User Guide

AWS Security Hub

AWS Security Hub: User Guide

Table of Contents

What is AWS Security Hub?

AWS Security Hub 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.

Security Hub 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 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 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 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 findings to other AWS services and supported third-party products.

Security Hub 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

Benefits of Security Hub

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

Reduced effort to collect and prioritize findings

Security Hub reduces the effort to collect and prioritize security findings across accounts from integrated AWS services and AWS partner products. Security Hub 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 also correlates findings across providers to help you prioritize the most important ones.

Automatic security checks against best practices and standards

Security Hub automatically runs continuous, account-level configuration and security checks based on AWS best practices and industry standards. Security Hub 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 consolidates your security findings across accounts and provider products and displays results on the Security Hub console. You can also retrieve findings through the Security Hub 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 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

Security Hub is available in most AWS Regions. For a list of Regions where Security Hub is currently available, see AWS Security Hub 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 in any of the following ways:

Security Hub 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 console provides access to your Security Hub account, data, and resources. You can perform Security Hub tasks by using the Security Hub console—view findings, create automation rules, create an aggregation Region, and more.

Security Hub API

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

AWS CLI

With the AWS CLI, you can run commands at your system's command line to perform Security Hub 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 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 only detects and consolidates findings that are generated after you enable Security Hub. It doesn't retroactively detect and consolidate security findings that were generated before you enabled Security Hub.

Security Hub only receives and processes findings in the Region where you enabled Security Hub in your account.

For full compliance with CIS AWS Foundations Benchmark security checks, you must enable Security Hub in all supported AWS Regions.

Related services

To further secure your AWS environment, consider using other AWS services in combination with Security Hub.

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

Security Hub uses service-linked rules from AWS Config to run security checks for most controls. You must enable AWS Config and record resources in AWS Config for Security Hub to generate most control findings. For more information, see Configuring AWS Config.

Security Hub free trial and pricing

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

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

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

> **Note**
>
> The Security Hub free trial is not supported in the China (Beijing) Region.

Viewing usage details and estimated cost

Security Hub provides usage information, including an estimated 30-day cost for using Security Hub. The usage details include the time remaining in the free trial. The usage information can help you to understand what your Security Hub 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 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 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 “How cross-Region aggregation works”.

Pricing details

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

Terminology and concepts

This topic describes the key concepts in AWS Security Hub to help you get started.

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.

An account can invite other accounts to enable Security Hub and become associated with that account in Security Hub. 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 administrator account for the organization. The Security Hub 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.

Administrator account

An account in Security Hub 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. 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 administrator account. The Security Hub 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.

See *Cross-Region aggregation*.

Archived finding

A finding that has a `RecordState` set to `ARCHIVED`. Archiving a finding indicates that the finding provider believes that the finding is no longer relevant. The record state is separate from the workflow status, which tracks the status of an investigation into a finding.

Finding providers can use the `BatchImportFindings` operation of the Security Hub API to archive findings that they created. Security Hub automatically archives findings for controls if the control is disabled or the associated resource is deleted, based on one of the following criteria.

- The finding is not updated in three to five days (note that this is best effort and not guaranteed).
- The associated AWS Config evaluation returns `NOT_APPLICABLE`.

By default, archived findings are excluded from findings lists in the Security Hub console. You can update the filter to include archived findings.

The `GetFindings` operation of the Security Hub API returns both active and archived findings. You can include a filter for the record state.

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

AWS Security Finding Format (ASFF)

A standardized format for the contents of findings that Security Hub aggregates or generates. The AWS Security Finding Format enables you to use Security Hub to view and analyze findings that are generated by AWS security services, third-party solutions, or Security Hub 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 only supports standard controls in the AWS GovCloud (US) Region and China Regions. Security controls are supported in all other Regions.

Custom action

A Security Hub mechanism for sending selected findings to EventBridge. A custom action is created in Security Hub. 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 (console)”.

Delegated administrator account (Organizations)

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

In Security Hub, the Security Hub administrator account is also the delegated administrator account for Security Hub. When the organization management account first designates a Security Hub administrator account, Security Hub 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 administrator account in all Regions.

Finding

The observable record of a security check or security-related detection. Security Hub generates a finding after completing a security check of a control. These are called control findings. Findings may also come from third party product integrations.

For more information about findings in Security Hub, see *Findings*.

> **Note**
>
> Findings are deleted 90 days after the most recent update or 90 days after the creation date if no update occurs. To store findings for longer than 90 days, you can configure a rule in EventBridge that routes findings to your Amazon S3 bucket.

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.

See *Cross-Region aggregation*.

Finding ingestion

The import of findings into Security Hub 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 offers several managed (default) insights that you can't modify. You can also create custom Security Hub insights to track security issues that are unique to your AWS environment and usage. For more information, see *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.

See *Cross-Region aggregation*.

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 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 administrator account

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

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

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

The Security Hub administrator account can enable any organization account as a member account. The Security Hub 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. To learn more about security standards in Security Hub, see *Standards and controls*.

Severity

The severity assigned to a Security Hub 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 assigns severity to a control, see Assigning severity to control findings.

