should be private
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC

- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Israel (Tel Aviv)

The following controls are not supported in the Israel (Tel Aviv) Region.

- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged

- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities

- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces

- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.33] EC2 transit gateway attachments should be tagged
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.55] VPCs should be configured with an interface endpoint for ECR API
- [EC2.56] VPCs should be configured with an interface endpoint for Docker Registry
- [EC2.57] VPCs should be configured with an interface endpoint for Systems Manager
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.175] EC2 launch templates should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECR.4] ECR public repositories should be tagged
- [ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys
- [ECS.16] ECS task sets should not automatically assign public IP addresses
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EFS.8] EFS file systems should be encrypted at rest
- [EKS.2] EKS clusters should run on a supported Kubernetes version

- [EKS.6] EKS clusters should be tagged
- [EKS.7] EKS identity provider configurations should be tagged
- [EKS.8] EKS clusters should have audit logging enabled
- [ELB.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged

- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.2] GuardDuty filters should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.9] MFA should be enabled for the root user
- [IAM.10] Password policies for IAM users should have strong configurations
- [IAM.11] Ensure IAM password policy requires at least one uppercase letter
- [IAM.12] Ensure IAM password policy requires at least one lowercase letter
- [IAM.13] Ensure IAM password policy requires at least one symbol
- [IAM.14] Ensure IAM password policy requires at least one number
- [IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
- [IAM.16] Ensure IAM password policy prevents password reuse
- [IAM.17] Ensure IAM password policy expires passwords within 90 days or less
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled

- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [Kinesis.2] Kinesis streams should be tagged

- [Kinesis.3] Kinesis streams should have an adequate data retention period
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.4] Amazon MQ brokers should be tagged
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged

- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [RDS.1] RDS snapshot should be private
- [RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.29] RDS DB cluster snapshots should be tagged
- [RDS.31] RDS DB security groups should be tagged
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.5] SageMaker models should have network isolation enabled
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT

- [SSM.4] SSM documents should not be public
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Mexico (Central)

The following controls are not supported in the Mexico (Central) Region.

- [ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [Account.1] Security contact information should be provided for an AWS account
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.1] API Gateway REST and WebSocket API execution logging should be enabled
- [APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication
- [APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled
- [APIGateway.4] API Gateway should be associated with a WAF Web ACL
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged

- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.4] AWS AppSync GraphQL APIs should be tagged
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [Athena.2] Athena data catalogs should be tagged
- [Athena.3] Athena workgroups should be tagged
- [Athena.4] Athena workgroups should have logging enabled
- [AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones
- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones
- [AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.2] AWS Backup recovery points should be tagged
- [Backup.3] AWS Backup vaults should be tagged
- [Backup.4] AWS Backup report plans should be tagged
- [Backup.5] AWS Backup backup plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.2] Batch scheduling policies should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFormation.2] CloudFormation stacks should be tagged

- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [CloudTrail.7] Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket
- [CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys
- [CloudWatch.17] CloudWatch alarm actions should be activated
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [CodeBuild.3] CodeBuild S3 logs should be encrypted
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeBuild.7] CodeBuild report group exports should be encrypted at rest
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities

- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DataFirehose.1] Firehose delivery streams should be encrypted at rest
- [DataSync.1] DataSync tasks should have logging enabled
- [DataSync.2] DataSync tasks should be tagged
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.1] Database Migration Service replication instances should not be public
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [DynamoDB.6] DynamoDB tables should have deletion protection enabled
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period

- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.33] EC2 transit gateway attachments should be tagged
- [EC2.34] EC2 transit gateway route tables should be tagged
- [EC2.40] EC2 NAT gateways should be tagged
- [EC2.48] Amazon VPC flow logs should be tagged
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.52] EC2 transit gateways should be tagged
- [EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports
- [EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports
- [EC2.55] VPCs should be configured with an interface endpoint for ECR API
- [EC2.56] VPCs should be configured with an interface endpoint for Docker Registry
- [EC2.57] VPCs should be configured with an interface endpoint for Systems Manager
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.171] EC2 VPN connections should have logging enabled
- [EC2.172] EC2 VPC Block Public Access settings should block internet gateway traffic
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.174] EC2 DHCP option sets should be tagged
- [EC2.175] EC2 launch templates should be tagged

- [EC2.176] EC2 prefix lists should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.178] EC2 traffic mirror filters should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECR.4] ECR public repositories should be tagged
- [ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys
- [ECS.3] ECS task definitions should not share the host's process namespace
- [ECS.4] ECS containers should run as non-privileged
- [ECS.5] ECS containers should be limited to read-only access to root filesystems
- [ECS.8] Secrets should not be passed as container environment variables
- [ECS.9] ECS task definitions should have a logging configuration
- [ECS.10] ECS Fargate services should run on the latest Fargate platform version
- [ECS.12] ECS clusters should use Container Insights
- [ECS.16] ECS task sets should not automatically assign public IP addresses
- [ECS.17] ECS task definitions should not use host network mode
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EFS.5] EFS access points should be tagged
- [EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch
- [EFS.7] EFS file systems should have automatic backups enabled
- [EFS.8] EFS file systems should be encrypted at rest
- [EKS.2] EKS clusters should run on a supported Kubernetes version
- [EKS.3] EKS clusters should use encrypted Kubernetes secrets
- [EKS.6] EKS clusters should be tagged

- [EKS.7] EKS identity provider configurations should be tagged
- [EKS.8] EKS clusters should have audit logging enabled
- [ELB.10] Classic Load Balancer should span multiple Availability Zones
- [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [EMR.2] Amazon EMR block public access setting should be enabled
- [EMR.3] Amazon EMR security configurations should be encrypted at rest
- [EMR.4] Amazon EMR security configurations should be encrypted in transit
- [ES.3] Elasticsearch domains should encrypt data sent between nodes
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [ES.9] Elasticsearch domains should be tagged

- [EventBridge.2] EventBridge event buses should be tagged
- [EventBridge.3] EventBridge custom event buses should have a resource-based policy attached
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment
- [FSx.4] FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment
- [FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.1] AWS Glue jobs should be tagged
- [Glue.3] AWS Glue machine learning transforms should be encrypted at rest
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.1] GuardDuty should be enabled
- [GuardDuty.2] GuardDuty filters should be tagged
- [GuardDuty.3] GuardDuty IPSets should be tagged
- [GuardDuty.4] GuardDuty detectors should be tagged
- [GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled
- [GuardDuty.6] GuardDuty Lambda Protection should be enabled
- [GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled
- [GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled
- [GuardDuty.9] GuardDuty RDS Protection should be enabled
- [GuardDuty.10] GuardDuty S3 Protection should be enabled
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled
- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.1] IAM policies should not allow full "*" administrative privileges

- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.7] Password policies for IAM users should have strong configurations
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.9] MFA should be enabled for the root user
- [IAM.10] Password policies for IAM users should have strong configurations
- [IAM.11] Ensure IAM password policy requires at least one uppercase letter
- [IAM.12] Ensure IAM password policy requires at least one lowercase letter
- [IAM.13] Ensure IAM password policy requires at least one symbol
- [IAM.14] Ensure IAM password policy requires at least one number
- [IAM.15] Ensure IAM password policy requires minimum password length of 14 or greater
- [IAM.16] Ensure IAM password policy prevents password reuse
- [IAM.17] Ensure IAM password policy expires passwords within 90 days or less
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.23] IAM Access Analyzer analyzers should be tagged
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled

- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoT.4] AWS IoT Core authorizers should be tagged
- [IoT.5] AWS IoT Core role aliases should be tagged
- [IoT.6] AWS IoT Core policies should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [Kinesis.2] Kinesis streams should be tagged
- [Kinesis.3] Kinesis streams should have an adequate data retention period
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys

- [KMS.5] KMS keys should not be publicly accessible
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.4] Amazon MQ brokers should be tagged
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones
- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets

- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [PCA.2] AWS Private CA certificate authorities should be tagged
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.18] RDS instances should be deployed in a VPC
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.27] RDS DB clusters should be encrypted at rest
- [RDS.31] RDS DB security groups should be tagged
- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.36] RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs
- [RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs
- [RDS.38] RDS for PostgreSQL DB instances should be encrypted in transit

- [RDS.39] RDS for MySQL DB instances should be encrypted in transit
- [RDS.40] RDS for SQL Server DB instances should publish logs to CloudWatch Logs
- [RDS.41] RDS for SQL Server DB instances should be encrypted in transit
- [RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs
- [RDS.44] RDS for MariaDB DB instances should be encrypted in transit
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.1] Amazon Redshift clusters should prohibit public access
- [Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Redshift.4] Amazon Redshift clusters should have audit logging enabled
- [Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled
- [Redshift.7] Redshift clusters should use enhanced VPC routing
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.11] Redshift clusters should be tagged
- [Redshift.12] Redshift event notification subscriptions should be tagged
- [Redshift.13] Redshift cluster snapshots should be tagged
- [Redshift.14] Redshift cluster subnet groups should be tagged
- [Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins
- [Redshift.16] Redshift cluster subnet groups should have subnets from multiple Availability Zones
- [Redshift.17] Redshift cluster parameter groups should be tagged
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys

- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.7] S3 general purpose buckets should use cross-Region replication
- [S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.12] ACLs should not be used to manage user access to S3 general purpose buckets
- [S3.13] S3 general purpose buckets should have Lifecycle configurations
- [S3.19] S3 access points should have block public access settings enabled
- [S3.20] S3 general purpose buckets should have MFA delete enabled
- [S3.22] S3 general purpose buckets should log object-level write events
- [S3.23] S3 general purpose buckets should log object-level read events
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SageMaker.8] SageMaker notebook instances should run on supported platforms
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [SecretsManager.1] Secrets Manager secrets should have automatic rotation enabled
- [SecretsManager.2] Secrets Manager secrets configured with automatic rotation should rotate successfully

- [SecretsManager.3] Remove unused Secrets Manager secrets
- [SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days
- [ServiceCatalog.1] Service Catalog portfolios should be shared within an AWS organization only
- [SNS.4] SNS topic access policies should not allow public access
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.3] Amazon EC2 instances managed by Systems Manager should have an association compliance status of COMPLIANT
- [SSM.4] SSM documents should not be public
- [SSM.5] SSM documents should be tagged
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [StepFunctions.2] Step Functions activities should be tagged
- [Transfer.1] AWS Transfer Family workflows should be tagged
- [Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection
- [Transfer.3] Transfer Family connectors should have logging enabled
- [Transfer.4] Transfer Family agreements should be tagged
- [Transfer.5] Transfer Family certificates should be tagged
- [Transfer.6] Transfer Family connectors should be tagged
- [Transfer.7] Transfer Family profiles should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.2] AWS WAF Classic Regional rules should have at least one condition
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WAF.11] AWS WAF web ACL logging should be enabled

- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Middle East (Bahrain)

The following controls are not supported in the Middle East (Bahrain) Region.

- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.10] CloudTrail Lake event data stores should be encrypted with customer managed AWS KMS keys
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled

- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DocumentDB.6] Amazon DocumentDB clusters should be encrypted in transit
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys
- [ECS.17] ECS task definitions should not use host network mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment
- [FSx.4] FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment
- [FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged

- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled
- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [RDS.41] RDS for SQL Server DB instances should be encrypted in transit

- [RDS.42] RDS for MariaDB DB instances should publish logs to CloudWatch Logs
- [RDS.44] RDS for MariaDB DB instances should be encrypted in transit
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.4] Redshift Serverless namespaces should be encrypted with customer managed AWS KMS keys
- [RedshiftServerless.5] Redshift Serverless namespaces should not use the default admin username
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [RedshiftServerless.7] Redshift Serverless namespaces should not use the default database name
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.8] SageMaker notebook instances should run on supported platforms
- [SQS.3] SQS queue access policies should not allow public access
- [Transfer.3] Transfer Family connectors should have logging enabled
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

Middle East (UAE)

The following controls are not supported in the Middle East (UAE) Region.

- [APIGateway.8] API Gateway routes should specify an authorization type

- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [AutoScaling.1] Auto Scaling groups associated with a load balancer should use ELB health checks
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [Backup.4] AWS Backup report plans should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [CloudWatch.16] CloudWatch log groups should be retained for a specified time period

- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [Detective.1] Detective behavior graphs should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.3] DMS event subscriptions should be tagged
- [DMS.4] DMS replication instances should be tagged
- [DMS.5] DMS replication subnet groups should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled
- [DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled
- [DMS.12] DMS endpoints for Redis OSS should have TLS enabled
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.12] Unused Amazon EC2 EIPs should be removed
- [EC2.14] Security groups should not allow ingress from 0.0.0.0/0 or ::/0 to port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts

- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.175] EC2 launch templates should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [EC2.180] EC2 network interfaces should have source/destination checking enabled
- [ECR.4] ECR public repositories should be tagged
- [ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions
- [EFS.1] Elastic File System should be configured to encrypt file data at-rest using AWS KMS
- [EFS.2] Amazon EFS volumes should be in backup plans
- [ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.17] Application and Network Load Balancers with listeners should use recommended security policies
- [ELB.18] Application and Network Load Balancer listeners should use secure protocols to encrypt data in transit
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch

- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue
- [GuardDuty.2] GuardDuty filters should be tagged
- [IAM.1] IAM policies should not allow full "*" administrative privileges
- [IAM.2] IAM users should not have IAM policies attached
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.4] IAM root user access key should not exist
- [IAM.5] MFA should be enabled for all IAM users that have a console password
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.8] Unused IAM user credentials should be removed
- [IAM.9] MFA should be enabled for the root user
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [IAM.19] MFA should be enabled for all IAM users
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.22] IAM user credentials unused for 45 days should be removed
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached
- [Inspector.1] Amazon Inspector EC2 scanning should be enabled
- [Inspector.2] Amazon Inspector ECR scanning should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [Inspector.4] Amazon Inspector Lambda standard scanning should be enabled
- [IoTEvents.1] AWS IoT Events inputs should be tagged

- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [KMS.1] IAM customer managed policies should not allow decryption actions on all KMS keys
- [KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys
- [KMS.4] AWS KMS key rotation should be enabled
- [Lambda.7] Lambda functions should have AWS X-Ray active tracing enabled
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.4] MSK clusters should have public access disabled
- [MSK.5] MSK connectors should have logging enabled
- [MSK.6] MSK clusters should disable unauthenticated access
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible

- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [Opensearch.9] OpenSearch domains should be tagged
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.31] RDS DB security groups should be tagged
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [Redshift.18] Redshift clusters should have Multi-AZ deployments enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SQS.2] SQS queues should be tagged
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.1] Amazon EC2 instances should be managed by AWS Systems Manager

- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [SSM.7] SSM documents should have the block public sharing setting enabled
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

South America (São Paulo)

The following controls are not supported in the South America (São Paulo) Region.

- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CodeArtifact.1]CodeArtifact repositories should be tagged

- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [ECR.4] ECR public repositories should be tagged
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoT.1] AWS IoT Device Defender security profiles should be tagged
- [IoT.2] AWS IoT Core mitigation actions should be tagged
- [IoT.3] AWS IoT Core dimensions should be tagged
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged

- [IVS.3] IVS channels should be tagged
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group

AWS GovCloud (US-East)

The following controls are not supported in the AWS GovCloud (US-East) Region.

- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [Account.1] Security contact information should be provided for an AWS account
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged
- [AppFlow.1] Amazon AppFlow flows should be tagged

- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.4] AWS AppSync GraphQL APIs should be tagged
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones
- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones
- [AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates
- [Backup.4] AWS Backup report plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.2] Batch scheduling policies should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins
- [CloudFront.13] CloudFront distributions should use origin access control

- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudWatch.17] CloudWatch alarm actions should be activated
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeBuild.3] CodeBuild S3 logs should be encrypted
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Cognito.2] Cognito identity pools should not allow unauthenticated identities
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [Connect.2] Amazon Connect instances should have CloudWatch logging enabled
- [DataSync.2] DataSync tasks should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests

- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.47] Amazon VPC endpoint services should be tagged
- [EC2.52] EC2 transit gateways should be tagged
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.174] EC2 DHCP option sets should be tagged
- [EC2.175] EC2 launch templates should be tagged
- [EC2.176] EC2 prefix lists should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.178] EC2 traffic mirror filters should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECR.4] ECR public repositories should be tagged
- [ECS.3] ECS task definitions should not share the host's process namespace
- [ECS.4] ECS containers should run as non-privileged
- [ECS.5] ECS containers should be limited to read-only access to root filesystems
- [ECS.8] Secrets should not be passed as container environment variables
- [ECS.9] ECS task definitions should have a logging configuration
- [ECS.10] ECS Fargate services should run on the latest Fargate platform version
- [ECS.12] ECS clusters should use Container Insights
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EKS.2] EKS clusters should run on a supported Kubernetes version

- [EKS.8] EKS clusters should have audit logging enabled
- [ELB.10] Classic Load Balancer should span multiple Availability Zones
- [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.2] Amazon EMR block public access setting should be enabled
- [EMR.3] Amazon EMR security configurations should be encrypted at rest
- [EMR.4] Amazon EMR security configurations should be encrypted in transit
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.3] EventBridge custom event buses should have a resource-based policy attached
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged

- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [Glue.3] AWS Glue machine learning transforms should be encrypted at rest
- [GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled
- [GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled
- [GuardDuty.9] GuardDuty RDS Protection should be enabled
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled
- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.9] MFA should be enabled for the root user
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.26] Expired SSL/TLS certificates managed in IAM should be removed
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged

- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged
- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [KMS.5] KMS keys should not be publicly accessible
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones

- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets
- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [PCA.2] AWS Private CA certificate authorities should be tagged
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.27] RDS DB clusters should be encrypted at rest
- [RDS.31] RDS DB security groups should be tagged
- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.10] Redshift clusters should be encrypted at rest

- [Redshift.17] Redshift cluster parameter groups should be tagged
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing
- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.12] ACLs should not be used to manage user access to S3 general purpose buckets
- [S3.13] S3 general purpose buckets should have Lifecycle configurations
- [S3.20] S3 general purpose buckets should have MFA delete enabled
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SES.1] SES contact lists should be tagged
- [SES.2] SES configuration sets should be tagged
- [SNS.4] SNS topic access policies should not allow public access
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.4] SSM documents should not be public
- [SSM.5] SSM documents should be tagged
- [SSM.6] SSM Automation should have CloudWatch logging enabled
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [StepFunctions.2] Step Functions activities should be tagged
- [Transfer.4] Transfer Family agreements should be tagged

- [Transfer.5] Transfer Family certificates should be tagged
- [Transfer.6] Transfer Family connectors should be tagged
- [Transfer.7] Transfer Family profiles should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.2] AWS WAF Classic Regional rules should have at least one condition
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule
- [WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled
- [WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest
- [WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest

AWS GovCloud (US-West)

The following controls are not supported in the AWS GovCloud (US-West) Region.

- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [Account.1] Security contact information should be provided for an AWS account
- [Account.2] AWS accounts should be part of an AWS Organizations organization
- [APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication
- [APIGateway.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [Amplify.1] Amplify apps should be tagged
- [Amplify.2] Amplify branches should be tagged
- [AppConfig.1] AWS AppConfig applications should be tagged
- [AppConfig.2] AWS AppConfig configuration profiles should be tagged
- [AppConfig.3] AWS AppConfig environments should be tagged
- [AppConfig.4] AWS AppConfig extension associations should be tagged

- [AppFlow.1] Amazon AppFlow flows should be tagged
- [AppRunner.1] App Runner services should be tagged
- [AppRunner.2] App Runner VPC connectors should be tagged
- [AppSync.1] AWS AppSync API caches should be encrypted at rest
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.4] AWS AppSync GraphQL APIs should be tagged
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [AppSync.6] AWS AppSync API caches should be encrypted in transit
- [AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones
- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones
- [AutoScaling.9] Amazon EC2 Auto Scaling groups should use Amazon EC2 launch templates
- [Backup.4] AWS Backup report plans should be tagged
- [Batch.1] Batch job queues should be tagged
- [Batch.2] Batch scheduling policies should be tagged
- [Batch.3] Batch compute environments should be tagged
- [Batch.4] Compute resources properties in managed Batch compute environments should be tagged
- [CloudFront.1] CloudFront distributions should have a default root object configured
- [CloudFront.3] CloudFront distributions should require encryption in transit
- [CloudFront.4] CloudFront distributions should have origin failover configured
- [CloudFront.5] CloudFront distributions should have logging enabled
- [CloudFront.6] CloudFront distributions should have WAF enabled
- [CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates
- [CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests
- [CloudFront.9] CloudFront distributions should encrypt traffic to custom origins
- [CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins
- [CloudFront.12] CloudFront distributions should not point to non-existent S3 origins

- [CloudFront.13] CloudFront distributions should use origin access control
- [CloudFront.14] CloudFront distributions should be tagged
- [CloudFront.15] CloudFront distributions should use the recommended TLS security policy
- [CloudWatch.17] CloudWatch alarm actions should be activated
- [CodeArtifact.1]CodeArtifact repositories should be tagged
- [CodeBuild.3] CodeBuild S3 logs should be encrypted
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeGuruProfiler.1] CodeGuru Profiler profiling groups should be tagged
- [CodeGuruReviewer.1] CodeGuru Reviewer repository associations should be tagged
- [Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication
- [Connect.1] Amazon Connect Customer Profiles object types should be tagged
- [DataSync.2] DataSync tasks should be tagged
- [DMS.2] DMS certificates should be tagged
- [DMS.6] DMS replication instances should have automatic minor version upgrade enabled
- [DMS.7] DMS replication tasks for the target database should have logging enabled
- [DMS.8] DMS replication tasks for the source database should have logging enabled
- [DMS.9] DMS endpoints should use SSL
- [DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest
- [DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period
- [DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public
- [DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs
- [DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit
- [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
- [EC2.22] Unused Amazon EC2 security groups should be removed
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used

- [EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.38] EC2 instances should be tagged
- [EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts
- [EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager
- [EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.173] EC2 Spot Fleet requests with launch parameters should enable encryption for attached EBS volumes
- [EC2.174] EC2 DHCP option sets should be tagged
- [EC2.175] EC2 launch templates should be tagged
- [EC2.176] EC2 prefix lists should be tagged
- [EC2.177] EC2 traffic mirror sessions should be tagged
- [EC2.178] EC2 traffic mirror filters should be tagged
- [EC2.179] EC2 traffic mirror targets should be tagged
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.2] ECR private repositories should have tag immutability configured
- [ECR.3] ECR repositories should have at least one lifecycle policy configured
- [ECR.4] ECR public repositories should be tagged
- [ECS.3] ECS task definitions should not share the host's process namespace
- [ECS.4] ECS containers should run as non-privileged
- [ECS.5] ECS containers should be limited to read-only access to root filesystems
- [ECS.8] Secrets should not be passed as container environment variables
- [ECS.9] ECS task definitions should have a logging configuration
- [ECS.10] ECS Fargate services should run on the latest Fargate platform version
- [ECS.12] ECS clusters should use Container Insights
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [EKS.2] EKS clusters should run on a supported Kubernetes version
- [EKS.8] EKS clusters should have audit logging enabled

- [ELB.10] Classic Load Balancer should span multiple Availability Zones
- [ELB.12] Application Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.13] Application, Network and Gateway Load Balancers should span multiple Availability Zones
- [ELB.14] Classic Load Balancer should be configured with defensive or strictest desync mitigation mode
- [ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL
- [ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled
- [ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled
- [ElastiCache.3] ElastiCache replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group
- [ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled
- [ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [EMR.2] Amazon EMR block public access setting should be enabled
- [EMR.3] Amazon EMR security configurations should be encrypted at rest
- [EMR.4] Amazon EMR security configurations should be encrypted in transit
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [EventBridge.3] EventBridge custom event buses should have a resource-based policy attached
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [FraudDetector.1] Amazon Fraud Detector entity types should be tagged
- [FraudDetector.2] Amazon Fraud Detector labels should be tagged
- [FraudDetector.3] Amazon Fraud Detector outcomes should be tagged
- [FraudDetector.4] Amazon Fraud Detector variables should be tagged
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes

- [FSx.2] FSx for Lustre file systems should be configured to copy tags to backups
- [GlobalAccelerator.1] Global Accelerator accelerators should be tagged
- [GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled
- [GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled
- [GuardDuty.9] GuardDuty RDS Protection should be enabled
- [GuardDuty.11] GuardDuty Runtime Monitoring should be enabled
- [GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled
- [GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled
- [IAM.6] Hardware MFA should be enabled for the root user
- [IAM.9] MFA should be enabled for the root user
- [IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services
- [IAM.24] IAM roles should be tagged
- [IAM.25] IAM users should be tagged
- [IAM.28] IAM Access Analyzer external access analyzer should be enabled
- [Inspector.3] Amazon Inspector Lambda code scanning should be enabled
- [IoTEvents.1] AWS IoT Events inputs should be tagged
- [IoTEvents.2] AWS IoT Events detector models should be tagged
- [IoTEvents.3] AWS IoT Events alarm models should be tagged
- [IoTSiteWise.1] AWS IoT SiteWise asset models should be tagged
- [IoTSiteWise.2] AWS IoT SiteWise dashboards should be tagged
- [IoTSiteWise.3] AWS IoT SiteWise gateways should be tagged
- [IoTSiteWise.4] AWS IoT SiteWise portals should be tagged
- [IoTSiteWise.5] AWS IoT SiteWise projects should be tagged
- [IoTTwinMaker.1] AWS IoT TwinMaker sync jobs should be tagged
- [IoTTwinMaker.2] AWS IoT TwinMaker workspaces should be tagged
- [IoTTwinMaker.3] AWS IoT TwinMaker scenes should be tagged
- [IoTTwinMaker.4] AWS IoT TwinMaker entities should be tagged
- [IoTWireless.1] AWS IoT Wireless multicast groups should be tagged
- [IoTWireless.2] AWS IoT Wireless service profiles should be tagged
- [IoTWireless.3] AWS IoT FUOTA tasks should be tagged

- [IVS.1] IVS playback key pairs should be tagged
- [IVS.2] IVS recording configurations should be tagged
- [IVS.3] IVS channels should be tagged
- [Keyspaces.1] Amazon Keyspaces keyspaces should be tagged
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [KMS.5] KMS keys should not be publicly accessible
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be enabled
- [MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled
- [MQ.5] ActiveMQ brokers should use active/standby deployment mode
- [MQ.6] RabbitMQ brokers should use cluster deployment mode
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [MSK.3] MSK Connect connectors should be encrypted in transit
- [MSK.5] MSK connectors should have logging enabled
- [Neptune.1] Neptune DB clusters should be encrypted at rest
- [Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs
- [Neptune.3] Neptune DB cluster snapshots should not be public
- [Neptune.4] Neptune DB clusters should have deletion protection enabled
- [Neptune.5] Neptune DB clusters should have automated backups enabled
- [Neptune.6] Neptune DB cluster snapshots should be encrypted at rest
- [Neptune.7] Neptune DB clusters should have IAM database authentication enabled
- [Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots
- [Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones
- [NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones
- [NetworkFirewall.2] Network Firewall logging should be enabled
- [NetworkFirewall.3] Network Firewall policies should have at least one rule group associated
- [NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets

- [NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets
- [NetworkFirewall.6] Stateless Network Firewall rule group should not be empty
- [NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled
- [Opensearch.1] OpenSearch domains should have encryption at rest enabled
- [Opensearch.2] OpenSearch domains should not be publicly accessible
- [Opensearch.3] OpenSearch domains should encrypt data sent between nodes
- [Opensearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled
- [Opensearch.5] OpenSearch domains should have audit logging enabled
- [Opensearch.6] OpenSearch domains should have at least three data nodes
- [Opensearch.7] OpenSearch domains should have fine-grained access control enabled
- [Opensearch.8] Connections to OpenSearch domains should be encrypted using the latest TLS security policy
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [PCA.2] AWS Private CA certificate authorities should be tagged
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.25] RDS database instances should use a custom administrator username
- [RDS.27] RDS DB clusters should be encrypted at rest
- [RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [RDS.45] Aurora MySQL DB clusters should have audit logging enabled
- [Redshift.7] Redshift clusters should use enhanced VPC routing
- [Redshift.8] Amazon Redshift clusters should not use the default Admin username
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Redshift.11] Redshift clusters should be tagged
- [Redshift.13] Redshift cluster snapshots should be tagged
- [Redshift.17] Redshift cluster parameter groups should be tagged
- [RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing

- [RedshiftServerless.2] Connections to Redshift Serverless workgroups should be required to use SSL
- [RedshiftServerless.3] Redshift Serverless workgroups should prohibit public access
- [RedshiftServerless.6] Redshift Serverless namespaces should export logs to CloudWatch Logs
- [Route53.1] Route 53 health checks should be tagged
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.12] ACLs should not be used to manage user access to S3 general purpose buckets
- [S3.13] S3 general purpose buckets should have Lifecycle configurations
- [S3.20] S3 general purpose buckets should have MFA delete enabled
- [S3.24] S3 Multi-Region Access Points should have block public access settings enabled
- [S3.25] S3 directory buckets should have lifecycle configurations
- [SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SageMaker.5] SageMaker models should have network isolation enabled
- [SageMaker.6] SageMaker app image configurations should be tagged
- [SageMaker.7] SageMaker images should be tagged
- [SNS.4] SNS topic access policies should not allow public access
- [SQS.3] SQS queue access policies should not allow public access
- [SSM.4] SSM documents should not be public
- [SSM.5] SSM documents should be tagged
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [StepFunctions.2] Step Functions activities should be tagged
- [Transfer.4] Transfer Family agreements should be tagged
- [Transfer.5] Transfer Family certificates should be tagged
- [Transfer.6] Transfer Family connectors should be tagged
- [Transfer.7] Transfer Family profiles should be tagged
- [WAF.1] AWS WAF Classic Global Web ACL logging should be enabled
- [WAF.2] AWS WAF Classic Regional rules should have at least one condition
- [WAF.3] AWS WAF Classic Regional rule groups should have at least one rule

- [WAF.4] AWS WAF Classic Regional web ACLs should have at least one rule or rule group
- [WAF.6] AWS WAF Classic global rules should have at least one condition
- [WAF.7] AWS WAF Classic global rule groups should have at least one rule
- [WAF.8] AWS WAF Classic global web ACLs should have at least one rule or rule group
- [WAF.10] AWS WAF web ACLs should have at least one rule or rule group
- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled

Creating Security Hub CSPM resources with CloudFormation

AWS Security Hub CSPM integrates with AWS CloudFormation, which is a service that helps you model and set up your AWS resources so that you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want (such as automation rules), and AWS CloudFormation provisions and configures those resources for you.

When you use AWS CloudFormation, you can reuse your template to set up your Security Hub CSPM resources consistently and repeatedly. Describe your resources once, and then provision the same resources over and over in multiple AWS accounts and Regions.

Security Hub CSPM and AWS CloudFormation templates

To provision and configure resources for Security Hub CSPM and related services, you must understand how AWS CloudFormation templates work. Templates are text files in JSON or YAML format. These templates describe the resources that you want to provision in your AWS CloudFormation stacks.

If you're unfamiliar with JSON or YAML, you can use AWS CloudFormation Designer to help you get started with AWS CloudFormation templates. For more information, see What is AWS CloudFormation Designer? in the *AWS CloudFormation User Guide*.

You can create AWS CloudFormation templates for the following types of Security Hub CSPM resources:

- Enabling Security Hub CSPM
- Designating the delegated Security Hub CSPM administrator for an organization
- Specify the way your organization is configured in Security Hub CSPM
- Enabling a security standard

- Enabling cross-Region aggregation
- Creating a central configuration policy and associating it with accounts, organizational unit (OUs), or the root
- Creating a custom insight
- Creating an automation rule
- Customizing control parameters
- Subscribing to a third-party product integration

For more information, including examples of JSON and YAML templates for resources, see the AWS Security Hub CSPM resource type reference in the *AWS CloudFormation User Guide*.

Learn more about AWS CloudFormation

To learn more about AWS CloudFormation, see the following resources:

- AWS CloudFormation
- AWS CloudFormation User Guide
- AWS CloudFormation API Reference
- AWS CloudFormation Command Line Interface User Guide

Subscribing to Security Hub CSPM announcements with Amazon SNS

This section provides information about subscribing to AWS Security Hub CSPM announcements with Amazon Simple Notification Service (Amazon SNS) to receive notifications about Security Hub CSPM.

After subscribing, you will receive notifications about the following events (note the corresponding `AnnouncementType` for each event):

- `GENERAL` – General notifications about the Security Hub CSPM service.
- `UPCOMING_STANDARDS_CONTROLS` – Specified Security Hub CSPM controls or standards will be released soon. This type of announcement helps you prepare response and remediation workflows in advance of a release.
- `NEW_REGIONS` – Support for Security Hub CSPM is available in a new AWS Region.

- `NEW_STANDARDS_CONTROLS` – New Security Hub CSPM controls or standards have been added.
- `UPDATED_STANDARDS_CONTROLS` – Existing Security Hub CSPM controls or standards have been updated.
- `RETIRED_STANDARDS_CONTROLS` – Existing Security Hub CSPM controls or standards have been retired.
- `UPDATED_ASFF` – The AWS Security Finding Format (ASFF) syntax, fields, or values have been updated.
- `NEW_INTEGRATION` – New integrations with other AWS services or third-party products are available.
- `NEW_FEATURE` – New Security Hub CSPM features are available.
- `UPDATED_FEATURE` – Existing Security Hub CSPM features have been updated.

Notifications are available in all formats that Amazon SNS supports. You can subscribe to Security Hub CSPM announcements in all AWS Regions that Security Hub CSPM is available in.

A user must have `Subscribe` permissions to subscribe to an Amazon SNS topic. You can achieve this with Amazon SNS policies, IAM policies, or both. For more information, see IAM and Amazon SNS policies together in the *Amazon Simple Notification Service Developer Guide*.

> **Note**
>
> Security Hub CSPM sends Amazon SNS announcements about updates to the Security Hub CSPM service to any subscribed AWS account. To receive notifications about Security Hub CSPM findings, see Reviewing finding details and history in Security Hub CSPM.

You can subscribe to an Amazon Simple Queue Service (Amazon SQS) queue for an Amazon SNS topic, but you must use an Amazon SNS topic Amazon Resource Name (ARN) that is in the same Region. For more information, see Subscribing a queue to an Amazon SNS topic in the *Amazon Simple Queue Service Developer Guide*.

You can also use an AWS Lambda function to invoke events when you receive notifications. For more information, including sample function code, see Tutorial: Using AWS Lambda with Amazon Simple Notification Service in the *AWS Lambda Developer Guide*.

The Amazon SNS topic ARNs for each Region are as follows.

AWS Region	Amazon SNS topic ARN
US East (Ohio)	arn:aws:sns:us-east-2:291342846459:SecurityHubAnnouncements
US East (N. Virginia)	arn:aws:sns:us-east-1:088139225913:SecurityHubAnnouncements
US West (N. California)	arn:aws:sns:us-west-1:137690824926:SecurityHubAnnouncements
US West (Oregon)	arn:aws:sns:us-west-2:393883065485:SecurityHubAnnouncements
Africa (Cape Town)	arn:aws:sns:af-south-1:463142546776:SecurityHubAnnouncements
Asia Pacific (Hong Kong)	arn:aws:sns:ap-east-1:464812404305:SecurityHubAnnouncements
Asia Pacific (Hyderabad)	arn:aws:sns:ap-south-2:849907286123:SecurityHubAnnouncements
Asia Pacific (Jakarta)	arn:aws:sns:ap-southeast-3:627843640627:SecurityHubAnnouncements
Asia Pacific (Mumbai)	arn:aws:sns:ap-south-1:707356269775:SecurityHubAnnouncements

AWS Region	Amazon SNS topic ARN
Asia Pacific (Osaka)	`arn:aws:sns:ap-northeast-3:633550238216:SecurityHubAnnouncements`
Asia Pacific (Seoul)	`arn:aws:sns:ap-northeast-2:374299265323:SecurityHubAnnouncements`
Asia Pacific (Singapore)	`arn:aws:sns:ap-southeast-1:512267288502:SecurityHubAnnouncements`
Asia Pacific (Sydney)	`arn:aws:sns:ap-southeast-2:475730049140:SecurityHubAnnouncements`
Asia Pacific (Tokyo)	`arn:aws:sns:ap-northeast-1:592469075483:SecurityHubAnnouncements`
Canada (Central)	`arn:aws:sns:ca-central-1:137749997395:SecurityHubAnnouncements`
China (Beijing)	`arn:aws-cn:sns:cn-north-1:672341567257:SecurityHubAnnouncements`
China (Ningxia)	`arn:aws-cn:sns:cn-northwest-1:672534482217:SecurityHubAnnouncements`
Europe (Frankfurt)	`arn:aws:sns:eu-central-1:871975303681:SecurityHubAnnouncements`

AWS Region	Amazon SNS topic ARN
Europe (Ireland)	arn:aws:sns:eu-west-1:705756202095:SecurityHubAnnouncements
Europe (London)	arn:aws:sns:eu-west-2:883600840440:SecurityHubAnnouncements
Europe (Milan)	arn:aws:sns:eu-south-1:151363035580:SecurityHubAnnouncements
Europe (Paris)	arn:aws:sns:eu-west-3:313420042571:SecurityHubAnnouncements
Europe (Spain)	arn:aws:sns:eu-south-2:777487947751:SecurityHubAnnouncements
Europe (Stockholm)	arn:aws:sns:eu-north-1:191971010772:SecurityHubAnnouncements
Europe (Zurich)	arn:aws:sns:eu-central-2:704347005078:SecurityHubAnnouncements
Israel (Tel Aviv)	arn:aws:sns:il-central-1:726652212146:SecurityHubAnnouncements
Middle East (Bahrain)	arn:aws:sns:me-south-1:585146626860:SecurityHubAnnouncements

AWS Region	Amazon SNS topic ARN
Middle East (UAE)	`arn:aws:sns:me-central-1:431548502100:SecurityHubAnnouncements`
South America (São Paulo)	`arn:aws:sns:sa-east-1:359811883282:SecurityHubAnnouncements`
AWS GovCloud (US-East)	`arn:aws-us-gov:sns:us-gov-east-1:239368469855:SecurityHubAnnouncements`
AWS GovCloud (US-West)	`arn:aws-us-gov:sns:us-gov-west-1:239334163374:SecurityHubAnnouncements`

Messages are typically the same across Regions within a partition, so you can subscribe to one Region in each partition to receive announcements that affect all Regions in that partition. Announcements associated with member accounts are not replicated in the administrator account. As a result, each account, including the administrator account, will only have one copy of each announcement. You can decide which account you want to use to subscribe to Security Hub CSPM announcements.

For information about the cost of subscribing to Security Hub CSPM announcements, see Amazon SNS pricing.

Subscribing to Security Hub CSPM announcements (console)

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the Region list, choose the Region in which you want to subscribe to Security Hub CSPM announcements. This example uses the `us-west-2` Region.
3. In the navigation pane, choose **Subscriptions**, and then choose **Create subscription**.
4. Enter the topic ARN into the **Topic ARN** box. For example, `arn:aws:sns:us-west-2:393883065485:SecurityHubAnnouncements`.

5. For **Protocol**, choose how you want to receive Security Hub CSPM announcements. If you choose **Email**, for **Endpoint**, enter the email address that you want to use to receive announcements.
6. Choose **Create subscription**.
7. Confirm the subscription. For example, if you chose email protocol, Amazon SNS will send a subscription confirmation message to the email you provided.

Subscribing to Security Hub CSPM announcements (AWS CLI)

1. Run the following command:

```
 aws  sns --region us-west-2 subscribe --topic-arn arn:aws:sns:us-
west-2:393883065485:SecurityHubAnnouncements --protocol email --notification-
endpoint your_email@your_domain.com
```

2. Confirm the subscription. For example, if you chose email protocol, Amazon SNS will send a subscription confirmation message to the email you provided.

Amazon SNS message format

The following examples show Security Hub CSPM announcements from Amazon SNS about the introduction of new security controls. Message content varies based on announcement type, but the format is the same for all announcement types. Optionally, a `Link` field that provides details about the announcement may be included.