Workflow status

The status of an investigation into a finding. 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 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.

Prerequisites and recommendations

The following prerequisites and recommendations can help you get started with using AWS Security Hub.

Integrating with AWS Organizations

AWS Organizations is a global account management service that enables AWS administrators to consolidate and centrally manage multiple AWS accounts and organizational units (OUs). 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 integrates with multiple AWS services, including Security Hub, Amazon GuardDuty, and Amazon Macie.

To help automate and streamline the management of accounts, we strongly recommend integrating Security Hub and AWS Organizations. You can integrate with Organizations if you have more than one AWS account that uses Security Hub.

For instructions on activating the integration, see Integrating Security Hub with AWS Organizations.

Using central configuration

When you integrate Security Hub and Organizations, you have the option to use a feature called central configuration to set up and manage Security Hub 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 across accounts, OUs, and Regions. The delegated administrator configures Security Hub by creating configuration policies. Within a configuration policy, you can specify the following settings:

- Whether Security Hub 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.

To learn more about central configuration, see Central configuration in Security Hub.

Configuring AWS Config

AWS Security Hub uses service-linked AWS Config rules to perform security checks for most controls.

To support these controls, AWS Config must be enabled on all accounts—both the administrator account and member accounts—in each AWS Region where Security Hub is enabled. In addition, for each enabled standard AWS Config must be configured to record resources that are required for enabled controls.

We recommend that you turn on resource recording in AWS Config before you enable Security Hub standards. If Security Hub tries to run security checks when resource recording is turned off, the checks return errors.

Security Hub does not manage AWS Config for you. If you already have AWS Config enabled, you can configure its settings through the AWS Config console or APIs.

If you enable a standard but haven't enabled AWS Config, Security Hub tries to create the AWS Config rules according to the following schedule:

- On the day 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 also tries to create the AWS Config rules when you re-apply a configuration policy that enables one or more standards.

Enabling AWS Config

If you have not enabled AWS Config already, you can enable it in one of the following ways:

- **Console or AWS CLI** – You can manually enable AWS Config using the AWS Config console or AWS CLI. See Getting started with AWS Config in the *AWS Config Developer Guide*.
- **AWS CloudFormation template** – If you want to enable AWS Config on a large number of accounts, you can enable AWS Config with the CloudFormation template **Enable AWS Config**. To access this template, see AWS CloudFormation StackSets sample templates in the *AWS CloudFormation User Guide*.
- **Github script** – Security Hub offers a GitHub script that enables Security Hub for multiple accounts across Regions. This script is useful if you haven't integrated with Organizations or if you have accounts that are not part of your organization. When you use this script to enable Security Hub, it also automatically enables AWS Config for these accounts.

For more information about enabling AWS Config to help you run Security Hub security checks, see Optimize AWS Config for AWS Security Hub to effectively manage your cloud security posture.

Turning on resource recording in AWS Config

When you turn on resource recording in AWS Config with default settings, it records all supported types of *Regional resources* that AWS Config discovers in the AWS Region in which it is running. You can also configure AWS Config to record supported types of *global resources*. You only need to record global resources in a single Region (we recommend that this be your home Region if you're using central configuration).

If you are using CloudFormation StackSets to enable AWS Config, we recommend that you run two different StackSets. Run one StackSet to record all resources, including global resources, in a single Region. Run a second StackSet to record all resources except global resources in other Regions.

You can also use Quick Setup, a capability of AWS Systems Manager, to quickly configure resource recording in AWS Config across your accounts and Regions. During the Quick Setup process, you can choose which Region you would like to record global resources in. For more information, see AWS Config configuration recorder in the *AWS Systems Manager User Guide*.

The security control Config.1 will generate failed findings in Regions where global resources are not recorded. This is expected, and you can use an automation rule to suppress these findings.

If you use the multi-account script to enable Security Hub, it automatically enables resource recording for all resources, including global resources, in all Regions. You can then update the configuration to record global resources in a single Region only. For information, see Selecting which resources AWS Config records in the *AWS Config Developer Guide*.

In order for Security Hub to accurately report findings for controls that rely on AWS Config rules, you must enable recording for the relevant resources. For a list of controls and their related AWS Config resources, see AWS Config resources required to generate control findings.AWS Config lets you 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 may delay the generation of Security Hub findings for change-triggered controls until a 24-hour period is complete.

> **Note**
>
> To generate new findings after security checks and avoid stale findings, you must have sufficient permissions for the IAM role that is attached to the configuration recorder to evaluate the underlying resources.

Cost considerations

For details about the costs associated with resource recording, see AWS Security Hub pricing and AWS Config pricing.

Security Hub may impact your AWS Config configuration recorder costs by updating the `AWS::Config::ResourceCompliance` configuration item. Updates may occur each time a Security Hub 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, and don't use this configuration item for other purposes, we recommend turning off recording for it in the AWS Config console or AWS CLI. This can reduce your AWS Config costs. You don't need to record `AWS::Config::ResourceCompliance` for security checks to work in Security Hub.

Enabling Security Hub

There are two ways to enable AWS Security Hub, 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 manually.