Example: Security Hub CSPM announcement for new controls (email protocol)

```
{
"AnnouncementType":"NEW_STANDARDS_CONTROLS",
"Title":"[New Controls] 36 new Security Hub CSPM controls added to the AWS Foundational
 Security Best Practices standard",
"Description":"We have added 36 new controls to the AWS Foundational Security Best
 Practices standard. These include controls for Amazon Auto Scaling (AutoScaling.3,
 AutoScaling.4, AutoScaling.6), AWS CloudFormation (CloudFormation.1), Amazon
 CloudFront (CloudFront.10), Amazon Elastic Compute Cloud (Amazon EC2) (EC2.23,
 EC2.24, EC2.27), Amazon Elastic Container Registry (Amazon ECR) (ECR.1, ECR.2), Amazon
 Elastic Container Service (Amazon ECS) (ECS.3, ECS.4, ECS.5, ECS.8, ECS.10, ECS.12),
 Amazon Elastic File System (Amazon EFS) (EFS.3, EFS.4), Amazon Elastic Kubernetes
 Service (Amazon EKS) (EKS.2), Elastic Load Balancing (ELB.12, ELB.13, ELB.14), Amazon
```

```
 Kinesis (Kinesis.1), AWS Network Firewall (NetworkFirewall.3, NetworkFirewall.4,
 NetworkFirewall.5), Amazon OpenSearch Service (OpenSearch.7), Amazon Redshift
 (Redshift.9),
Amazon Simple Storage Service (Amazon S3) (S3.13), Amazon Simple Notification Service
 (SNS.2), AWS WAF (WAF.2, WAF.3, WAF.4, WAF.6, WAF.7, WAF.8). If you enabled the AWS
 Foundational Security Best Practices standard in an account and configured Security
 Hub CSPM to automatically enable new controls, these controls are enabled by default.
 Availability of controls can vary by Region. "
}
```

Example: Security Hub CSPM announcement for new controls (email-JSON protocol)

```
{
  "Type" : "Notification",
  "MessageId" : "d124c9cf-326a-5931-9263-92a92e7af49f",
  "TopicArn" : "arn:aws:sns:us-west-2:393883065485:SecurityHubAnnouncements",
  "Message" : "{\"AnnouncementType\":\"NEW_STANDARDS_CONTROLS\",\"Title\":\"[New
 Controls] 36 new Security Hub CSPM controls added to the AWS Foundational Security
 Best Practices standard\",\"Description\":\"We have added 36 new controls to the
 AWS Foundational Security Best Practices standard. These include controls for Amazon
 Auto Scaling (AutoScaling.3, AutoScaling.4, AutoScaling.6), AWS CloudFormation
 (CloudFormation.1), Amazon CloudFront (CloudFront.10), Amazon Elastic Compute Cloud
 (Amazon EC2) (EC2.23, EC2.24, EC2.27), Amazon Elastic Container Registry (Amazon ECR)
 (ECR.1, ECR.2), Amazon Elastic Container Service (Amazon ECS) (ECS.3, ECS.4, ECS.5,
 ECS.8, ECS.10, ECS.12), Amazon Elastic File System (Amazon EFS) (EFS.3, EFS.4), Amazon
 Elastic Kubernetes Service (Amazon EKS) (EKS.2), Elastic Load Balancing (ELB.12,
 ELB.13, ELB.14), Amazon Kinesis (Kinesis.1), AWS Network Firewall (NetworkFirewall.3,
 NetworkFirewall.4, NetworkFirewall.5), Amazon OpenSearch Service (OpenSearch.7),
 Amazon Redshift (Redshift.9),
Amazon Simple Storage Service (Amazon S3) (S3.13), Amazon Simple Notification Service
 (SNS.2), AWS WAF (WAF.2, WAF.3, WAF.4, WAF.6, WAF.7, WAF.8). If you enabled the AWS
 Foundational Security Best Practices standard in an account and configured SSecurity
 Hub CSPM to automatically enable new controls, these controls are enabled by default.
 Availability of controls can vary by Region. \"}",
  "Timestamp" : "2022-08-04T19:11:12.652Z",
  "SignatureVersion" : "1",
  "Signature" :
 "HTHgNFRYMetCvisulgLM4CVySvK9qCXFPHQDxYl9tuCFQuIrd7YO4m4YFR28XKMgzqrF20YP
+EilipUm2SOTpEEtOTekU5bn74+YmNZfwr4aPFx0vUuQCVOshmHl37hjkiLjhCg/t53QQiLfP7MH
+MTXIUPR37k5SuFCXvjpRQ8ynV532AH3Wpv0HmojDLMg+eg51V1fUsOG8yiJVCBEJhJ1yS
+gkwJdhRk2UQab9RcAmE6COK3hRWcjDwqTXz5nR6Ywv1ZqZfLIl7gYKslt+jsyd/k+7kOqGmOJRDr7qhE7H
+7vaGRLOptsQnbW8VmeYnDbahEO8FV+Mp1rpV+7Qg==",
```

```
  "SigningCertURL" : "https://sns.us-west-2.amazonaws.com/
SimpleNotificationService-56e67fcb41f6fec09b0196692625d385.pem",
  "UnsubscribeURL" : "https://sns.us-west-2.amazonaws.com/?
Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-
west-2:393883065485:SecurityHubAnnouncements:9d0230d7-d582-451d-9f15-0c32818bf61f"
}
```

Disabling Security Hub CSPM

You can disable AWS Security Hub CSPM by using the Security Hub CSPM console or the Security Hub API. If you disable Security Hub CSPM, you can enable it again later.

If your organization uses central configuration, the delegated Security Hub CSPM administrator can create configuration policies that disable Security Hub CSPM for specific accounts and organizational units (OUs) and keep Security Hub CSPM enabled for others. Configuration policies affect the home Region and all linked Regions. For more information, see Understanding central configuration in Security Hub CSPM.

If you disable Security Hub CSPM for an account, the following occurs:

- All Security Hub CSPM standards and controls are disabled for the account.
- Security Hub CSPM stops generating, updating, and ingesting findings for the account.
- After 30 days, Security Hub CSPM permanently deletes all existing archived findings for the account. The findings cannot be recovered by using Security Hub CSPM.
- After 90 days, Security Hub CSPM permanently deletes all existing active findings for the account. The findings cannot be recovered by using Security Hub CSPM.
- After 90 days, Security Hub CSPM permanently deletes all existing insights and Security Hub CSPM configuration settings for the account. The data and settings cannot be recovered.

To retain existing findings, you can export the findings to an S3 bucket before you disable Security Hub CSPM. You can do this by using a custom action with an Amazon EventBridge rule. For more information, see Using EventBridge for automated response and remediation.

If you re-enable Security Hub CSPM within 90 days of disabling it for an account, you regain access to existing active findings, as well as insights and Security Hub CSPM configuration settings for the account. If you re-enable Security Hub CSPM within 30 days, you also regain access to existing archived findings for the account. However, existing findings might be inaccurate because they

will reflect the state of your AWS environment when you disabled Security Hub CSPM. In addition, as you re-enable individual standards and controls, Security Hub CSPM might initially generate duplicate findings for specific AWS resources, depending on the standards and controls that you enable. For these reasons, we recommend that you do one of the following:

- Change the workflow status of all existing findings to `RESOLVED` before you disable Security Hub CSPM. For more information, see Setting the workflow status of findings.
- Disable all standards at least six days before you disable Security Hub CSPM. Security Hub CSPM then archives all existing findings on a best-effort basis, typically within three to five days. For more information, see Disabling a standard.

You can't disable Security Hub CSPM in the following cases:

- Your account is the delegated Security Hub CSPM administrator account for an organization. If you use central configuration, you can't associate a configuration policy that disables Security Hub CSPM for the delegated administrator account. The association can succeed for other accounts, but Security Hub CSPM doesn't apply the policy to the delegated administrator account.
- Your account is a Security Hub CSPM administrator account by invitation, and you have member accounts. Before you can disable Security Hub CSPM, you must disassociate all of your member accounts. To learn how, see the section called “Disassociating member accounts”.

Before the owner of a member account can disable Security Hub CSPM, the account must disassociate from its administrator account. For an organization account, only the administrator account can disassociate a member account. For more information, see the section called “Disassociating organization member accounts”. For a manually invited account, either the administrator account or the member account can disassociate the account. For more information, see the section called “Disassociating member accounts” or the section called “Disassociating from an administrator account”. Disassociation isn't required if you use central configuration because the Security Hub CSPM administrator can create a policy that disables Security Hub CSPM for specific member accounts.

When you disable Security Hub CSPM for an account, it's disabled only in the current AWS Region. However, if you use central configuration to disable Security Hub CSPM for specific accounts, it's disabled in the home Region and all linked Regions.

To disable Security Hub CSPM, choose your preferred method and follow the steps.

Security Hub CSPM console

Follow these steps to disable Security Hub CSPM by using the console.

To disable Security Hub CSPM

1. Open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, under **Settings**, choose **General**.
3. In the **Disable Security Hub CSPM** section, choose **Disable Security Hub CSPM**.
4. When prompted for confirmation, choose **Disable Security Hub CSPM**.

Security Hub API

To disable Security Hub CSPM programmatically, use the DisableSecurityHub operation of the AWS Security Hub API. Or, if you're using the AWS CLI, run the disable-security-hub command. For example, the following command disables Security Hub CSPM in the current AWS Region:

```
$ aws securityhub disable-security-hub
```

Security in AWS Security Hub

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Security Hub, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Security Hub. The following topics show you how to configure Security Hub to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Security Hub resources.

Topics

Data protection in AWS Security Hub

The AWS shared responsibility model applies to data protection in AWS Security Hub. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the

AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Security Hub or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Security Hub is a multi-tenant service offering. To ensure data protection, Security Hub encrypts data at rest and data in transit between component services.

AWS Identity and Access Management for Security Hub

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in)

and *authorized* (have permissions) to use Security Hub resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Security Hub.

Service user – If you use the Security Hub service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Security Hub features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Security Hub, see Troubleshooting AWS Security Hub identity and access.

Service administrator – If you're in charge of Security Hub resources at your company, you probably have full access to Security Hub. It's your job to determine which Security Hub features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Security Hub, see How Security Hub works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Security Hub. To view example Security Hub identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Security Hub.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and AWS Multi-factor authentication in IAM in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. To temporarily assume an IAM role in the AWS Management Console, you can switch from a user to an IAM role (console). You can assume a

role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Methods to assume a role in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Create a role for a third-party identity provider (federation) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can

perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user

or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.

- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – RCPs are JSON policies that you can use to set the maximum available permissions for resources in your accounts without updating the IAM policies attached to each resource that you own. The RCP limits permissions for resources in member accounts and can impact the effective permissions for identities, including the AWS account root user, regardless of whether they belong to your organization. For more information about Organizations and RCPs, including a list of AWS services that support RCPs, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Security Hub works with IAM

Before you use AWS Identity and Access Management (IAM) to manage access to AWS Security Hub, learn which IAM features are available to use with Security Hub.

IAM features you can use with AWS Security Hub

IAM feature	Security Hub support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	No
Policy condition keys	Yes
Access control lists (ACLs)	No
Attribute-based access control (ABAC) – tags in policies	Yes
Temporary credentials	Yes
Forward access sessions (FAS)	Yes
Service roles	No
Service-linked roles	Yes

For a high-level view of how Security Hub and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Security Hub

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an

identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Security Hub supports identity-based policies. For more information, see Identity-based policy examples for AWS Security Hub.

Resource-based policies for Security Hub

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Security Hub does not support resource-based policies. You can't attach an IAM policy directly to a Security Hub resource.

Policy actions for Security Hub

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation.

There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in Security Hub use the following prefix before the action:

```
securityhub:
```

For example, to grant a user permission to enable Security Hub, which is an action that corresponds to the `EnableSecurityHub` operation of the Security Hub API, include the `securityhub:EnableSecurityHub` action in their policy. Policy statements must include either an `Action` or `NotAction` element. Security Hub defines its own set of actions that describe tasks that you can perform with this service.

```
"Action": "securityhub:EnableSecurityHub"
```

To specify multiple actions in a single statement, separate them with commas. For example:

```
"Action": [
      "securityhub:EnableSecurityHub",
      "securityhub:BatchEnableStandards"
```

You can also specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Get`, include the following action:

```
"Action": "securityhub:Get*"
```

However, as a best practice, you should create policies that follow the principle of least privilege. In other words, you should create policies that include only the permissions that are required to perform a specific task.

The user must have access to the `DescribeStandardsControl` operation in order to have access to `BatchGetSecurityControls`, `BatchGetStandardsControlAssociations`, and `ListStandardsControlAssociations`.

The user must have access to the `UpdateStandardsControls` operation in order to have access to `BatchUpdateStandardsControlAssociations`, and `UpdateSecurityControl`.

For a list of Security Hub actions, see Actions defined by AWS Security Hub in the *Service Authorization Reference*. For examples of policies that specify Security Hub actions, see Identity-based policy examples for AWS Security Hub.

Policy resources for Security Hub

Supports policy resources: No

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

Security Hub defines the following resource types:

- Hub
- Product
- Finding aggregator, also referred to as a *cross-Region aggregator*
- Automation rule
- Configuration policy

You can specify these types of resources in policies by using ARNs.

For a list of Security Hub resource types and the ARN syntax for each one, see Resource types defined by AWS Security Hub in the *Service Authorization Reference*. To learn which actions you can specify for each type of resource, see Actions defined by AWS Security Hub in the *Service Authorization Reference*. For examples of policies that specify resources, see Identity-based policy examples for AWS Security Hub.

Policy condition keys for Security Hub

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

For a list of Security Hub condition keys, see Condition keys for AWS Security Hub in the *Service Authorization Reference*. To learn which actions and resources you can use a condition key with, see Actions defined by AWS Security Hub. For examples of policies that use condition keys, see Identity-based policy examples for AWS Security Hub.

Access control lists (ACLs) in Security Hub

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Security Hub doesn't support ACLs, which means you can't attach an ACL to a Security Hub resource.

Attribute-based access control (ABAC) with Security Hub

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

You can attach tags to Security Hub resources. You can also control access to resources by providing tag information in the `Condition` element of a policy.

For information about tagging Security Hub resources, see Tagging Security Hub resources. For an example of an identity-based policy that controls access to a resource based on tags, see Identity-based policy examples for AWS Security Hub.

Using temporary credentials with Security Hub

Supports temporary credentials: Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your

company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switch from a user to an IAM role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

Security Hub supports the use of temporary credentials.

Forward access sessions for Security Hub

Supports forward access sessions (FAS): Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

For example, Security Hub makes FAS requests to downstream AWS services when you integrate Security Hub with AWS Organizations and when you designate the delegated Security Hub administrator account for an organization in Organizations.

For other tasks, Security Hub uses a service-linked role to perform actions on your behalf. For details about this role, see Service-linked roles for AWS Security Hub.

Service roles for Security Hub

Security Hub doesn't assume or use service roles. To perform actions on your behalf, Security Hub uses a service-linked role. For details about this role, see Service-linked roles for AWS Security Hub.

> **Warning**
>
> Changing the permissions for a service role may create operational issues with your use of Security Hub. Edit service roles only when Security Hub provides guidance to do so.

Service-linked roles for Security Hub

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

Security Hub uses a service-linked role to perform actions on your behalf. For details about this role, see Service-linked roles for AWS Security Hub.

Identity-based policy examples for AWS Security Hub

By default, users and roles don't have permission to create or modify Security Hub resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating Policies on the JSON Tab in the *IAM User Guide*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Security Hub resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Security Hub console

To access the AWS Security Hub console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Security Hub resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that those users and roles can use the Security Hub console, also attach the following AWS managed policy to the entity. For more information, see Adding permissions to a user in the *IAM User Guide*:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "securityhub:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "securityhub.amazonaws.com"
                }
            }
        }
    ]
}
```

Example: Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Example: Allow users to create and manage a configuration policy

This example shows how you might create an IAM policy that allows a user to create, view, update, and delete configuration policies. This example policy also allows the user to start, stop, and view

policy associations. For this IAM policy to work, the user must be the delegated Security Hub administrator for an organization.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAndUpdateConfigurationPolicy",
            "Effect": "Allow",
            "Action": [
                "securityhub:CreateConfigurationPolicy",
                "securityhub:UpdateConfigurationPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewConfigurationPolicy",
            "Effect": "Allow",
            "Action": [
                "securityhub:GetConfigurationPolicy",
                "securityhub:ListConfigurationPolicies"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteConfigurationPolicy",
            "Effect": "Allow",
            "Action": [
                "securityhub:DeleteConfigurationPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewConfigurationPolicyAssociation",
            "Effect": "Allow",
            "Action": [
                "securityhub:BatchGetConfigurationPolicyAssociations",
                "securityhub:GetConfigurationPolicyAssociation",
                "securityhub:ListConfigurationPolicyAssociations"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "UpdateConfigurationPolicyAssociation",
            "Effect": "Allow",
            "Action": [
                "securityhub:StartConfigurationPolicyAssociation",
                "securityhub:StartConfigurationPolicyDisassociation"
            ],
            "Resource": "*"
        }
    ]
}
```

Example: Allow users to view findings

This example shows how you might create an IAM policy that allows a user to view Security Hub findings.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReviewFindings",
            "Effect": "Allow",
            "Action": [
                "securityhub:GetFindings"
            ],
            "Resource": "*"
        }
    ]
}
```

Example: Allow users to create and manage automation rules

This example shows how you might create an IAM policy that allows a user to create, view, update, and delete Security Hub automation rules. For this IAM policy to work, the user must be a Security

Hub administrator. To limit permissions— for example, to allow a user to only view automation rules—you can remove the create, update, and delete permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAndUpdateAutomationRules",
            "Effect": "Allow",
            "Action": [
                "securityhub:CreateAutomationRule",
                "securityhub:BatchUpdateAutomationRules"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewAutomationRules",
            "Effect": "Allow",
            "Action": [
                "securityhub:BatchGetAutomationRules",
                "securityhub:ListAutomationRules"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteAutomationRules",
            "Effect": "Allow",
            "Action": [
                "securityhub:BatchDeleteAutomationRules"
            ],
            "Resource": "*"
        }
    ]
}
```

Service-linked roles for AWS Security Hub

AWS Security Hub uses an AWS Identity and Access Management (IAM) service-linked role named `AWSServiceRoleForSecurityHub`. This service-linked role is an IAM role that's linked directly to

Security Hub. It's predefined by Security Hub, and it includes all the permissions that Security Hub requires to call other AWS services and monitor AWS resources on your behalf. Security Hub uses this service-linked role in all the AWS Regions where Security Hub is available.

A service-linked role makes setting up Security Hub easier because you don't have to manually add the necessary permissions. Security Hub defines the permissions of its service-linked role, and unless defined otherwise, only Security Hub can assume the role. The defined permissions include the trust policy and the permissions policy, and you can't attach that permissions policy to any other IAM entity.

To review the details of the service-linked role, you can use the Security Hub console. In the navigation pane, choose **General** under **Settings**. Then, in the **Service permissions** section, choose **View service permissions**.

You can delete the Security Hub service-linked role only after you disable Security Hub in all the Regions where it's enabled. This protects your Security Hub resources because you can't inadvertently remove permissions to access them.

For information about other services that support service-linked roles, see AWS services that work with IAM in the *IAM User Guide* and locate the services that have **Yes** in the **Service-linked roles** column. Choose a **Yes** with a link to review the service-linked role documentation for that service.

Topics

Service-linked role permissions for Security Hub

Security Hub uses the service-linked role named `AWSServiceRoleForSecurityHub`. It's a service-linked role required for AWS Security Hub to access your resources. This service-linked role allows Security Hub to perform tasks such as receive findings from other AWS services and configure the requisite AWS Config infrastructure to run security checks for controls. The `AWSServiceRoleForSecurityHub` service-linked role trusts the `securityhub.amazonaws.com` service to assume the role.

The AWSServiceRoleForSecurityHub service-linked role uses the managed policy AWSSecurityHubServiceRolePolicy.

You must grant permissions to allow an IAM identity (such as a role, group, or user) to create, edit, or delete a service-linked role. For the AWSServiceRoleForSecurityHub service-linked role to be successfully created, the IAM identity that you use to access Security Hub must have the required permissions. To grant the required permissions, attach the following policy to the IAM identity.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "securityhub:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "securityhub.amazonaws.com"
                }
            }
        }
    ]
}
```

Creating a service-linked role for Security Hub

The AWSServiceRoleForSecurityHub service-linked role is created automatically when you enable Security Hub for the first time or you enable Security Hub in a Region where you didn't previously enable it. You can also create the AWSServiceRoleForSecurityHub service-linked role manually by using the IAM console, the IAM CLI, or the IAM API. For more information about creating the role manually, see Creating a service-linked role in the *IAM User Guide*.

> **Important**
>
> The service-linked role that's created for a Security Hub administrator account doesn't apply to associated Security Hub member accounts.

Editing a service-linked role for Security Hub

Security Hub doesn't allow you to edit the `AWSServiceRoleForSecurityHub` service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role by using IAM. For more information, see Editing a service-linked role in the *IAM User Guide*.

Deleting a service-linked role for Security Hub

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete the role. That way, you don't have an unused entity that isn't actively monitored or maintained.

When you disable Security Hub, Security Hub doesn't automatically delete the `AWSServiceRoleForSecurityHub` service-linked role for you. If you enable Security Hub again, the service can then start using the existing service-linked role again. If you no longer need to use Security Hub, you can manually delete the service-linked role.

> **Important**
>
> Before you delete the `AWSServiceRoleForSecurityHub` service-linked role, you must first disable Security Hub in all the Regions where it's enabled. For more information, see Disabling Security Hub CSPM. If Security Hub isn't disabled when you try to delete the service-linked role, the deletion fails.

To delete the `AWSServiceRoleForSecurityHub` service-linked role, you can use the IAM console, the IAM CLI, or the IAM API. For more information, see Deleting a service-linked role in the *IAM User Guide*.

AWS managed policies for Security Hub

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: AWSSecurityHubFullAccess

You can attach the `AWSSecurityHubFullAccess` policy to your IAM identities.

This policy grants administrative permissions that allow a principal full access to all Security Hub CSPM actions. This policy must be attached to a principal before they enable Security Hub CSPM manually for their account. For example, principals with these permissions can both view and update the status of findings. They can also configure custom insights, enable integrations, and enable and disable standards and controls. Principals for an administrator account can also manage member accounts.

Permissions details

This policy includes the following permissions:

- `securityhub` – Allows principals full access to all Security Hub CSPM actions.
- `guardduty` – Allows principals to get information about account status in Amazon GuardDuty.
- `iam` – Allows principals to create a service-linked role for Security Hub CSPM and Security Hub.
- `inspector` – Allows principals to get information about account status in Amazon Inspector.

- `pricing` – Allows principals to get a price list of AWS services and products.

To review the permissions for this policy, see AWSSecurityHubFullAccess in the *AWS Managed Policy Reference Guide*.

AWS managed policy: AWSSecurityHubReadOnlyAccess

You can attach the `AWSSecurityHubReadOnlyAccess` policy to your IAM identities.

This policy grants read-only permissions that allow users to view information in Security Hub CSPM. Principals with this policy attached cannot make any updates in Security Hub CSPM. For example, principals with these permissions can view the list of findings associated with their account, but cannot change the status of a finding. They can view the results of insights, but cannot create or configure custom insights. They cannot configure controls or product integrations.

Permissions details

This policy includes the following permissions:

- `securityhub` – Allows users to perform actions that return a list of items or details about an item. This includes API operations that start with `Get`, `List`, or `Describe`.

To review the permissions for this policy, see AWSSecurityHubReadOnlyAccess in the *AWS Managed Policy Reference Guide*.

AWS managed policy: AWSSecurityHubOrganizationsAccess

You can attach the `AWSSecurityHubOrganizationsAccess` policy to your IAM identities.

This policy grants administrative permissions to enable and manage Security Hub and Security Hub CSPM for an organization in AWS Organizations. The permissions for this policy allow the organization management account to designate the delegated administrator account for Security Hub and Security Hub CSPM. They also allow the delegated administrator account to enable organization accounts as member accounts.

This policy only provides permissions for AWS Organizations. The organization management account and delegated administrator account also require permissions for associated actions. These permissions can be granted using the `AWSSecurityHubFullAccess` managed policy.

Permissions details

This policy includes the following permissions:

- `organizations:ListAccounts` – Allows principals to retrieve the list of accounts that are part of an organization.
- `organizations:DescribeOrganization` – Allows principals to retrieve information about the organization.
- `organizations:ListRoots` – Allows principals to list the root of an organization.
- `organizations:ListDelegatedAdministrators` – Allows principals to list the delegated administrator of an organization.
- `organizations:ListAWSServiceAccessForOrganization` – Allows principals to list the AWS services that an organization uses.
- `organizations:ListOrganizationalUnitsForParent` – Allows principals to list the child organizational units (OU) of a parent OU.
- `organizations:ListAccountsForParent` – Allows principals to list the child accounts of a parent OU.
- `organizations:ListParents` – Lists the root or organizational units (OUs) that serve as the immediate parent of the specified child OU or account.
- `organizations:DescribeAccount` – Allows principals to retrieve information about an account in the organization.
- `organizations:DescribeOrganizationalUnit` – Allows principals to retrieve information about an OU in the organization.
- `organizations:ListPolicies` – Retrieves the list of all policies in an organization of a specified type.
- `organizations:ListPoliciesForTarget` – Lists the policies that are directly attached to the specified target root, organizational unit (OU), or account.
- `organizations:ListTargetsForPolicy` – Lists all the roots, organizational units (OUs), and accounts that the specified policy is attached to.
- `organizations:EnableAWSServiceAccess` – Allows principals to enable the integration with Organizations.
- `organizations:RegisterDelegatedAdministrator` – Allows principals to designate the delegated administrator account.
- `organizations:DeregisterDelegatedAdministrator` – Allows principals to remove the delegated administrator account.

- `organizations:DescribePolicy` – Retrieves information about a policy.
- `organizations:DescribeEffectivePolicy` – Returns the contents of the effective policy for specified policy type and account.
- `organizations:CreatePolicy` – Creates a policy of a specified type that you can attach to a root, an organizational unit (OU), or an individual AWS account.
- `organizations:UpdatePolicy` – Updates an existing policy with a new name, description, or content.
- `organizations:DeletePolicy` – Deletes the specified policy from your organization.
- `organizations:AttachPolicy` – Attaches a policy to a root, an organizational unit (OU), or an individual account.
- `organizations:DetachPolicy` – Detaches a policy from a target root, organizational unit (OU), or account.
- `organizations:EnablePolicyType` – Enables a policy type in a root.
- `organizations:DisablePolicyType` – Disables an organizational policy type in a root.
- `organizations:TagResource` – Adds one or more tags to a specified resource.
- `organizations:UntagResource` – Removes any tags with the specified keys from a specified resource.
- `organizations:ListTagsForResource` – Lists tags that are attached to a specified resource.

To review the permissions for this policy, see AWSSecurityHubOrganizationsAccess in the *AWS Managed Policy Reference Guide*.

AWS managed policy: AWSSecurityHubServiceRolePolicy

You can't attach `AWSSecurityHubServiceRolePolicy` to your IAM entities. This policy is attached to a service-linked role that allows Security Hub CSPM to perform actions on your behalf. For more information, see the section called “Service-linked roles”.

This policy grants administrative permissions that allow the service-linked role to perform tasks such as run security checks for Security Hub CSPM controls.

Permissions details

This policy includes the following permissions:

- `cloudtrail` – Retrieve information about CloudTrail trails.

- `cloudwatch` – Retrieve current CloudWatch alarms.
- `logs` – Retrieve metric filters for CloudWatch logs.
- `sns` – Retrieve the list of subscriptions to an SNS topic.
- `config` – Retrieve information about configuration recorders, resources, and AWS Config rules. Also allows the service-linked role to create and delete AWS Config rules, and to run evaluations against the rules.
- `iam` – Retrieve and generate credential reports for accounts.
- `organizations` – Retrieve account and organizational unit (OU) information for an organization.
- `securityhub` – Retrieve information about how the Security Hub CSPM service, standards, and controls are configured.
- `tag` – Retrieve information about resource tags.

To review the permissions for this policy, see AWSSecurityHubServiceRolePolicy in the *AWS Managed Policy Reference Guide*.

AWS managed policy: AWSSecurityHubV2ServiceRolePolicy

> **Note**
>
> Security Hub is in preview release and subject to change.

This policy allows Security Hub to manage AWS Config rules and Security Hub resources for your organization and on your behalf. This policy is attached to a service-linked role that allows the service to perform actions on your behalf. You cannot attach this policy to your IAM identities. For more information, see the section called “Service-linked roles”.

Permissions details

This policy includes the following permissions:

- `config` – Manage service-linked configuration recorders for Security Hub resources.
- `iam` – Create the service-linked role for AWS Config.
- `organizations` – Retrieve account and organizational unit (OU) information for an organization.

- `securityhub` – Manage the Security Hub configuration.
- `tag` – Retrieve information about resource tags.

To review the permissions for this policy, see AWSSecurityHubV2ServiceRolePolicy in the *AWS Managed Policy Reference Guide*.

Security Hub updates to AWS managed policies

The following table provides details about updates to AWS managed policies for AWS Security Hub and Security Hub CSPM since this service began tracking these changes. For automatic alerts about updates to the policies, subscribe to the RSS feed on the Security Hub document history page.

Change	Description	Date
AWSSecurityHubOrganizationsAccess – Update to an existing policy	Security Hub added new permissions to the policy. The permissions allow the organization management to enable and manage Security Hub and Security Hub CSPM for an organization.	June 17, 2025
AWSSecurityHubFullAccess – Update to an existing policy	Security Hub CSPM added new permissions that allow principals to create a service-linked role for Security Hub.	June 17, 2025
AWSSecurityHubV2ServiceRolePolicy – New policy	Security Hub added a new policy to allow Security Hub to manage AWS Config rules and Security Hub resources for a customer's organization and on the customer's behalf. Security Hub is in preview release and subject to change.	June 17, 2025

Change	Description	Date
AWSSecurityHubFullAccess – Update to an existing policy	Security Hub CSPM updated the policy to get pricing details for AWS services and products.	April 24, 2024
AWSSecurityHubReadOnlyAccess – Update to an existing policy	Security Hub CSPM updated this managed policy by adding a `Sid` field.	February 22, 2024
AWSSecurityHubFullAccess – Update to an existing policy	Security Hub CSPM updated the policy so it can determine if Amazon GuardDuty and Amazon Inspector are enabled in an account. This helps customers bring together security-related information from multiple AWS services.	November 16, 2023
AWSSecurityHubOrganizationsAccess – Update to an existing policy	Security Hub CSPM updated the policy to grant additional permissions to allow read-only access to AWS Organizations delegated administrator functionality. This includes details like the root, organizational units (OUs), accounts, organizational structure, and service access.	November 16, 2023

Change	Description	Date
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub CSPM added the `BatchGetSecurityControls` , `DisassociateFromAdministratorAccount` , and `UpdateSecurityControl` permissions to read and update customizable security control properties.	November 26, 2023
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub CSPM added the `tag:GetResources` permission to read resource tags related to findings.	November 7, 2023
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub CSPM added the `BatchGetStandardsControlAssociations` permission to get information about the enablement status of a control in a standard.	September 27, 2023
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub CSPM added new permissions to get AWS Organizations data and read and update Security Hub CSPM configurations, including standards and controls.	September 20, 2023

Change	Description	Date
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub CSPM moved the existing `config:DescribeConfigRuleEvaluationStatus` permission to a different statement within the policy. The `config:DescribeConfigRuleEvaluationStatus` permission is now applied to all resources.	March 17, 2023
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub CSPM moved the existing `config:PutEvaluations` permission to a different statement within the policy. The `config:PutEvaluations` permission is now applied to all resources.	July 14, 2021
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub CSPM added a new permission to allow the service-linked role to deliver evaluation results to AWS Config.	June 29, 2021
AWSSecurityHubServiceRolePolicy – Added to the list of managed policies	Added information about the managed policy AWSSecurityHubServiceRolePolicy, which is used by the Security Hub CSPM service-linked role.	June 11, 2021

Change	Description	Date
AWSSecurityHubOrganizationsAccess – New policy	Security Hub CSPM added a new policy that grants permissions that are needed for the Security Hub CSPM integration with Organizations.	March 15, 2021
Security Hub CSPM started tracking changes	Security Hub CSPM started tracking changes for its AWS managed policies.	March 15, 2021

Troubleshooting AWS Security Hub identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS Security Hub and IAM.

Topics

- I am not authorized to perform an action in Security Hub
- I am not authorized to perform iam:PassRole
- I want programmatic access to Security Hub
- I'm an administrator and want to allow others to access Security Hub
- I want to allow people outside my AWS account to access my Security Hub resources

I am not authorized to perform an action in Security Hub

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your sign-in credentials.

The following example error occurs when the user `mateojackson` tries to use the console to view details about a *`widget`* but does not have `securityhub:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 securityhub:GetWidget on resource: my-example-widget
```

In this case, Mateo asks his administrator to update his policies to allow him to access the *my-example-widget* resource using the `securityhub:GetWidget` action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Security Hub.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Security Hub. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want programmatic access to Security Hub

Users need programmatic access if they want to interact with AWS outside of the AWS Management Console. The way to grant programmatic access depends on the type of user that's accessing AWS.

To grant users programmatic access, choose one of the following options.

Which user needs programmatic access?	To	By
Workforce identity (Users managed in IAM Identity Center)	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use.

Which user needs programmatic access?	To	By
		• For the AWS CLI, see Configuring the AWS CLI to use AWS IAM Identity Center in the *AWS Command Line Interface User Guide*. • For AWS SDKs, tools, and AWS APIs, see IAM Identity Center authentication in the *AWS SDKs and Tools Reference Guide*.
IAM	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions in Using temporary credentials with AWS resources in the *IAM User Guide*.

Which user needs programmatic access?	To	By
IAM	(Not recommended) Use long-term credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use. • For the AWS CLI, see Authenticating using IAM user credentials in the *AWS Command Line Interface User Guide*. • For AWS SDKs and tools, see Authenticate using long-term credentials in the *AWS SDKs and Tools Reference Guide*. • For AWS APIs, see Managing access keys for IAM users in the *IAM User Guide*.

I'm an administrator and want to allow others to access Security Hub

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.

- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.

- IAM users:

- Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
- (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

I want to allow people outside my AWS account to access my Security Hub resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Security Hub supports these features, see How Security Hub works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Compliance validation for AWS Security Hub

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security Compliance & Governance – These solution implementation guides discuss architectural considerations and provide steps for deploying security and compliance features.
- HIPAA Eligible Services Reference – Lists HIPAA eligible services. Not all AWS services are HIPAA eligible.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in AWS Security Hub

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones

without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS Security Hub

As a managed service, AWS Security Hub is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Security Hub through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

AWS Security Hub and interface VPC endpoints (AWS PrivateLink)

You can establish a private connection between your VPC and AWS Security Hub by creating an *interface VPC endpoint*. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to privately access Security Hub APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Security Hub APIs. Traffic between your VPC and Security Hub does not leave the Amazon network.

Each interface endpoint is represented by one or more Elastic Network Interfaces in your subnets. For more information, see Access an AWS service using an interface VPC endpoint in the *Amazon Virtual Private Cloud Guide*.

Considerations for Security Hub VPC endpoints

Before you set up an interface VPC endpoint for Security Hub, ensure that you review the prerequisites and other information in the Amazon Virtual Private Cloud Guide.

Security Hub supports making calls to all of its API actions from your VPC.

Creating an interface VPC endpoint for Security Hub

You can create a VPC endpoint for the Security Hub service using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Create a VPC endpoint in the *Amazon Virtual Private Cloud Guide*.

Create a VPC endpoint for Security Hub using the following service name:

`com.amazonaws.region.securityhub`

Where *region* is the Region code for the applicable AWS Region.

If you enable private DNS for the endpoint, you can make API requests to Security Hub using its default DNS name for the Region, for example, `securityhub.us-east-1.amazonaws.com` for the US East (N. Virginia) Region.

Creating a VPC endpoint policy for Security Hub

You can attach an endpoint policy to your VPC endpoint that controls access to Security Hub. The policy specifies the following information:

- The principal that can perform actions.
- The actions that can be performed.
- The resources on which actions can be performed.

For more information, see Control access to VPC endpoints using endpoint policies in the *Amazon Virtual Private Cloud Guide*.

Example: VPC endpoint policy for Security Hub actions

The following is an example of an endpoint policy for Security Hub. When attached to an endpoint, this policy grants access to the listed Security Hub actions for all principals on all resources.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "securityhub:getFindings",
            "securityhub:getEnabledStandards",
            "securityhub:getInsights"
         ],
         "Resource":"*"
      }
   ]
}
```

Shared subnets

You can't create, describe, modify, or delete VPC endpoints in subnets that are shared with you. However, you can use the VPC endpoints in subnets that are shared with you. For information about VPC sharing, see Share your VPC subnets with other accounts in the *Amazon Virtual Private Cloud Guide*.

Logging Security Hub API calls with CloudTrail

AWS Security Hub CSPM is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Security Hub CSPM. CloudTrail captures API calls for Security Hub CSPM as events. The captured calls include calls from the Security Hub CSPM console and code calls to the Security Hub CSPM API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Security Hub CSPM. If you don't configure a trail, you can still view the most recent events on the CloudTrail console in **Event history**. Using the information that CloudTrail collects, you can determine the request that was made to Security Hub CSPM, the IP address that the request was made from, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

Security Hub CSPM information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in Security Hub CSPM, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your account. For more information, see Viewing events with CloudTrail event history.

For an ongoing record of events in your account, including events for Security Hub CSPM, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail on the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple regions and Receiving CloudTrail log files from multiple accounts

Security Hub CSPM supports logging all of the Security Hub CSPM API actions as events in CloudTrail logs. To view a list of Security Hub CSPM operations, see the Security Hub CSPM API Reference.

When activity for the following actions is logged to CloudTrail, the value for `responseElements` is set to `null`. This ensures that sensitive information isn't included in CloudTrail logs.

- `BatchImportFindings`
- `GetFindings`
- `GetInsights`
- `GetMembers`
- `UpdateFindings`

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials
- Whether the request was made with temporary security credentials for a role or federated user
- Whether the request was made by another AWS service

For more information, see the CloudTrail userIdentity element.

Example: Security Hub CSPM log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateInsight` action. In this example, an insight called `Test Insight` is created. The `ResourceId` attribute is specified as the **Group by** aggregator, and no optional filters for this insight are specified. For more information about insights, see Viewing insights in Security Hub CSPM.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAJK6U5DS22IAVUI7BW",
        "arn": "arn:aws:iam::012345678901:user/TestUser",
        "accountId": "012345678901",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "TestUser"
    },
    "eventTime": "2018-11-25T01:02:18Z",
    "eventSource": "securityhub.amazonaws.com",
    "eventName": "CreateInsight",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.251.233.179",
    "userAgent": "aws-cli/1.11.76 Python/2.7.10 Darwin/17.7.0 botocore/1.5.39",
    "requestParameters": {
        "Filters": {},
        "ResultField": "ResourceId",
        "Name": "Test Insight"
    },
    "responseElements": {
        "InsightArn": "arn:aws:securityhub:us-west-2:0123456789010:insight/custom/
f4c4890b-ac6b-4c26-95f9-e62cc46f3055"
    },
    "requestID": "c0fffccd-f04d-11e8-93fc-ddcd14710066",
    "eventID": "3dabcebf-35b0-443f-a1a2-26e186ce23bf",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "012345678901"
}
```

Tagging Security Hub resources

A *tag* is an optional label that you can define and assign to AWS resources, including certain types of AWS Security Hub CSPM resources. Tags can help you identify, categorize, and manage resources in different ways, such as by purpose, owner, environment, or other criteria. For example, you can use tags to distinguish between resources, identify resources that support certain compliance requirements or workflows, or allocate costs.

You can add tags to the following types of Security Hub CSPM resources:

- Automation rules
- Configuration policies
- Hub resource

Tagging fundamentals

A resource can have as many as 50 tags. Each tag consists of a required *tag key* and an optional *tag value*, both of which you define. A *tag key* is a general label that acts as a category for a more specific tag value. A *tag value* acts as a descriptor for a tag key.

For example, if you create different automation rules for different environments (one set of automation rules for test accounts and another for production accounts), you might assign an `Environment` tag key to those rules. The associated tag value might be `Test` for the rules that are associated with test accounts, and `Prod` for the rules that are associated with production accounts and OUs.

As you define and assign tags to AWS Security Hub CSPM resources, keep the following in mind:

- Each resource can have a maximum of 50 tags.
- For each resource, each tag key must be unique and it can have only one tag value.
- Tag keys and values are case sensitive. As a best practice, we recommend that you define a strategy for capitalizing tags and implement that strategy consistently across your resources.
- A tag key can have a maximum of 128 UTF-8 characters. A tag value can have a maximum of 256 UTF-8 characters. The characters can be letters, numbers, spaces, or the following symbols: _ . : / = + - @

- The `aws:` prefix is reserved for use by AWS. You can't use it in any tag keys or values that you define. In addition, you can't change or remove tag keys or values that use this prefix. Tags that use this prefix don't count against the quota of 50 tags per resource.
- Any tags that you assign are available only for your AWS account and only in the AWS Region in which you assign them.
- If you assign tags to a resource by using Security Hub CSPM, the tags are applied only to the resource that's stored directly in Security Hub CSPM in the applicable AWS Region. They aren't applied to any associated, supporting resources that Security Hub CSPM creates, uses, or maintains for you in other AWS services. For example, if you assign tags to an automation rule that updates findings related to Amazon Simple Storage Service (Amazon S3), the tags are applied only to your automation rule in Security Hub CSPM for the specified Region. They aren't applied to your S3 buckets. To also assign tags to an associated resource, you can use AWS Resource Groups or the AWS service that stores the resource—for example, Amazon S3 for an S3 bucket. Assigning tags to associated resources can help you identify supporting resources for your Security Hub CSPM resources.
- If you delete a resource, any tags that are assigned to the resource are also deleted.

> **Important**
>
> Do not store confidential or other types of sensitive data in tags. Tags are accessible from many AWS services, including AWS Billing and Cost Management. They aren't intended to be used for sensitive data.

To add and manage tags for Security Hub CSPM resources, you can use the Security Hub CSPM console, the Security Hub CSPM API, or the AWS Resource Groups Tagging API. With Security Hub CSPM, you can add tags to a resource when you create the resource. You can also add and manage tags for individual existing resources. With Resource Groups, you can add and manage tags in bulk for multiple existing resources spanning multiple AWS services, including Security Hub CSPM.

For additional tagging tips and best practices, see Tagging your AWS resources in the *Tagging AWS Resources User Guide*.

Using tags in IAM policies

After you start tagging resources, you can define tag-based, resource-level permissions in AWS Identity and Access Management (IAM) policies. By using tags in this way, you can implement granular control of which users and roles in your AWS account have permission to create and tag resources, and which users and roles have permission to add, edit, and remove tags more generally. To control access based on tags, you can use tag-related condition keys in the Condition element of IAM policies.

For example, you can create an IAM policy that allows a user to have full access to all AWS Security Hub CSPM resources, if the `Owner` tag for the resource specifies their username:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ModifyResourceIfOwner",
            "Effect": "Allow",
            "Action": "securityhub:*",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"aws:ResourceTag/Owner":
 "${aws:username}"}
            }
        }
    ]
}
```

If you define tag-based, resource-level permissions, the permissions take effect immediately. This means that your resources are more secure as soon as they're created, and you can quickly start enforcing the use of tags for new resources. You can also use resource-level permissions to control which tag keys and values can be associated with new and existing resources. For more information, see Controlling access to AWS resources using tags in the *IAM User Guide*.

Adding tags to Security Hub CSPM resources

A *tag* is a label that you can define and assign to AWS resources, including certain types of AWS Security Hub CSPM resources. By using tags, you can identify, categorize, and manage resources in different ways, such as by purpose, owner, environment, or other criteria. For example, you can use tags to: apply policies, allocate costs, distinguish between versions of resources, or identify resources that support certain compliance requirements or workflows.

You can add tags to the following types of Security Hub CSPM resources:

- Automation rules
- Configuration policies
- Hub resource

A resource can have as many as 50 tags. Each tag consists of a required *tag key* and an optional *tag value*. A *tag key* is a general label that acts as a category for a more specific tag value. A *tag value* acts as a descriptor for a tag key. For more information about tagging options and requirements, see Tagging fundamentals.

To add tags to a Security Hub CSPM resource, you can use the Security Hub CSPM console or the Security Hub CSPM API. However, the console doesn't support adding tags to the Hub resource.

After adding tags, you can edit the tag and change the tag key or tag value.

To add or edit tags for multiple Security Hub CSPM resources at the same time, use the tagging operations of the AWS Resource Groups Tagging API.

Important

Adding tags to a resource can affect access to the resource. Before you add a tag to a resource, review any AWS Identity and Access Management (IAM) policies that might use tags to control access to resources.

Console

To add tags to a Security Hub CSPM resource (console)

When you create an automation rule or a configuration policy, the Security Hub CSPM console provides options for adding tags to it. You can provide the tag key and tag value in the **Tags** section.

Security Hub CSPM API

To add tags to a Security Hub CSPM resource (API)

To create a resource and add one or more tags to it programmatically, use the appropriate operation for the type of resource that you want to create:

- To create a configuration policy and add one or more tags to it, invoke the CreateConfigurationPolicy API or, if you're using the AWS CLI, run the create-configuration-policy command.
- To create an automation rule and add one or more tags to it, invoke the CreateAutomationRule API or, if you're using the AWS CLI, run the create-automation-rule command.
- To enable Security Hub CSPM and add one or more tags to your Hub resource, invoke the EnableSecurityHub API or, if you're using the AWS Command Line Interface (AWS CLI), run the enable-security-hub command.

In your request, use the `tags` parameter to specify the tag key and optional tag value for each tag to add to the resource. The `tags` parameter specifies an array of objects. Each object specifies a tag key and its associated tag value.

To add one or more tags to an existing resource, use the TagResource operation of the Security Hub CSPM API or, if you're using the AWS CLI, run the tag-resource command. In your request, specify the Amazon Resource Name (ARN) of the resource that you want to add a tag to. Use the `tags` parameter to specify the tag key (`key`) and optional tag value (`value`) for each tag to add. The `tags` parameter specifies an array of objects, one object for each tag key and its associated tag value.

For example, the following AWS CLI command adds an `Environment` tag key with a `Prod` tag value to the specified configuration policy. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

Example CLI command:

```
$ aws securityhub tag-resource \
```

```
--resource-arn arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
--tags '{"Environment":"Prod"}'
```

Where:

- `resource-arn` specifies the ARN of the configuration policy to add a tag to.
- *`Environment`* is the tag key of the tag to add to the rule.
- *`Prod`* is the tag value for the specified tag key (*`Environment`*).

In the following example, the command adds several tags to the configuration policy.

```
$ aws securityhub tag-resource \
--resource-arn arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
--tags '{"Environment":"Prod", "CostCenter":"12345", "Owner":"jane-doe"}'
```

For each object in a `tags` array, both the `key` and `value` arguments are required. However, the value for the `value` argument can be an empty string. If you don't want to associate a tag value with a tag key, don't specify a value for the `value` argument. For example, the following command adds an `Owner` tag key with no associated tag value:

```
$ aws securityhub tag-resource \
--resource-arn arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
--tags '{"Owner":""}'
```

If a tagging operation succeeds, Security Hub CSPM returns an empty HTTP 200 response. Otherwise, Security Hub CSPM returns an HTTP 4*xx* or 500 response that indicates why the operation failed.

Editing tags for Security Hub CSPM resources

As your environment or requirements change over time, you can evaluate existing tags for your AWS Security Hub CSPM resources and change the tags as necessary. A *tag* is a label that you define and assign to one or more AWS resources, including certain types of Macie resources. Each tag consists of a required *tag key* and an optional *tag value*. A *tag key* is a general label that acts as a category for a more specific tag value. A *tag value* acts as a descriptor for a tag key.

Tags can help you identify, categorize, and manage resources in different ways, such as by purpose, owner, environment, or other criteria. For example, you can use tags to: apply policies, allocate costs, distinguish between versions of resources, or identify resources that support certain compliance requirements or workflows.

You can add tags to the following types of Security Hub CSPM resources:

- Automation rules
- Configuration policies
- Hub resource

To edit tag keys or tag values for a Security Hub CSPM resource, you can use the Security Hub CSPM API. The Security Hub CSPM console currently doesn't support tag editing.

> **Important**
>
> Editing tags for a resource can affect access to the resource. Before you edit a tag for a resource, review any AWS Identity and Access Management (IAM) policies that might use tags to control access to resources.

Security Hub CSPM API

To edit tags for a Security Hub CSPM resource (API)

When you edit a tag for a resource programmatically, you overwrite the existing tag with new values. Therefore, the best way to edit a tag depends on whether you want to edit a tag key, a tag value, or both. To edit a tag key, remove the current tag and add a new tag.

To edit or remove only the tag value that's associated with a tag key, overwrite the existing value by using the TagResource operation of the Security Hub CSPM API. If you're using the AWS CLI, run the tag-resource command. In your request, specify the Amazon Resource Name (ARN) of the resource whose tag value you want to edit or remove.

To edit a tag value, use the `tags` parameter to specify the tag key whose tag value you want to change. You should also specify the new tag value for the key. For example, the following AWS CLI command changes the tag value from `Prod` to `Test` for the `Environment` tag key that's assigned to the specified automation rule. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securityhub tag-resource \
--resource-arn arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
--tags '{"Environment":"Test"}'
```

Where:

- `resource-arn` specifies the ARN of the configuration policy.
- *`Environment`* is the tag key that's associated with the tag value to change.
- *`Test`* is the new tag value for the specified tag key (*`Environment`*).

To remove a tag value from a tag key, don't specify a value for the `value` argument of the key in the `tags` parameter. For example:

```
$ aws securityhub tag-resource \
--resource-arn arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
--tags '{"Owner":""}'
```

If the operation succeeds, Security Hub CSPM returns an empty HTTP 200 response. Otherwise, Security Hub CSPM returns an HTTP 4*xx* or 500 response that indicates why the operation failed.

Reviewing tags for Security Hub CSPM resources

After you add or edit tags for AWS Security Hub CSPM resources, you can view what tag keys and tag values a resource currently has. A *tag* is a label that you define and assign to one or more AWS resources, including certain types of Macie resources. Each tag consists of a required *tag key* and an optional *tag value*. A *tag key* is a general label that acts as a category for a more specific tag value. A *tag value* acts as a descriptor for a tag key.

Tags can help you identify, categorize, and manage resources in different ways, such as by purpose, owner, environment, or other criteria. For example, you can use tags to: apply policies, allocate costs, distinguish between versions of resources, or identify resources that support certain compliance requirements or workflows.

You can add tags to the following types of Security Hub CSPM resources:

- Automation rules
- Configuration policies
- Hub resource

You can review the tags for a Security Hub CSPM automation rule or configuration policy by using the Security Hub CSPM console or the Security Hub CSPM API. The console doesn't support reviewing tags for the Hub resource. Programmatically, you can review tags for any resource.

To review tags for multiple Security Hub CSPM resources at the same time, use the tagging operations of the AWS Resource Groups Tagging API.

Console

To review tags for a Security Hub CSPM resource (console)

1. Using the credentials of the Security Hub CSPM administrator, open the AWS Security Hub CSPM console at https://console.aws.amazon.com/securityhub/.
2. Depending on the type of resource that you want to add a tag to, do one of the following:
   - To review the tags for an automation rule, choose **Automations** in the navigation pane. Then, choose an automation rule.
   - To review the tags for a configuration policy, choose **Configuration** in the navigation pane. Then, on the **Policies** tab, select the option next to a configuration policy. A side panel opens that shows you the number of tags assigned to the policy. You can expand the **Tags** header to see the tag keys and tag values.

   The **Tags** section lists all the tags that are currently assigned to the resource.

Security Hub CSPM API

To review tags for a Security Hub CSPM resource (API)

To retrieve and review the tags for an existing resource, invoke the ListTagsForResource API. In your request, use the `resourceArn` parameter to specify the Amazon Resource Name (ARN) of the resource.

If you're using the AWS CLI, run the list-tags-for-resource command and use the `resource-arn` parameter to specify the ARN of the resource. For example:

```
$ aws securityhub list-tags-for-resource --resource-arn arn:aws:securityhub:us-east-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

If the operation succeeds, Security Hub CSPM returns a `tags` array. Each object in the array specifies a tag (both the tag key and tag value) that's currently assigned to the resource. For example:

```
{
    "tags": [
        {
            "key": "Environment",
            "value": "Prod"
        },
        {
            "key": "CostCenter",
            "value": "12345"
        },
        {
            "key": "Owner",
            "value": ""
        }
    ]
}
```

Where `Environment`, `CostCenter`, and `Owner` are the tag keys that are assigned to the resource. `Prod` is the tag value that's associated with the `Environment` tag key. `12345` is the tag value that's associated with the `CostCenter` tag key. The `Owner` tag key doesn't have an associated tag value.

To retrieve a list of all the Security Hub CSPM resources that have tags and all the tags that are assigned to each of those resources, use the GetResources operation of the AWS Resource Groups Tagging API. In your request, set the value for the `ResourceTypeFilters` parameter to `securityhub`. To do this using the AWS CLI, run the get-resources command and set the value for the `resource-type-filters` parameter to `securityhub`. For example:

```
$ aws resourcegroupstaggingapi get-resources -\-resource-type-filters "securityhub"
```

If the operation succeeds, Resource Groups returns a `ResourceTagMappingList` array. The array contains one object for each Security Hub CSPM resource that has tags. Each object

specifies the ARN of a Security Hub CSPM resource, and the tag keys and values that are assigned to the resource.

Removing tags from Security Hub CSPM resources

If you add tags to an AWS Security Hub CSPM resource, you can subsequently remove one or more of them. A *tag* is a label that you define and assign to AWS resources, including certain types of Security Hub CSPM resources. You can add, edit, and remove tags from the following types of Security Hub CSPM resources: automation rules, configuration policies, and the Hub resource.

To remove tags from an individual AWS Security Hub CSPM resource, you can use the Security Hub CSPM API. The Security Hub CSPM console currently doesn't support tag removal.

To remove tags from multiple Security Hub CSPM resources at the same time, use the tagging operations of the AWS Resource Groups Tagging API.

> **Important**
>
> Removing tags from a resource can affect access to the resource. Before you remove a tag, review any AWS Identity and Access Management (IAM) policies that might use the tag to control access to resources.

Security Hub CSPM API

To remove tags from a Security Hub CSPM resource (API)

To remove one or more tags from a resource programmatically, use the UntagResource operation of the Security Hub CSPM API. In your request, use the `resourceArn` parameter to specify the Amazon Resource Name (ARN) of the resource to remove a tag from. Use the `tagKeys` parameter to specify the tag key of the tag to remove. To remove multiple tags, append the `tagKeys` parameter and argument for each tag to remove, separated by an ampersand (&)—for example, `tagKeys=key1&tagKeys=key2`. To remove only a specific tag value (not a tag key) from a resource, edit the tag instead of removing the tag.

If you're using the AWS CLI, run the untag-resource command to remove one or more tags from a resource. For the `resource-arn` parameter, specify the ARN of the resource to remove a tag from. Use the `tag-keys` parameter to specify the tag key of the tag to remove. For example,

the following command removes the `Environment` tag (both the tag key and tag value) from the specified configuration policy:

```
$ aws securityhub untag-resource \
--resource-arn arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
--tag-keys Environment
```

Where `resource-arn` specifies the ARN of the configuration policy to remove a tag from, and *`Environment`* is the tag key of the tag to remove.

To remove multiple tags from a resource, add each additional tag key as an argument for the `tag-keys` parameter. For example:

```
$ aws securityhub untag-resource \
--resource-arn arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
--tag-keys Environment Owner
```

If the operation succeeds, Security Hub CSPM returns an empty HTTP 200 response. Otherwise, Security Hub CSPM returns an HTTP 4*xx* or 500 response that indicates why the operation failed.

Quotas for Security Hub

Your AWS account has certain default quotas, formerly referred to as *limits*, for each AWS service. These quotas are the maximum number of service resources or operations for your account. This topic links to the quotas that apply to AWS Security Hub resources and operations for your account. Unless otherwise noted, each quota applies to your account in each AWS Region.

Some quotas can be increased, while others cannot. To request an increase to a quota, use the Service Quotas console. To learn how to request an increase, see Requesting a quota increase in the *Service Quotas User Guide*. If a quota isn't available on the Service Quotas console, use the service limit increase form on the AWS Support Center Console to request an increase to the quota.

Maximum quotas

For a list of quotas that apply to AWS Security Hub resources, see AWS Security Hub endpoints and quotas in the *AWS General Reference*.

Rate quotas

For a list of quotas that apply to AWS Security Hub API operations, see the AWS Security Hub API Reference.

If you set up cross-Region aggregation in Security Hub CSPM, one call to `BatchImportFindings` and `BatchUpdateFindings` impacts linked Regions and the aggregation Region. The `GetFindings` operation retrieves findings from linked Regions and the aggregation Region. However, the `BatchEnableStandards` and `UpdateStandardsControl` operations are Region-specific.

Document history for the AWS Security Hub User Guide

The following table describes the important changes to the documentation since the last release of AWS Security Hub and Security Hub CSPM. For releases of new Security Hub CSPM controls, the date specifies when the controls begin to be available in supported AWS Regions. It can take 1-2 weeks for controls to be available in all supported Regions. For details about material changes to existing controls, see Change log for Security Hub CSPM controls.

To receive notifications about updates to the *AWS Security Hub User Guide*, you can subscribe to an RSS feed.

Change	Description	Date
New security controls	Security Hub CSPM released two new controls for the AWS Foundational Security Best Practices (FSBP) standard: CloudFront.16 and RDS.46.	September 3, 2025
Regional availability	The AWS Resource Tagging standard is now available in the Asia Pacific (Thailand) and Mexico (Central) Regions.	August 29, 2025
Updates to security standards and controls	Due to Amazon Redshift limitations, we are planning to retire the Redshift.9 and RedshiftServerless.7 controls and remove them from all applicable standards on September 15, 2025. Currently, these controls apply to the AWS Foundational Security Best Practices (FSBP) standard and the NIST SP 800-53 Rev. 5 standard. The Redshift.9 control also	August 15, 2025

	applies to the AWS Control Tower service-managed standard.	
New resource details object in the ASFF	The AWS Security Finding Format (ASFF) now includes a `CodeRepository` resource object. This object provides details about an external code repository that you connected to AWS resources and configured Amazon Inspector to scan for vulnerabilities.	August 1, 2025
Regional availability	Security Hub CSPM is now available in the Asia Pacific (Taipei) Region. For a complete list of AWS Regions where Security Hub CSPM is currently available, see AWS Security Hub endpoints and quotas in the *AWS General Reference*.	July 23, 2025
New third-party integration	Dynatrace is a new third-party integration that can receive findings from Security Hub CSPM.	July 18, 2025

New security controls	Security Hub CSPM released 13 new controls. Most of the controls support the AWS Foundational Security Best Practices (FSBP) standard. Some of the controls support NIST SP 800-53 Rev. 5 requirements. • For the AWS FSBP standard, IDs for the applicable controls are: CloudFront.15, Cognito.2, EC2.180, ELB.18, MSK.4, MSK.5, MSK.6, RDS.45, Redshift.18, S3.25, SSM.6, and SSM.7. • For NIST SP 800-53 Rev. 5, IDs for the applicable controls are: Lambda.7 and RDS.45.	July 15, 2025
Updates to generation of control findings	To help you track compliance changes, Security Hub CSPM now updates existing control findings, instead of generating new findings, when there are changes to the compliance status of individual resources. This means that you can use the data provided by individual findings to track compliance changes for particular resources against particular controls.	July 3, 2025

Updates to security standards and controls	We removed the IAM.13 control from the PCI DSS v4.0.1 standard. We also removed the IAM.17 control from the NIST SP 800-171 Revision 2 standard. The standards don't explicitly require the checks that these controls provide. We also updated related requireme nt details for these standards for certain controls that check IAM password policies: IAM.7, and IAM.10 through IAM.17.	June 30, 2025
Updates to finding retention	Security Hub CSPM now stores archived findings for 30 days instead of 90 days, which can reduce finding noise. For longer-term retention, you can export findings to an S3 bucket by using a custom action with an Amazon EventBridge rule.	June 20, 2025

Updates to existing managed policies	Security Hub CSPM added new permission to the AWS managed policy named `AWSSecurityHubOrganizationsAccess`. The permission allows the organization management to enable and manage Security Hub and Security Hub CSPM within an organization. Security Hub CSPM also added new permission to the AWS managed policy named `AWSSecurityHubFullAccess`. The permission allows principals to create a service-linked role for Security Hub.	June 18, 2025
Public preview release and a new managed policy for Security Hub	Public preview release of AWS Security Hub and the *AWS Security Hub User Guide*. This release includes a new AWS managed policy, `AWSSecurityHubV2ServiceRolePolicy`. The policy allows Security Hub to manage AWS Config rules and Security Hub resources in a customer's organization and on the customer's behalf. Security Hub is in preview release and subject to change.	June 17, 2025

Updates to security standards and controls	We removed the IAM.10 control from the PCI DSS v4.0.1 standard. This control checks whether account password policies for IAM users meet minimum requirements, including a minimum password length of 7 characters. PCI DSS v4.0.1 now requires passwords to have a minimum of 8 characters. The IAM.10 control continues to apply to the PCI DSS v3.2.1 standard, which has different password requirements.	May 30, 2025
New security standard	Security Hub CSPM now provides a security standard that aligns with the NIST SP 800-171 Revision 2 cybersecurity and compliance framework. This new standard includes more than 60 existing security controls. The controls perform automated checks that evaluate certain AWS services and resources for compliance with a subset of the requirements defined by the framework.	May 29, 2025

Updates to security controls	Security Hub CSPM rolled back the release of the following control in all AWS Regions: *[RDS.46] RDS DB instances should not be deployed in public subnets with routes to internet gateways*. Previously, this control supported the AWS Foundational Security Best Practices (FSBP) standard.	May 8, 2025
New security controls	Security Hub CSPM released 9 new controls. Most of the controls support the AWS Foundational Security Best Practices (FSBP) standard. Some of the controls support NIST SP 800-53 Rev. 5 requirements. • For the AWS FSBP standard, IDs for the applicable controls are: DocumentDB.6, RDS.44, RedshiftServerless.2, RedshiftServerless.3, RedshiftServerless.5, RedshiftServerless.6, and RedshiftServerless.7. • For NIST SP 800-53 Rev. 5, IDs for the applicable controls are: CloudTrail.10, RedshiftServerless.4, and RedshiftServerless.7.	May 7, 2025

New security controls	Security Hub CSPM released 24 new controls. Most of the controls support the AWS Foundational Security Best Practices (FSBP) or AWS Resource Tagging standard. Some of the controls support NIST SP 800-53 Rev. 5 requirements. • For the AWS FSBP standard, IDs for the applicable controls are: EC2.173, RDS.41, RDS.42, and SageMaker.8. • For the AWS Resource Tagging standard, IDs for the applicable controls are: Amplify.1, Amplify.2 , Batch.4, DataSync.2, EC2.174, EC2.175, EC2.176, EC2.177, EC2.178, EC2.179, Redshift.17, SageMaker .6, SageMaker.7, SSM.5, Transfer.4, Transfer.5, Transfer.6, and Transfer.7. • For NIST SP 800-53 Rev. 5, IDs for the applicable controls are: ECS.17 and RDS.42.	April 16, 2025

New security controls	Security Hub CSPM released four new controls for the AWS Foundational Security Best Practices standard. The controls are: • the section called "[FSx.3] FSx for OpenZFS file systems should be configured for Multi-AZ deployment" • the section called "[FSx.4] FSx for NetApp ONTAP file systems should be configured for Multi-AZ deployment" • the section called "[FSx.5] FSx for Windows File Server file systems should be configured for Multi-AZ deployment" • the section called "[RedshiftServerless.1] Amazon Redshift Serverless workgroups should use enhanced VPC routing"	March 18, 2025

Updates to security standards and controls	We removed the RDS.18 security control from the AWS Foundational Security Best Practices standard and automated checks for NIST SP 800-53 Rev. 5 requirements. Since Amazon EC2-Classic networking was retired, Amazon Relational Database Service (Amazon RDS) instances can no longer be deployed outside a VPC. The control continues to be part of the AWS Control Tower service-managed standard.	March 7, 2025
Updates to control findings	Security Hub CSPM now generates `WARNING` findings for an enabled control if resource recording isn't turned on in AWS Config for the type of resource that the control checks. This can help you identify and address potential configuration gaps in your security control checks.	February 25, 2025

New security controls	Security Hub CSPM released 11 new controls. The controls are: • the section called "[Connect .2] Amazon Connect instances should have CloudWatch logging enabled" • the section called "[ECR.5] ECR repositories should be encrypted with customer managed AWS KMS keys" • the section called "[ELB.17] Application and Network Load Balancers with listeners should use recommended security policies" • the section called "[Glue.4] AWS Glue Spark jobs should run on supported versions of AWS Glue" • the section called "[GuardDuty.11] GuardDuty Runtime Monitoring should be enabled" • the section called "[GuardDuty.12] GuardDuty ECS Runtime Monitoring should be enabled" • the section called "[GuardDuty.13] GuardDuty EC2 Runtime Monitoring should be enabled"	February 24, 2025

- the section called "[NetworkFirewall.10] Network Firewall firewalls should have subnet change protection enabled"
- the section called "[RDS.40] RDS for SQL Server DB instances should publish logs to CloudWatch Logs"
- the section called "[SQS.3] SQS queue access policies should not allow public access"
- the section called "[Transfer.3] Transfer Family connectors should have logging enabled"

New security controls	Security Hub CSPM released 37 new controls for the AWS Resource Tagging Standard. Security Hub CSPM also released the following new controls: • the section called "[EMR.3] Amazon EMR security configurations should be encrypted at rest" • the section called "[EMR.4] Amazon EMR security configurations should be encrypted in transit" • the section called "[SageMaker.5] SageMaker models should have network isolation enabled"	January 22, 2025
New security control	Security Hub CSPM released EC2.172 EC2 VPC Block Public Access settings should block internet gateway traffic.	January 15, 2025

New security controls	The following new Security Hub CSPM controls are available. • the section called "[Cognito.1] Cognito user pools should have threat protection activated with full function enforcement mode for standard authentication" • the section called "[RDS.38] RDS for PostgreSQL DB instances should be encrypted in transit" • the section called "[RDS.39] RDS for MySQL DB instances should be encrypted in transit" • the section called "[Redshift.16] Redshift cluster subnet groups should have subnets from multiple Availability Zones"	December 17, 2024
Security Hub CSPM supports PCI DSS v4.0.1	Security Hub CSPM now supports v4.0.1 of the Payment Card Industry Data Security Standard (PCI DSS). For more information about the standard and the controls that apply to it, see PCI DSS in Security Hub CSPM.	December 11, 2024

Security Hub CSPM receives GuardDuty attack sequence findings	Security Hub CSPM now receives attack sequence findings from Amazon GuardDuty Extended Threat Detection. Attack sequence finding details are available in the Detection object of the AWS Security Finding Format (ASFF).	December 1, 2024
Security Hub CSPM supported in new AWS Region	Security Hub CSPM is now available in the Asia Pacific (Malaysia) Region. Some security controls have Regional limitations. For a list of controls that aren't available in this Region, see Regional limits on Security Hub CSPM controls.	November 22, 2024
Changes to Config.1	Security Hub CSPM increased the severity of the Config.1 control from `MEDIUM` to `CRITICAL`, and added new status codes and status reasons for failed Config.1 findings. For more informati on about the changes, see the entry for November 20, 2024 in the Change log for Security Hub CSPM controls.	November 20, 2024

New security controls	The following new Security Hub CSPM controls are available. These controls are part of AWS Foundational Security Best Practices and NIST SP 800-53 Rev. 5, and they evaluate whether a virtual private cloud (VPC) that you manage has an interface VPC endpoint for an AWS service or AWS resource. • the section called "[EC2.55] VPCs should be configured with an interface endpoint for ECR API" • the section called "[EC2.56] VPCs should be configured with an interface endpoint for Docker Registry" • the section called "[EC2.57] VPCs should be configured with an interface endpoint for Systems Manager" • the section called "[EC2.58] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager Contacts" • the section called "[EC2.60] VPCs should be configured with an interface endpoint for Systems Manager Incident Manager"	November 15, 2024

New security controls	The following new Security Hub CSPM controls are available. • the section called "[AppSync.1] AWS AppSync API caches should be encrypted at rest" • the section called "[AppSync.6] AWS AppSync API caches should be encrypted in transit" • the section called "[EC2.170] EC2 launch templates should use Instance Metadata Service Version 2 (IMDSv2)" • the section called "[EC2.171] EC2 VPN connections should have logging enabled" • the section called "[EFS.8] EFS file systems should be encrypted at rest" • the section called "[KMS.5] KMS keys should not be publicly accessible" • the section called "[SNS.4] SNS topic access policies should not allow public access"	October 18, 2024

New security controls	The following new Security Hub CSPM controls are available. • the section called "[ECS.16] ECS task sets should not automatically assign public IP addresses" • the section called "[GuardDuty.7] GuardDuty EKS Runtime Monitoring should be enabled" • the section called "[Kinesis .3] Kinesis streams should have an adequate data retention period" • the section called "[MSK.3] MSK Connect connector s should be encrypted in transit" • the section called "[RDS.36] RDS for PostgreSQL DB instances should publish logs to CloudWatch Logs" • the section called "[RDS.37] Aurora PostgreSQL DB clusters should publish logs to CloudWatch Logs" • the section called "[S3.24] S3 Multi-Region Access Points should have block public access settings enabled"	October 3, 2024

New security controls	The following new Security Hub CSPM controls are available. • the section called "[Athena.4] Athena workgroups should have logging enabled" • the section called "[CodeBuild.7] CodeBuild report group exports should be encrypted at rest" • the section called "[DataSync.1] DataSync tasks should have logging enabled" • the section called "[EFS.7] EFS file systems should have automatic backups enabled" • Glue.2 (retired) • the section called "[Glue.3] AWS Glue machine learning transforms should be encrypted at rest" • the section called "[WorkSpaces.1] WorkSpaces user volumes should be encrypted at rest" • the section called "[WorkSpaces.2] WorkSpaces root volumes should be encrypted at rest"	August 30, 2024

New finding panel	The new finding panel on the Security Hub CSPM console helps you quickly take action on findings, review resource details and finding history, and find other pertinent information about a finding.	August 16, 2024
Update to Config.1 control	The Config.1 control checks whether AWS Config is enabled, uses the service-linked role, and records resources for enabled controls. Security Hub CSPM added a custom control parameter named `includeConfigServiceLinkedRoleCheck`. By setting this parameter to `false`, you can opt out of checking whether AWS Config uses the service-linked role.	August 15, 2024
Designate a home Region without linked Regions	You can now create a finding aggregator and establish a home Region without linking any AWS Regions to the home Region. This allows you to enable central configuration without specifying linked Regions.	July 25, 2024

Select controls available in more Regions	The following controls are now available in additional AWS Regions, including US East (N. Virginia) and US East (Ohio). • the section called "[DataFirehose.1] Firehose delivery streams should be encrypted at rest" • the section called "[DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled" • the section called "[DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled" • the section called "[DMS.12] DMS endpoints for Redis OSS should have TLS enabled" • the section called "[DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit" • the section called "[EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch" • the section called "[EKS.3] EKS clusters should use	July 15, 2024

encrypted Kubernetes secrets”

- the section called “[FSx.2] FSx for Lustre file systems should be configured to copy tags to backups”
- the section called “[MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch”
- the section called “[MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled”
- the section called “[Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes”
- the section called “[Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins”
- the section called “[SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1”
- the section called “[Service Catalog.1] Service Catalog portfolios should be shared within an AWS organization only”

- the section called "[Transfer.2] Transfer Family servers should not use FTP protocol for endpoint connection"

New security controls	The following new Security Hub CSPM controls are available: • the section called "[GuardDuty.5] GuardDuty EKS Audit Log Monitoring should be enabled" • the section called "[GuardDuty.6] GuardDuty Lambda Protection should be enabled" • the section called "[GuardDuty.8] GuardDuty Malware Protection for EC2 should be enabled" • the section called "[GuardDuty.9] GuardDuty RDS Protection should be enabled" • the section called "[GuardDuty.10] GuardDuty S3 Protection should be enabled" • the section called "[Inspector.1] Amazon Inspector EC2 scanning should be enabled" • the section called "[Inspector.2] Amazon Inspector ECR scanning should be enabled" • the section called "[Inspector.3] Amazon Inspector	July 11, 2024

Lambda code scanning should be enabled”

- the section called “[Inspector.4] Amazon Inspector Lambda standard scanning should be enabled”

Release of CIS AWS Foundations Benchmark v3.0.0	Security Hub CSPM released Center for Internet Security (CIS) AWS Foundations Benchmark v3.0.0. The release includes the following new controls, as well as mappings to several existing controls. • the section called "[EC2.53] EC2 security groups should not allow ingress from 0.0.0.0/0 to remote server administration ports" • the section called "[EC2.54] EC2 security groups should not allow ingress from ::/0 to remote server administration ports" • the section called "[IAM.26] Expired SSL/TLS certificates managed in IAM should be removed" • the section called "[IAM.27] IAM identities should not have the AWSCloudShellFullAccess policy attached" • the section called "[IAM.28] IAM Access Analyzer external access analyzer should be enabled" • the section called "[S3.22] S3 general purpose buckets	May 13, 2024

should log object-level write events”

- the section called “[S3.23] S3 general purpose buckets should log object-level read events”

New security controls	The following new Security Hub CSPM controls are available: • the section called "[DataFirehose.1] Firehose delivery streams should be encrypted at rest" • the section called "[DMS.10] DMS endpoints for Neptune databases should have IAM authorization enabled" • the section called "[DMS.11] DMS endpoints for MongoDB should have an authentication mechanism enabled" • the section called "[DMS.12] DMS endpoints for Redis OSS should have TLS enabled" • the section called "[DynamoDB.7] DynamoDB Accelerator clusters should be encrypted in transit" • the section called "[EFS.6] EFS mount targets should not be associated with subnets that assign public IP addresses on launch" • the section called "[EKS.3] EKS clusters should use encrypted Kubernetes secrets"	May 3, 2024

- the section called "[FSx.2] FSx for Lustre file systems should be configured to copy tags to backups"
- the section called "[MQ.2] ActiveMQ brokers should stream audit logs to CloudWatch"
- the section called "[MQ.3] Amazon MQ brokers should have automatic minor version upgrade enabled"
- the section called "[Opensearch.11] OpenSearch domains should have at least three dedicated primary nodes"
- the section called "[Redshift.15] Redshift security groups should allow ingress on the cluster port only from restricted origins"
- the section called "[SageMaker.4] SageMaker endpoint production variants should have an initial instance count greater than 1"
- the section called "[Service Catalog.1] Service Catalog portfolios should be shared within an AWS organization only"
- the section called "[Transfer.2] Transfer Family servers

	should not use FTP protocol for endpoint connection”	
AWS Resource Tagging Standard	The AWS Resource Tagging Standard from Security Hub CSPM is now generally available, along with new controls that apply to the standard.	April 30, 2024
Update to existing managed policy	Security Hub CSPM updated the AWS managed policy named `AmazonSecurityHubFullAccess` to get pricing details for AWS services and products.	April 24, 2024
In-context configuration of control parameters	If you use central configuration, you can now configure control parameters in context, from the details page of a control on the Security Hub CSPM console.	March 29, 2024
Update to existing managed policy	Security Hub CSPM updated the AWS managed policy named `AWSSecurityHubReadOnlyAccess` by adding a `Sid` field.	February 22, 2024
New security control	The control [Macie.2] Macie automated sensitive data discovery should be enabled is now available. For Regional limits on this control, see Availability of controls by Region.	February 19, 2024

Security Hub CSPM available in Canada West (Calgary)	Security Hub CSPM is now available in Canada West (Calgary). All Security Hub CSPM features are now available in this Region, with the exception of certain security controls. For more information, see Availability of controls by Region.	December 20, 2023

New security controls	The following new Security Hub CSPM controls are available: • the section called "[Backup.1] AWS Backup recovery points should be encrypted at rest" • the section called "[DynamoDB.6] DynamoDB tables should have deletion protection enabled" • the section called "[EC2.51] EC2 Client VPN endpoints should have client connection logging enabled" • the section called "[EKS.8] EKS clusters should have audit logging enabled" • the section called "[EMR.2] Amazon EMR block public access setting should be enabled" • the section called "[FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes" • the section called "[Macie.1] Amazon Macie should be enabled" • the section called "[MSK.2] MSK clusters should have	December 14, 2023

enhanced monitoring configured”

- the section called “[Neptune.9] Neptune DB clusters should be deployed across multiple Availability Zones”
- the section called “[NetworkFirewall.1] Network Firewall firewalls should be deployed across multiple Availability Zones”
- the section called “[NetworkFirewall.2] Network Firewall logging should be enabled”
- the section called “[Opensearch.10] OpenSearch domains should have the latest software update installed”
- the section called “[PCA.1] AWS Private CA root certificate authority should be disabled”
- the section called “[S3.19] S3 access points should have block public access settings enabled”
- the section called “[S3.20] S3 general purpose buckets should have MFA delete enabled”

Finding enrichment	Security Hub CSPM added the new finding fields `AwsAccountName` , `ApplicationArn` , and `ApplicationName` to the AWS Security Finding Format (ASFF).	November 27, 2023
Enhancements to Summary dashboard	You can now access more dashboard widgets on the **Summary** page of the Security Hub CSPM console, save dashboard filter sets to quickly focus on specific security issues, and customize the dashboard layout.	November 27, 2023
Central configuration	Central configuration is now available. With central configuration, the Security Hub CSPM delegated administrator can configure Security Hub CSPM, standards , and controls across multiple organization accounts, organizational units (OUs), and Regions.	November 27, 2023
Updates to managed policy	Security Hub CSPM added new permissions to the `AWSSecurityHubServiceRolePolicy` managed policy that allow Security Hub CSPM to read and update customizable security control properties.	November 26, 2023

Custom control parameters	You can now customize parameter values for select Security Hub CSPM controls. This can make findings for a specific control more relevant to your business requirements and security expectations.	November 26, 2023
Updates to managed policies	Security Hub CSPM updated the `AWSSecurityHubFullAccess` and `AWSSecurityHubOrganizationsAccess` managed policies that permit you to use, respectively, Security Hub CSPM features and the integration with AWS Organizations.	November 16, 2023

Existing security controls added to Service-Managed Standard: AWS Control Tower	The following existing Security Hub CSPM controls have been added to Service-Managed Standard: AWS Control Tower. • **ACM.2** • **AppSync.5** • **CloudTrail.6** • **DMS.9** • **DocumentDB.3** • **DynamoDB.3** • **EC2.23** • **EKS.1** • **ElastiCache.3** • **ElastiCache.4** • **ElastiCache.5** • **ElastiCache.6** • **EventBridge.3** • **KMS.4** • **Lambda.3** • **MQ.5** • **MQ.6** • **MSK.1** • **RDS.12** • **RDS.15** • **S3.17**	November 14, 2023

Updates to managed policy	Security Hub CSPM added a new tagging permission to the `AWSSecurityHubServiceRolePolicy` managed policy that allows Security Hub CSPM to read resource tags related to findings.	November 7, 2023

New security controls	The following new Security Hub CSPM controls are available: • the section called "[AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys" • the section called "[DMS.6] DMS replication instances should have automatic minor version upgrade enabled" • the section called "[DMS.7] DMS replication tasks for the target database should have logging enabled" • the section called "[DMS.8] DMS replication tasks for the source database should have logging enabled" • the section called "[DMS.9] DMS endpoints should use SSL" • the section called "[DocumentDB.3] Amazon DocumentDB manual cluster snapshots should not be public" • the section called "[DocumentDB.4] Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs"	October 10, 2023

- the section called "[DocumentDB.5] Amazon DocumentDB clusters should have deletion protection enabled"
- the section called "[ECS.9] ECS task definitions should have a logging configuration"
- the section called "[EventBridge.3] EventBridge custom event buses should have a resource-based policy attached"
- the section called "[EventBridge.4] EventBridge global endpoints should have event replication enabled"
- the section called "[MSK.1] MSK clusters should be encrypted in transit among broker nodes"
- the section called "[MQ.5] ActiveMQ brokers should use active/standby deployment mode"
- the section called "[MQ.6] RabbitMQ brokers should use cluster deployment mode"
- the section called "[NetworkFirewall.9] Network Firewall firewalls should have deletion protection enabled"

	• the section called "[RDS.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs" • the section called "[RDS.35] RDS DB clusters should have automatic minor version upgrade enabled" • the section called "[Route53.2] Route 53 public hosted zones should log DNS queries" • the section called "[WAF.12] AWS WAF rules should have CloudWatch metrics enabled"	
Updates to managed policy	Security Hub CSPM added new Organizations actions to the `AWSSecurityHubServiceRolePolicy` managed policy that allow Security Hub CSPM to retrieve account and organizational unit (OU) information. We also added new Security Hub CSPM actions that allow Security Hub CSPM to read and update service configurations, including standards and controls.	September 27, 2023

Existing security controls added to Service-Managed Standard: AWS Control Tower	The following existing Security Hub CSPM controls have been added to Service-Managed Standard: AWS Control Tower. • the section called "[Athena.1] Athena workgroups should be encrypted at rest" • the section called "[DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest" • the section called "[DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period" • the section called "[Neptune.1] Neptune DB clusters should be encrypted at rest" • the section called "[Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs" • the section called "[Neptune.3] Neptune DB cluster snapshots should not be public" • the section called "[Neptune.4] Neptune	September 26, 2023

DB clusters should have deletion protection enabled”

- the section called “[Neptune.5] Neptune DB clusters should have automated backups enabled”
- the section called “[Neptune.6] Neptune DB cluster snapshots should be encrypted at rest”
- the section called “[Neptune.7] Neptune DB clusters should have IAM database authentication enabled”
- the section called “[Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots”
- the section called “[RDS.27] RDS DB clusters should be encrypted at rest”

Consolidated controls view and consolidated control findings available in AWS GovCloud (US)	Consolidated controls view and consolidated control findings are now available in the AWS GovCloud (US) Region. The **Controls** page of the Security Hub CSPM console shows all your controls across standards. Each control has the same control ID across standards. When you turn on consolidated control findings, you receive a single finding per security check even when a control applies to multiple enabled standards.	September 6, 2023
Consolidated controls view and consolidated control findings available in China Regions	Consolidated controls view and consolidated control findings are now available in the China Regions. The **Controls** page of the Security Hub CSPM console shows all your controls across standards. Each control has the same control ID across standards. When you turn on consolidated control findings, you receive a single finding per security check even when a control applies to multiple enabled standards.	August 28, 2023

Security Hub CSPM available in Israel (Tel Aviv) Region	Security Hub CSPM is now available in Israel (Tel Aviv). All Security Hub CSPM features are now available in this Region, with the exception of certain security controls. For more informati on, see Availability of controls by Region.	August 8, 2023

New security controls	The following new Security Hub CSPM controls are available: • the section called "[Athena.1] Athena workgroups should be encrypted at rest" • the section called "[DocumentDB.1] Amazon DocumentDB clusters should be encrypted at rest" • the section called "[DocumentDB.2] Amazon DocumentDB clusters should have an adequate backup retention period" • the section called "[Neptune.1] Neptune DB clusters should be encrypted at rest" • the section called "[Neptune.2] Neptune DB clusters should publish audit logs to CloudWatch Logs" • the section called "[Neptune.3] Neptune DB cluster snapshots should not be public" • the section called "[Neptune.4] Neptune DB clusters should have	July 28, 2023

	deletion protection enabled” • the section called “[Neptune.5] Neptune DB clusters should have automated backups enabled” • the section called “[Neptune.6] Neptune DB cluster snapshots should be encrypted at rest” • the section called “[Neptune.7] Neptune DB clusters should have IAM database authentication enabled” • the section called “[Neptune.8] Neptune DB clusters should be configured to copy tags to snapshots” • the section called “[RDS.27] RDS DB clusters should be encrypted at rest”	
New operators for automation rule criteria	You can now use CONTAINS and NOT_CONTAINS comparison operators for automation rule map and string criteria.	July 25, 2023
Automation rules	Security Hub CSPM now offers automation rules that automatically update findings based on criteria that you specify.	June 13, 2023

New third party integration	Snyk is a new third-party integration that sends findings to Security Hub CSPM.	June 12, 2023

Existing security controls added to Service-Managed Standard: AWS Control Tower	The following existing Security Hub CSPM controls have been added to Service-Managed Standard: AWS Control Tower. • the section called "[Account.1] Security contact information should be provided for an AWS account" • the section called "[APIGateway.8] API Gateway routes should specify an authorization type" • the section called "[APIGateway.9] Access logging should be configured for API Gateway V2 Stages" • the section called "[CodeBuild.3] CodeBuild S3 logs should be encrypted" • the section called "[EC2.25] Amazon EC2 launch templates should not assign public IPs to network interfaces" • the section called "[ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS" • the section called "[Redshift.10] Redshift clusters	June 12, 2023

should be encrypted at rest”

- the section called “[SageMaker.2] SageMaker notebook instances should be launched in a custom VPC”
- the section called “[SageMaker.3] Users should not have root access to SageMaker notebook instances”
- the section called “[WAF.10] AWS WAF web ACLs should have at least one rule or rule group”

New security controls	The following new Security Hub CSPM controls are available: • the section called "[ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits" • the section called "[AppSync.2] AWS AppSync should have field-level logging enabled" • the section called "[CloudFront.13] CloudFront distributions should use origin access control" • the section called "[Elastic Beanstalk.3] Elastic Beanstalk should stream logs to CloudWatch" • the section called "[S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys" • the section called "[StepFunctions.1] Step Functions state machines should have logging turned on"	June 6, 2023

Security Hub CSPM available in Asia Pacific (Melbourne)	Security Hub CSPM is now available in Asia Pacific (Melbourne). All Security Hub CSPM features are now available in this Region, with the exception of certain security controls. For more information, see Availability of controls by Region.	May 25, 2023
Finding history	Security Hub CSPM can now track the history of a finding during the last 90 days.	May 4, 2023
New security controls	The following new Security Hub CSPM controls are available: • the section called "[EKS.1] EKS cluster endpoints should not be publicly accessible" • the section called "[ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL" • the section called "[Redshift.10] Redshift clusters should be encrypted at rest" • the section called "[S3.15] S3 general purpose buckets should have Object Lock enabled"	March 29, 2023

Expanded support for consolidated control findings	The Automated Security Response on AWS v2.0.0 now supports consolidated control findings.	March 24, 2023
Security Hub CSPM available in new AWS Regions	Security Hub CSPM is now available in Asia Pacific (Hyderabad), Europe (Spain), and Europe (Zurich). Limits exist on which controls are available in these Regions.	March 21, 2023
Update to managed policy	Security Hub CSPM has updated an existing permission in the `AWSSecurityHubServiceRolePolicy` managed policy.	March 17, 2023

New security controls for NIST 800-53 standard	Security Hub CSPM has added the following security controls, which are applicable to the NIST 800-53 standard: • the section called "[Account.2] AWS accounts should be part of an AWS Organizations organization" • the section called "[CloudWatch.15] CloudWatch alarms should have specified actions configured" • the section called "[CloudWatch.16] CloudWatch log groups should be retained for a specified time period" • the section called "[CloudWatch.17] CloudWatch alarm actions should be activated" • the section called "[DynamoDB.4] DynamoDB tables should be present in a backup plan" • the section called "[EC2.28] EBS volumes should be covered by a backup plan" • **EC2.29 – EC2 instances should be launched in a VPC** (retired) • the section called "[RDS.26] RDS DB instances should	March 3, 2023

	be protected by a backup plan” • the section called “[S3.14] S3 general purpose buckets should have versioning enabled” • the section called “[WAF.11] AWS WAF web ACL logging should be enabled”	
National Institute of Standards and Technology (NIST) 800-53 Rev. 5	Security Hub CSPM now supports the NIST 800-53 Rev. 5 standard with more than 200 applicable security controls.	February 28, 2023
Consolidated controls view and control findings	With the release of consolidated controls view, the **Controls** page of the Security Hub CSPM console shows all your controls across standards. Each control has the same control ID across standards. When you turn on consolidated control findings, you receive a single finding per security check even when a control applies to multiple enabled standards.	February 23, 2023

New security controls	The following new Security Hub CSPM controls are available. Some controls have Regional limitations. • the section called "[ElastiCache.1] ElastiCache (Redis OSS) clusters should have automatic backups enabled" • the section called "[ElastiCache.2] ElastiCache clusters should have automatic minor version upgrades enabled" • the section called "[ElastiCache.3] ElastiCache replication groups should have automatic failover enabled" • the section called "[ElastiCache.4] ElastiCache replication groups should be encrypted at rest" • the section called "[ElastiCache.5] ElastiCache replication groups should be encrypted in transit" • the section called "[ElastiCache.6] ElastiCache (Redis OSS) replication groups of earlier versions should have Redis OSS AUTH enabled" • the section called "[ElastiCache.7] ElastiCache clusters	February 16, 2023

	should not use the default subnet group"	
New ASFF fields	Security Hub CSPM has added ProductFields.ArchivalReasons:0/Description and ProductFields.ArchivalReasons:0/ReasonCode to the AWS Security Finding Format (ASFF).	February 8, 2023
New ASFF fields	Security Hub CSPM has added Compliance.AssociatedStandards and Compliance.SecurityControlId to the AWS Security Finding Format (ASFF).	January 31, 2023
Vulnerability details now available	You can now see vulnerability details in the Security Hub CSPM console for findings that Amazon Inspector sends to Security Hub CSPM.	January 14, 2023
Security Hub CSPM is available in Middle East (UAE)	Security Hub CSPM is now available in Middle East (UAE). Some controls have Regional limits.	January 12, 2023
Added third-party integration with MetricStream	Security Hub CSPM now supports a third-party integration with MetricStream in all Regions except China and AWS GovCloud (US).	January 11, 2023

Increased organizational account limit	Security Hub CSPM now supports up to 11,000 member accounts for each Security Hub CSPM administrator account per Region.	December 27, 2022
ElasticBeanstalk.3 rolled back	Security Hub CSPM rolled back the control **[ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch** from the FSBP standard in all Regions.	December 21, 2022
Security Hub CSPM adds new security controls	New Security Hub CSPM controls are available to customers who have enabled the FSBP standard. Some controls have Regional limitations.	December 15, 2022
Guidance on upcoming features	Security Hub CSPM is planning to release two new features: consolidated controls view and consolidated control findings. These upcoming features may impact existing workflows that rely on control finding fields and values.	December 9, 2022
Amazon Security Lake integration now available	Security Lake now integrates with Security Hub CSPM by receiving Security Hub CSPM findings.	November 29, 2022

Support for Service-Managed Standard: AWS Control Tower	Security Hub CSPM supports a new security standard called Service-Managed Standard: AWS Control Tower. AWS Control Tower manages this standard.	November 28, 2022
CIS AWS Foundations Benchmark v1.4.0 now available in China Regions	Security Hub CSPM now supports CIS AWS Foundations Benchmark v1.4.0 in the China Regions.	November 18, 2022
Jira Service Management Cloud integration now available	Jira Service Management Cloud now receives Security Hub CSPM findings in all available Regions, except the China Regions.	November 17, 2022
AWS IoT Device Defender integration now available	AWS IoT Device Defender now sends findings to Security Hub CSPM in all available Regions.	November 17, 2022
Support for CIS AWS Foundations Benchmark v1.4.0	Security Hub CSPM now provides security controls that support CIS AWS Foundations Benchmark v1.4.0. This standard is available in all available Regions, except the China Regions.	November 9, 2022

Support for Security Hub CSPM announcements in AWS GovCloud (US)	You can now subscribe to Security Hub CSPM announcements with Amazon Simple Notification Service (Amazon SNS) in AWS GovCloud (US-East) and AWS GovCloud (US-West) to receive notifications about Security Hub CSPM.	October 3, 2022
AWS Security Hub CSPM adds a new security control	The new Security Hub CSPM control **AutoScaling.9** is available to customers who have enabled the FSBP standard. Controls may have Regional limitations.	September 1, 2022
Subscribe to Security Hub CSPM announcements	You can now subscribe to Security Hub CSPM announcements with Amazon Simple Notification Service (Amazon SNS) to receive notifications about Security Hub CSPM.	August 29, 2022
Region expansion for cross-Region aggregation	Cross-Region aggregation is now available for findings, finding updates, and insights across AWS GovCloud (US).	August 2, 2022
New third-party product integrations	Fortinet - FortiCNP is a third-party integration that receives Security Hub CSPM findings, and JFrog is a third-par ty integration that sends findings to Security Hub CSPM.	July 26, 2022

EC2.27 is retired	Security Hub CSPM has retired **EC2.27 - Running EC2 Instances should not use key pairs**, a former control in the AWS Foundational Security Best Practices (FSBP) standard.	July 20, 2022
Lambda.2 no longer supports python3.6	Security Hub CSPM no longer supports python3.6 as a parameter for **Lambda.2 - Lambda functions should use supported runtimes**, a control in the AWS Foundational Security Best Practices (FSBP) standard.	July 19, 2022
AWS Security Hub CSPM adds new security controls	New Security Hub CSPM controls are available to customers who have enabled the FSBP standard. Some controls have Regional limitations.	June 22, 2022
AWS Security Hub CSPM supports a new Region	Security Hub CSPM is now available in Asia Pacific (Jakarta). Some controls are not available in this Region.	June 7, 2022
Improved integration between AWS Security Hub CSPM and AWS Config	Security Hub CSPM users can see the results of AWS Config rule evaluations as findings in Security Hub CSPM.	June 6, 2022

Added ability to opt out of auto-enabled standards	For users who have integrated with AWS Organizations, this feature allows you to log into the Security Hub CSPM administrator account and opt new member accounts out of auto-enabled standards.	April 25, 2022
Expanded cross-Region aggregation	Added cross-Region aggregation to control statuses and security scores.	April 20, 2022
CompanyName and ProductName are now top level attributes	Added new top level attributes for setting company and product names associated with custom integrations	April 1, 2022
Added new controls to the AWS Foundational Security Best Practices standard	Added 5 new controls to the AWS Foundational Security Best Practices standard.	March 31, 2022
Added new resource details objectes to ASFF	Added `AwsRdsDbSecurityGroup` resource type to ASFF.	March 25, 2022
Added additional resources details in ASFF	Added additional details to `AwsAutoScalingScalingGroup` , `AwsElbLoadBalancer` , `AwsRedshiftCluster` , and `AwsCodeBuildProject` .	March 25, 2022
Added new controls to the AWS Foundational Security Best Practices standard	Added 15 new controls to the AWS Foundational Security Best Practices standard.	March 16, 2022

Added new controls to the AWS Foundational Security Best Practices standard and Payment Card Industry Data Security Standard (PCI DSS)	Added new controls for Amazon OpenSearch Service, Amazon RDS, Amazon EC2, Elastic Load Balancing, and CloudFront to the AWS Foundational Security Best Practices standard. Also added two new controls for OpenSearch Service to the PCI DSS.	February 15, 2022
Added new field to ASFF	Added new field: Sample.	January 26, 2022
Added integration with AWS Health	AWS Health uses service-to-service event messaging to send findings to Security Hub CSPM.	January 19, 2022
Added integration with AWS Trusted Advisor	Trusted Advisor sends the results of its checks to Security Hub CSPM as Security Hub CSPM findings. Security Hub CSPM sends the results of its AWS Foundational Security Best Practices checks to Trusted Advisor.	January 18, 2022

Updated resource details objects in ASFF	Added `MixedInstancesPolicy` and `AvailabilityZones` to `AwsAutoScalingAutoScalingGroup`. Added `MetadataOptions` to `AwsAutoScalingLaunchConfiguration`. Added `BucketVersioningConfiguration` to `AwsS3Bucket`.	December 20, 2021
Updated output for ASFF documentation	The descriptions of ASFF attributes were previously in a single topic. Each top-level object and each resource details object is now in its own topic. The ASFF syntax topic contains links to those topics.	December 20, 2021
Added new resource details objects to ASFF for AWS Network Firewall	For AWS Network Firewall, added the following resource details objects: `AwsNetworkFirewallFirewall`, `AwsNetworkFireFirewallPolicy`, and `AwsNetworkFirewallRuleGroup`.	December 20, 2021
Added support for the new version of Amazon Inspector	Security Hub CSPM is integrated with the new version of Amazon Inspector as well as with Amazon Inspector Classic. Amazon Inspector sends findings to Security Hub CSPM.	November 29, 2021

Changed the severity of EC2.19	The severity of EC2.19 (Security groups should not allow unrestricted access to ports with high risk) is changed from High to Critical.	November 17, 2021
New integration with Sonrai Dig	Security Hub CSPM now offers an integration with Sonrai Dig. Sonrai Dig monitors cloud environments to identify security risks. Sonrai Dig sends findings to Security Hub CSPM.	November 12, 2021
Updated check for CIS 2.1 and CloudTrail.1 controls	In addition to checking that at least one multi-Region CloudTrail trail is in place, CIS 2.1 and CloudTrail.1 now also check that the `ExcludeManagementEventSources` parameter is empty in at least one of the multi-Region CloudTrail trails.	November 9, 2021
Added support for VPC endpoints	Security Hub CSPM is now integrated with AWS PrivateLink and supports VPC endpoints.	November 3, 2021
Added controls to the AWS Foundational Security Best Practices standard	Added new controls for Elastic Load Balancing (ELB.2 and ELB.8) and AWS Systems Manager (SSM.4).	November 2, 2021

Added ports to the check for the EC2.19 control	EC2.19 now also checks that security groups do not allow unrestricted ingress access to the following ports: 3000 (Go, Node.js, and Ruby web development frameworks), 5000 (Python web development frameworks), 8088 (legacy HTTP port), and 8888 (alternative HTTP port)	October 27, 2021
Added the integration with Logz.io Cloud SIEM	Logz.io is a provider of Cloud SIEM that provides advanced correlation of log and event data to help security teams to detect, analyze, and respond to security threats in real time. Logz.io receives findings from Security Hub CSPM.	October 25, 2021
Added support for cross-Region aggregation of findings	Cross-Region aggregation allows you to view all of your findings without having to change Regions. Administrator accounts choose an aggregation Region and linked Regions. Findings for the administrator account and its member accounts are aggregated from the linked Regions to the aggregation Region.	October 20, 2021

Updated resource details objects in ASFF	Added viewer certificate details to `AwsCloudFrontDistribution`. Added additional details to `AwsCodeBuildProject`. Added load balancer attributes to `AwsElbV2LoadBalancer`. Added the S3 bucket owner account identifier to `AwsS3Bucket`.	October 8, 2021
Added new resource details objects to ASFF	Added the following new resource details objects to ASFF: `AwsEc2VpcEndpointService`, `AwsEcrRepository`, `AwsEksCluster`, `AwsOpenSearchServiceDomain`, `AwsWafRateBasedRule`, `AwsWafRegionalRateBasedRule`, `AwsXrayEncryptionConfig`	October 8, 2021
Removed deprecated runtime from the Lambda.2 control	In the AWS Foundational Security Best Practices standard, removed the `dotnetcore2.1` runtime from **[Lambda.2] Lambda functions should use supported runtimes**.	October 6, 2021

New name for Check Point integration	The integration with Check Point Dome9 Arc is now Check Point CloudGuard Posture Management. The integration ARN did not change.	October 1, 2021
Removed the integration with Alcide	The integration with Alcide kAudit is discontinued.	September 30, 2021
Changed the severity of EC2.19	The severity of **[EC2.19] Security groups should not allow unrestricted access to ports with high risk** is changed from Medium to High.	September 30, 2021
Integration with AWS Organizations is now supported in the China Regions	The Security Hub CSPM integration with Organizations is now supported in China (Beijing) and China (Ningxia).	September 20, 2021
New AWS Config rule for the S3.1 and PCI.S3.6 controls	Both S3.1 and PCI.S3.6 verify that the Amazon S3 Block Public Access setting is enabled. The AWS Config rule for these controls is changed from `s3-account-level-public-access-blocks` to `s3-account-level-public-access-blocks-periodic`.	September 14, 2021

Removed deprecated runtimes from the Lambda.2 control	In the AWS Foundational Security Best Practices standard, removed the `nodejs10.x` and `ruby2.5` runtimes from **[Lambda.2] Lambda functions should use supported runtimes**.	September 13, 2021
Changed the severity of the CIS 2.2 control	In the CIS AWS Foundations Benchmark standard, the severity for **2.2. – Ensure CloudTrail log file validation is enabled** is changed from Low to Medium.	September 13, 2021
Updated ECS.1, Lambda.2, and SSM.1 in the AWS Foundational Security Best Practices standard	In the AWS Foundational Security Best Practices standard, ECS.1 now has a `SkipInactiveTaskDefinitions` parameter that is set to `true`. This ensures that the control only checks active task definitions. For Lambda.2, added Python 3.9 to the list of runtimes. SSM.1 now checks both stopped and running instances.	September 7, 2021
PCI.Lambda.2 control now excludes Lambda@Edge resources	In the Payment Card Industry Data Security Standard (PCI DSS) standard, the PCI.Lambda.2 control now excludes Lambda@Edge resources.	September 7, 2021

Added the integration with HackerOne Vulnerability Intelligence	Security Hub CSPM now offers an integration with HackerOne Vulnerability Intelligence. The integration sends findings to Security Hub CSPM.	September 7, 2021
Updated resource details objects in ASFF	For `AwsKmsKey` , added `KeyRotationStatus` . For `AwsS3Bucket` , added `AccessControlList` , `BucketLoggingConfiguration` , `BucketNotificationConfiguration` , and `BucketWebsiteConfiguration` .	September 2, 2021
Added new resource details objects to ASFF	Added the following new resource details objects to ASFF: `AwsAutoScalingLaunchConfiguration` , `AwsEc2VpnConnection` , and `AwsEcrContainerImage` .	September 2, 2021
Added details to the `Vulnerabilities` object in ASFF	In `Cvss` , added `Adjustments` and `Source`. In `VulnerablePackages` , added the file path and package manager.	September 2, 2021
Systems Manager Explorer and OpsCenter integration now supported in the China Regions	The Security Hub CSPM integration with SSM Explorer and OpsCenter is now supported in China (Beijing) and China (Ningxia).	August 31, 2021

Retiring the Lambda.4 control	Security Hub CSPM is retiring the control **[Lambda.4] Lambda functions should have a dead-letter queue configured**. When a control is retired, it no longer displays on the console, and Security Hub CSPM does not perform checks against it.	August 31, 2021
Retiring the PCI.EC2.3 control	Security Hub CSPM is retiring the control **[PCI.EC2.3] Unused EC2 security groups should be removed**. When a control is retired, it no longer displays on the console, and Security Hub CSPM does not perform checks against it.	August 27, 2021
Change to how Security Hub CSPM sends findings to custom actions	When you send findings to a custom action, Security Hub CSPM now sends each finding in a separate **Security Hub Findings - Custom Action** event.	August 20, 2021
Added a new compliance status reason code for custom Lambda runtimes	Added a new `LAMBDA_CUSTOM_RUNTIME_DETAILS_NOT_AVAILABLE` compliance status reason code. This reason code indicates that Security Hub CSPM could not perform a check against a custom Lambda runtime.	August 20, 2021

AWS Firewall Manager integration now supported in the China Regions	The Security Hub CSPM integration with Firewall Manager is now supported in China (Beijing) and China (Ningxia).	August 19, 2021
New integrations with Caveonix Cloud and Forcepoint Cloud Security Gateway	Security Hub CSPM now offers integrations with Caveonix Cloud and Forcepoint Cloud Security Gateway. Both integrations send findings to Security Hub CSPM.	August 10, 2021
Added new `CompanyName`, `ProductName`, and `Region` attributes to ASFF	Added `CompanyName`, `ProductName`, and `Region` fields to the top level of the ASFF. These fields are populated automatically and, except for custom product integrations, cannot be updated using `BatchImportFindings` or `BatchUpdateFindings`. On the console, finding filters use these new fields. In the API, the `CompanyName` and `ProductName` filters use the attributes that are under `ProductFields`.	July 23, 2021

Added and updated resource details objects in ASFF	Added a new `AwsRdsEventSubscription` resource type and resource details. Added resource details for the `AwsEcsService` resource type. Added attributes to the `AwsElasticsearchDomain` resource details object.	July 23, 2021
Added controls to the AWS Foundational Security Best Practices standard	Added new controls for Amazon API Gateway (APIGateway.5), Amazon EC2 (EC2.19), Amazon ECS (ECS.2), Elastic Load Balancing (ELB.7), Amazon OpenSearch Service (ES.5 through ES.8), Amazon RDS (RDS.16 through RDS.23), Amazon Redshift (Redshift.4), and Amazon SQS (SQS.1).	July 20, 2021
Moved a permission within the service-linked role managed policy	Moved the `config:PutEvaluations` permission within the managed policy `AWSSecurityHubServiceRolePolicy`, so that it is applied to all resources.	July 14, 2021

Added controls to the AWS Foundational Security Best Practices standard	Added new controls for Amazon API Gateway (APIGateway.4), Amazon CloudFront (CloudFront.5 and CloudFront.6), Amazon EC2 (EC2.17 and EC2.18), Amazon ECS (ECS.1), Amazon OpenSearch Service (ES.4), AWS Identity and Access Management (IAM.21), Amazon RDS (RDS.15), and Amazon S3 (S3.8).	July 8, 2021
Added new compliance status reason codes for control findings	`INTERNAL_SERVICE_ERROR` indicates that an unknown error occurred. `SNS_TOPIC_CROSS_ACCOUNT` indicates that the SNS topic is owned by a different account. `SNS_TOPIC_INVALID` indicates that the associated SNS topic is invalid.	July 6, 2021
Added the integration with Amazon Q Developer in chat applications	Added the integration with Amazon Q Developer in chat applications. Security Hub CSPM sends findings to Amazon Q Developer in chat applications.	June 30, 2021

Added a new permission to the service-linked role managed policy	Added a new permission to the managed policy `AWSSecurityHubServiceRolePolicy` to allow the service-linked role to deliver evaluation results to AWS Config.	June 29, 2021
New and updated resource details objects in the ASFF	Added new resource details objects for ECS clusters and ECS task definitions. Updated the EC2 instance object to list the associated network interfaces. Added the client certificate ID for the API Gateway V2 stages. Added the lifecycle configuration for S3 buckets.	June 24, 2021
Updated the calculation of aggregated control statuses and standard security scores	Security Hub CSPM now calculates the overall control status and standard security score every 24 hours. For administrator accounts, the score now reflects whether each control is enabled or disabled for each account.	June 23, 2021
Updated information about Security Hub CSPM handling of suspended accounts	Added information on how Security Hub CSPM handles accounts that are suspended in AWS.	June 23, 2021

Added tabs to display the enabled and disabled controls for the individual administrator account	For the administrator account, the main tabs on the standard details page contain aggregated information across accounts. The new **Enabled for this account** and **Disabled for this account** tabs list the accounts that are enabled or disabled for the individual administrator account.	June 23, 2021
Added `java8.al2` to the parameters for `Lambda.2`	In the AWS Foundational Security Best Practices standard, added `java8.al2` to the supported runtimes for the `Lambda.2` control.	June 8, 2021
New integrations with MicroFocus ArcSight and NETSCOUT Cyber Investigator	Added integrations with MicroFocus ArcSight and NETSCOUT Cyber Investigator. MicroFocus ArcSight receives findings from Security Hub CSPM. NETSCOUT Cyber Investigator sends findings to Security Hub CSPM.	June 7, 2021
Added details for `AWSSecurityHubServiceRolePolicy`	Updated the managed policies section to add details for the existing managed policy `AWSSecurityHubServiceRolePolicy`, which is used by the Security Hub CSPM service-linked role.	June 4, 2021

New integration with Jira Service Management	The AWS Service Management Connector for Jira sends findings to Jira and uses them to create Jira issues. When the Jira issues are updated, the corresponding findings in Security Hub CSPM also are updated.	May 26, 2021
Updated the supported controls list for the Asia Pacific (Osaka) Region	Updated the CIS AWS Foundations standard and the Payment Card Industry Data Security Standard (PCI DSS) to indicate the controls that are not supported in Asia Pacific (Osaka).	May 21, 2021
New integration with Sysdig Secure for cloud	Added an integration with Sysdig Secure for cloud. The integration sends findings to Security Hub CSPM.	May 14, 2021

Added controls to the AWS Foundational Security Best Practices standard	Added new controls for Amazon API Gateway (APIGateway.2 and APIGateway.3), AWS CloudTrail (CloudTrail.4 and CloudTrail.5), Amazon EC2 (EC2.15 and EC2.16), AWS Elastic Beanstalk (ElasticBeanstalk.1 and ElasticBeanstalk.2), AWS Lambda (Lambda.4), Amazon RDS (RDS.12 – RDS.14), Amazon Redshift (Redshift.7), AWS Secrets Manager (SecretsManager.3 and SecretsManager.4), and AWS WAF (WAF.1).	May 10, 2021
Updates to GuardDuty and Amazon RDS controls	Changed the severity of `GuardDuty.1` and `PCI.GuardDuty.1` from Medium to High. Added a `databaseEngines` parameter to `RDS.8`.	May 4, 2021
Added new resource details to the ASFF	In `Resources.Details`, added new resource details objects for Amazon EC2 network ACLs, Amazon EC2 subnets, and AWS Elastic Beanstalk environments.	May 3, 2021
Added console fields to provide filter values for Amazon EventBridge rules	The new predefined filter patterns for Security Hub CSPM EventBridge rules provide console fields that you can use to specify filter values.	April 30, 2021

Added the integration with AWS Systems Manager Explorer and OpsCenter	Security Hub CSPM now supports an integration with Systems Manager Explorer and OpsCenter. The integration receives findings from Security Hub CSPM and updates those findings in Security Hub CSPM.	April 26, 2021
New type for product integrations	A new integration type, `UPDATE_FINDINGS_IN_SECURITY_HUB`, indicates that a product integration updates findings that it receives from Security Hub CSPM.	April 22, 2021
Changed "master account" to "administrator account"	The term "master account" is changed to "administrator account." The term is also changed in the Security Hub CSPM console and API.	April 22, 2021
Updated APIGateway.1 to replace HTTP with Websocket	Updated the title, description, and remediation for APIGateway.1. The control now checks for Websocket API execution logging instead of for HTTP API execution logging.	April 9, 2021
Amazon GuardDuty integration now supported in Beijing and Ningxia	The Security Hub CSPM integration with GuardDuty is now supported in the China (Beijing) and China (Ningxia) Regions.	April 5, 2021

Added `nodejs14.x` to the supported runtimes for Lambda.2 control	The Lambda.2 control in the Foundational Security Best Practices standard now supports the `nodejs14.x` runtime.	March 30, 2021
Security Hub CSPM launched in Asia Pacific (Osaka)	Security Hub CSPM is now available in the Asia Pacific (Osaka) Region.	March 29, 2021
Added finding provider fields to finding details	On the finding details panel, the new **Finding Provider Fields** section contains the finding provider values for confidence, criticality, related findings, severity, and types.	March 24, 2021
Added option to receive sensitive findings from Amazon Macie	The integration with Macie can now be configured to send sensitive findings to Security Hub CSPM.	March 23, 2021
Transitioning to AWS Organizations for account management	For customers who have an existing administrator account with member accounts, added new information on how to change from managing accounts by invitation to managing accounts using Organizations.	March 22, 2021

New objects in ASFF for information about Amazon S3 Public Access Block configuration	In `Resources`, a new `AwsS3AccountPublicAccessBlock` resource type and details object provides information about the Amazon S3 Public Access Block configuration for accounts. In the `AwsS3Bucket` resource details object, the `PublicAccessBlockConfiguration` object provides the Public Access Block configuration for the S3 bucket.	March 18, 2021
New object in ASFF to allow finding providers to update specific fields	The new `FindingProviderFields` object in ASFF is used in `BatchImportFindings` to provide values for `Confidence`, `Criticality`, `RelatedFindings`, `Severity`, and `Types`. The original fields should only be updated using `BatchUpdateFindings`.	March 18, 2021
New `DataClassification` object for resources in ASFF	The new `Resources.DataClassification` object in ASFF is used to provide information about sensitive data that was detected on the resource.	March 18, 2021

Added CONFIG_RETURNS_NOT_APPLICABLE value to the available compliance status codes	For the NOT_AVAILABLE compliance status, removed the reason code RESOURCE_NO_LONGER_EXISTS and added the reason code CONFIG_RETURNS_NOT_APPLICABLE .	March 16, 2021
New managed policy for integration with AWS Organizations	A new managed policy, AWSSecurityHubOrganizationsAccess , provides the Organizations permissions that are needed by the organization management account and the delegated Security Hub CSPM administrator account.	March 15, 2021
Managed policy and service-linked role information moved to the Security chapter	The information on managed policies is revised and expanded. Both the managed policy information and the information on service-linked roles has moved to the Security chapter.	March 15, 2021
New integration with SecureCloudDB	Added SecureCloudDB to the list of third-party integrations. SecureCloudDB is a cloud native database security tool that provides comprehensive visibility of internal and external security postures and activity. SecureCloudDB sends findings to Security Hub CSPM.	March 4, 2021

Revised severity for CIS 1.1 and CIS 3.1 – CIS 3.14 controls	The severity of the CIS 1.1 and CIS 3.1 – CIS 3.14 controls is changed to Low.	March 3, 2021
Removed the RDS.11 control	Removed the RDS.11 control from the Foundational Security Best Practices standard.	March 3, 2021
Updated integration for Turbot	The Turbot integration is updated to both send and receive findings.	February 26, 2021
Added controls to the Foundational Security Best Practices standard	Added new controls for Amazon API Gateway (APIGateway.1), Amazon EC2 (EC2.9 and EC2.10), Amazon Elastic File System (EFS.2), Amazon OpenSearch Service (ES.2 and ES.3), Elastic Load Balancing (ELB.6), and AWS Key Management Service (AWS KMS) (KMS.3).	February 11, 2021
Added optional `ProductArn` filter to the `DescribeProducts` API	The `DescribeProducts` API operation now includes an optional `ProductArn` parameter. The `ProductArn` parameter is used to identify the specific product integration to return details for.	February 3, 2021
New integration with Antivirus for Amazon S3 from Cloud Storage Security	The integration with Antivirus for Amazon S3 sends the virus scan results to Security Hub CSPM as findings.	January 27, 2021

Updated the security score calculation process for administrator accounts	For an administrator account, Security Hub CSPM uses a separate process to calculate the security score. The new process ensures that the score includes controls that are enabled for member accounts but disabled for the administrator account.	January 21, 2021
New fields and objects in the ASFF	Added a new `Action` object to track actions that occurred against a resource. Added fields to the `AwsEc2NetworkInterface` object to track DNS names and IP addresses. Added a new `AwsSsmPatchCompliance` object to the resource details.	January 21, 2021
Added controls to the Foundational Security Best Practices standard	Added new controls for Amazon CloudFront (CloudFront.1 through CloudFront.4), Amazon DynamoDB (DynamoDB.1 through DynamoDB.3), Elastic Load Balancing (ELB.3 through ELB.5), Amazon RDS (RDS.9 through RDS.11), Amazon Redshift (Redshift.1 through Redshift.3 and Redshift.6), and Amazon SNS (SNS.1).	January 15, 2021

Workflow status is reset based on the record state or compliance status	Security Hub CSPM automatically resets the workflow status from `NOTIFIED` or `RESOLVED` to `NEW` if an archived finding is made active, or if the compliance status of a finding changes from `PASSED` to either `FAILED`, `WARNING`, or `NOT_AVAILABLE` . These changes indicate that additional investigation is required.	January 7, 2021
Added `ProductFields` information for control-based findings	For findings that are generated from controls, added information about the content of the `ProductFields` object in the AWS Security Finding Format (ASFF).	December 29, 2020
Updates to managed insights	Changed the title of insight 5. Added a new insight, 32, that checks for IAM users with suspicious activity.	December 22, 2020
Updates to IAM.7 and Lambda.1 controls	In the AWS Foundational Security Best Practices standard, updated the parameters for IAM.7. Updated the title and description of Lambda.1.	December 22, 2020

Expanded integration with ServiceNow ITSM	The ServiceNow ITSM integration allows users to automatically create incidents or problems when a Security Hub CSPM finding is received. Updates to these incidents or problems result in updates to the findings in Security Hub CSPM.	December 11, 2020
New integration with AWS Audit Manager	Security Hub CSPM now offers an integration with AWS Audit Manager. The integration allows Audit Manager to receive control-based findings from Security Hub CSPM.	December 8, 2020
New integration with Aqua Security Kube-bench	Security Hub CSPM added an integration with Aqua Security Kube-bench. The integration sends findings to Security Hub CSPM.	November 24, 2020
Cloud Custodian is now available in the China Regions	The integration with Cloud Custodian is now available in the China (Beijing) and China (Ningxia) Regions.	November 24, 2020

BatchImportFindings can now be used to update additional fields	Previously, you could not use BatchImportFindings to update the Confidence , Criticality , RelatedFindings , Severity, and Types fields. Now, if these fields have not been updated by BatchUpdateFindings , they can be updated by BatchImportFindings . Once they are updated by BatchUpdateFindings , they cannot be updated by BatchImportFindings .	November 24, 2020
Security Hub CSPM is now integrated with AWS Organizations	Customers can now manage member accounts using their Organizations account configuration. The organization management account designates the Security Hub CSPM administrator account, who determines which organization accounts to enable in Security Hub CSPM. The manual invitation process can still be used for accounts that are not part of an organization.	November 23, 2020
Removed the separate finding list format for high-volume controls	The finding list for a control no longer uses the **Findings** page format when there is a very large number of findings.	November 19, 2020

New and updated third-party integrations	Security Hub CSPM now supports integrations with cloudtamer.io, 3CORESec, Prowler, and StackRox Kubernetes Security. IBM QRadar no longer sends findings. It only receives findings.	October 30, 2020
Added option to download the list of findings from the control details page.	On the control details page, a new **Download** option allows you to download the finding list to a .csv file. The downloaded list respects any filters that are on the list. If you selected specific findings, then the downloaded list only includes those findings.	October 26, 2020
Added option to download the list of controls from the standard details page.	On the standard details page, a new **Download** option allows you to download the control list to a .csv file. The downloaded list respects any filters that are on the list. If you selected a specific control, then the downloaded list only includes that control.	October 26, 2020

New and updated partner integrations	Security Hub CSPM is now integrated with ThreatModeler. Updated the following partner integrations to reflect their new product names. Twistlock Enterprise Edition is now Palo Alto Networks - Prisma Cloud Compute. Also from Palo Alto Networks, Demisto is now Cortex XSOAR and Redlock is now Prisma Cloud Enterprise.	October 23, 2020
Security Hub CSPM launched in China (Beijing) and China (Ningxia)	Security Hub CSPM is now available in the China (Beijing) and China (Ningxia) Regions.	October 21, 2020
Revised format for ASFF attributes and third-party integrations	The lists of ASFF attributes and partner integrations now use a list-based format instead of tables. The ASFF syntax, attributes, and types taxonomy are now in separate topics.	October 15, 2020
Redesigned standard details page	The standard details page for an enabled standard now displays a tabbed list of controls. The tabs filter the control list based on the control status.	October 7, 2020
Replaced CloudWatch Events with EventBridge	Replaced references to Amazon CloudWatch Events with Amazon EventBridge.	October 1, 2020

New integrations with Blue Hexagon for AWS, Alcide kAudit, and Palo Alto Networks VM-Series.	Security Hub CSPM is now integrated with Blue Hexagon for AWS, Alcide kAudit, and Palo Alto Networks VM-Series. Blue Hexagon for AWS and kAudit send findings to Security Hub CSPM. VM-Series receives findings from Security Hub CSPM.	September 30, 2020
New and updated resource details objects in ASFF	Added new `Resources.Details` objects for `AwsApiGatewayRestApi`, `AwsApiGatewayStage`, `AwsApiGatewayV2Api`, `AwsApiGatewayV2Stage`, `AwsCertificateManagerCertificate`, `AwsElbLoadBalancer`, `AwsIamGroup`, and `AwsRedshiftCluster`. Added details to the `AwsCloudFrontDistribution`, `AwsIamRole` and `AwsIamAccessKey` objects.	September 30, 2020
New `ResourceRole` attribute for resources in ASFF to track whether a resource is an actor or a target.	The `ResourceRole` attribute for resources indicates whether the resource is the target of the finding activity or the perpetrator of the finding activity. The valid values are `ACTOR` and `TARGET`.	September 30, 2020

Added AWS Systems Manager Patch Manager to available AWS service integrations	AWS Systems Manager Patch Manager is now integrated with Security Hub CSPM. Patch Manager sends findings to Security Hub CSPM when instances in a customer's fleet go out of compliance with their patch compliance standard.	September 22, 2020
Added new controls to the AWS Foundational Security Best Practices standard	Added new controls for the following services: Amazon EC2 (EC2.7 and EC2.8), Amazon EMR (EMR.1), IAM (IAM.8), Amazon RDS (RDS.4 through RDS.8), Amazon S3 (S3.6), and AWS Secrets Manager (SecretsManager.1 and SecretsManager.2).	September 15, 2020
New context keys for IAM policy to control access to `BatchUpdateFindings` fields	IAM policies can now be configured to restrict access to fields and field values when using `BatchUpdateFindings` .	September 10, 2020
Expanded access to `BatchUpdateFindings` for member accounts	By default, member accounts now have the same access to `BatchUpdateFindings` as administrator accounts.	September 10, 2020

New controls for AWS KMS in the Foundational Security Best Practices Standard	Added two new controls (KMS.1 and KMS.2) to the Foundational Security Best Practices Standard. The new controls check whether IAM policies restrict access to AWS KMS decryption actions.	September 9, 2020
Removed account-level findings for controls	Security Hub CSPM no longer generates account-level findings for a control. Only resource-level findings are generated.	September 1, 2020
New PatchSummary object in ASFF	Added the `PatchSummary` object to the ASFF. The `PatchSummary` object provides information about the patch compliance of a resource relative to a selected compliance standard.	September 1, 2020
Redesigned control details page	The details page for controls is redesigned. The control finding list provides tabs to allow you to quickly filter the list based on the compliance status. You can also quickly see suppressed findings. Each entry provides access to additional details about the finding resource, AWS Config rule, and finding notes.	August 28, 2020

New filter options for findings	For finding filters, you can use the **is not** filter to find findings for which a field value is not equal to the filter value. You can use the **does not start with** to find findings for which a field value does not start with the specified filter value.	August 28, 2020
New resource details objects in ASFF	Added new `Resources.Details` objects for the following resource types: `AwsDynamoDbTable`, `AwsEc2Eip`, `AwsIamPolicy`, `AwsIamUser`, `AwsRdsDbCluster`, `AwsRdsDbClusterSnapshot`, `AwsRdsDbSnapshot`, `AwsSecretsManagerSecret`	August 18, 2020
New integration with RSA Archer	Security Hub CSPM is now integrated with RSA Archer. RSA Archer receives findings from Security Hub CSPM.	August 18, 2020
New Description field for AwsKmsKey	Added a `Description` field to the `AwsKmsKey` object under `Resources.Details`.	August 18, 2020
Added fields to AwsRdsDbInstance	Added several attributes to the `AwsRdsDbInstance` object under `Resources.Details`.	August 18, 2020

Updated how Security Hub CSPM determines the overall status of a control	For controls that have no findings, the status is **No data** instead of **Unknown**. The control status includes both account-level and resource-level findings. The control status does not use the workflow status of findings, except to ignore suppressed findings.	August 13, 2020
Updated how Security Hub CSPM calculates the security score for a standard	When calculating the security score for a standard, Security Hub CSPM now ignores controls with a status of **No Data**. The security score is proportion of passed controls to enabled controls, excluding controls with no data.	August 13, 2020
New option to automatically enable new controls in enabled standards	Added a **Settings** option to automatically enable new controls in standards that are enabled. You can also use the `UpdateSecurityHubConfiguration` API operation to configure this option.	July 31, 2020

New controls for the Payment Card Industry Data Security Standard (PCI DSS) standard	Added new controls to the PCI DSS standard. The identifiers of the new controls are PCI.DMS.1, PCI.EC2.5 , PCI.EC2.6, PCI.ELBV2.1, PCI.GuardDuty.1, PCI.IAM.7, PCI.IAM.8, PCI.S3.5, PCI.S3.6, PCI.SageMaker.1, PCI.SSM.2, and PCI.SSM.3.	July 29, 2020
New and updated controls for the Foundational Security Best Practices standard	Added new controls to the Foundational Security Best Practices standard. The identifiers of the new controls are AutoScaling.1, DMS.1, EC2.4, EC2.6, S3.5, and SSM.3. Updated the title of ACM.1 and changed the value of the `daysToExpiration` parameter to 30.	July 29, 2020
New `Vulnerabilities` object in the ASFF	Added the `Vulnerabilities` object, which provides information about vulnerabilities that are associated with the finding.	July 1, 2020
New `Resource.Details` objects in the ASFF for Auto Scaling groups, EC2 volumes, and EC2 VPCs	Added the `AwsAutoScalingAutoScalingGroup` , `AWSEc2Volume` , and `AwsEc2Vpc` objects to `Resource.Details` .	July 1, 2020

New `NetworkPath` object in the ASFF	Added the `NetworkPath` object, which provides information about a network path that is related to the finding.	July 1, 2020
Automatically resolve findings when `Compliance.Status` is PASSED	For findings from controls, if `Compliance.Status` is PASSED, then Security Hub CSPM automatically sets `Workflow.Status` to `RESOLVED`.	June 24, 2020
AWS Command Line Interface examples	Added AWS CLI syntax and examples for several Security Hub CSPM tasks. Includes enabling Security Hub CSPM, managing insights, managing standards and controls, managing product integrations, and disabling Security Hub CSPM.	June 24, 2020
New `Severity.Original` attribute in the ASFF	Added the `Severity.Original` attribute, which is the original severity from the finding provider. This replaces the deprecated `Severity.Product` attribute.	May 20, 2020
New `Compliance.StatusReasons` object in the ASFF for details about a control's status	Added the `Compliance.StatusReasons` object, which provides additional context for the current status of a control.	May 20, 2020

New AWS Foundational Security Best Practices standard	Added the new AWS Foundational Security Best Practices standard, which is a set of controls that detect when your deployed accounts and resources deviate from security best practices.	April 22, 2020
New console option to update the workflow status for a finding	Added information for using the Security Hub console or API to set the workflow status for findings.	April 16, 2020
New `BatchUpdateFindings` API for customer updates to findings	Added information on using `BatchUpdateFindings` to update information related to the process of investigating a finding. `BatchUpdateFindings` replaces `UpdateFindings`, which is deprecated.	April 16, 2020
Updates to the AWS Security Finding Format (ASFF)	Added several new resource types. Added a new `Label` attribute to the `Severity` object. `Label` is intended to replace the `Normalized` field. Added a new `Workflow` object to track the process of an investigation into a finding. `Workflow` contains a `Status` attribute, which replaces the existing `Workflowstate` attribute.	March 12, 2020

Updates to the Integrations page	Updated to reflect the changes to the **Integrations** page. For each integration, the page now shows the integration category and whether each integration sends findings to or receives findings from Security Hub CSPM. It also provides the specific steps required to enable each integration.	February 26, 2020
New third-party product integrations	Added the following new product integrations: Cloud Custodian, FireEye Helix, Forcepoint CASB, Forcepoint DLP, Forcepoint NGFW, Rackspace Cloud Native Security, and Vectra.ai Cognito Detect.	February 21, 2020
New security standard for the Payment Card Industry Data Security Standard (PCI DSS)	Added the Security Hub CSPM security standard for the Payment Card Industry Data Security Standard (PCI DSS). When this standard is enabled, Security Hub CSPM performs automated checks against controls related to PCI DSS requirements.	February 13, 2020

Updates to the AWS Security Finding Format (ASFF)	Added a field for related requirements for standards controls. Added new resource types and new resource details. The ASFF also now allows you to provide up to 32 resources.	February 5, 2020
New option to disable individual security standard controls	Added information on how to control whether each individual security standard control is enabled.	January 15, 2020
Updates to Security Hub CSPM concepts	Updated some descriptions and added new terms to Security Hub CSPM concepts.	September 21, 2019
AWS Security Hub CSPM general availability release	Content updates to reflect improvements made to Security Hub CSPM during the beta period.	June 25, 2019
Added remediation steps for CIS AWS Foundations checks	Added remediation steps to Security Standards Supported in AWS Security Hub CSPM.	April 15, 2019
beta release of AWS Security Hub CSPM	Published the beta release version of the *AWS Security Hub CSPM User Guide*.	November 18, 2018