Verifying necessary permissions

After you sign up for Amazon Web Services (AWS), you must enable Security Hub to use its capabilities and features. To enable Security Hub, you first have to set up permissions that allow you to access the Security Hub 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 through the Organizations integration, you also should attach the AWS managed policy called `AWSSecurityHubOrganizationsAccess`.

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

Enabling Security Hub with Organizations integration

To start using Security Hub with AWS Organizations, the AWS Organizations management account for the organization designates an account as the Security Hub delegated administrator account for the organization. Security Hub 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 console

To designate the Security Hub delegated administrator when onboarding

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. Choose **Go to Security Hub**. 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 API

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

After designating the delegated administrator, we recommend that you continue setting up Security Hub with central configuration. The console prompts you to do so. By using central configuration, you can simplify the process of enabling and configuring Security Hub for your organization and ensure that your organization has adequate security coverage.

Central configuration lets the delegated administrator customize Security Hub 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 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 use central configuration, the delegated administrator has a more limited ability to configure Security Hub. For more information, see Managing accounts with AWS Organizations.

Enabling Security Hub manually

You must enable Security Hub 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 manually, you designate a Security Hub 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. When you enable Security Hub from the console, you also have the option to enable the supported security standards.

Security Hub console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. When you open the Security Hub console for the first time, choose **Go to Security Hub**.
3. On the welcome page, the **Security standards** section lists the security standards that Security Hub 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 and controls, see Security controls and standards in AWS Security Hub.
4. Choose **Enable Security Hub**.

Security Hub API

Invoke the `EnableSecurityHub` API. When you enable Security Hub 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 across multiple accounts and Regions.

The Security Hub multi-account enablement script in GitHub allows you to enable Security Hub across accounts and Regions. The script also automates the process of sending invitations to member accounts and enabling AWS Config.

The script automatically enables resource recording for all resources, including global resources, in all Regions. It does not limit recording of global resources to a single Region.

There is a corresponding script to disable Security Hub across accounts and Regions.

Next steps after enabling Security Hub

After you enable Security Hub, we recommend enabling the security standards and security controls that are important for your security needs. After you enable controls, Security Hub begins running security checks and generating control findings. You can also leverage integrations between Security Hub and other AWS services and third-party solutions to see their findings in Security Hub.

Central configuration in Security Hub

Central configuration is a Security Hub feature that helps you set up and manage Security Hub across multiple AWS accounts and AWS Regions. To use central configuration, you must first integrate Security Hub and AWS Organizations. You can integrate the services by creating an organization and designating a delegated Security Hub administrator account for the organization.

From the delegated Security Hub administrator account, you can specify how the Security Hub service, security standards, and security controls are configured in your organization accounts and organizational units (OUs) across Regions. You can configure these settings in just a few steps from one primary Region, referred to as the *home Region*. If you don't use central configuration, you must configure Security Hub separately in each account and 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 configuration policies. Configuration policies let the delegated administrator specify whether Security Hub is enabled or disabled, and which standards and controls are enabled and disabled. They can also be used to customize parameters of 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. 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.

This section provides an overview of central configuration.

Benefits of central configuration

Benefits of central configuration include the following:

Simplify configuration of the Security Hub service and capabilities

When you use central configuration, Security Hub 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 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 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 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.

Who should use central configuration?

Central configuration is most beneficial for AWS environments that include multiple Security Hub accounts. It's designed to help you centrally manage Security Hub for multiple accounts.

You can use central configuration to configure the Security Hub service, security standards, and security controls. You can also use it to customize parameters of certain controls. For information about standards and controls, see Security controls and standards in AWS Security Hub.

Central configuration terms and concepts

Understanding the following key terms and concepts can help you use Security Hub central configuration.

Central configuration

A Security Hub feature that helps the delegated Security Hub administrator account for an organization configure the Security Hub service, security standards, and security controls across multiple accounts and Regions. To configure these settings, the delegated administrator creates and manages Security Hub 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 and AWS Organizations.

Home Region

The AWS Region from which the delegated administrator centrally configures Security Hub, 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 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. You must specify at least one linked Region to use central configuration.

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*.

Security Hub configuration policy

A collection of Security Hub settings that the delegated administrator can configure for centrally managed accounts. This includes:

- Whether to enable or disable Security Hub.
- 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 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 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 console, the delegated administrator can choose the Security Hub recommended configuration policy or create custom configuration policies. With the Security Hub 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, 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 and AWS Organizations. With local configuration, the delegated administrator can choose to automatically enable Security Hub 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 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 operations that only the Security Hub delegated Security Hub 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 operations that can be used to enable or disable Security Hub, 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 the status of their account, however, the owner of a centrally managed account *can* use any `Get` or `Describe` operations.

If you use local configuration or manual account management, instead of central configuration, these account-specific operations can be used.

Organizational unit (OU)

In AWS Organizations and Security Hub, 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

An account, OU, or root that only the delegated administrator can configure across Regions by using configuration policies.

The delegated administrator account specifies whether an account is centrally managed. The delegated administrator can also change an account's status from centrally managed to self-managed, or the other way around.

Self-managed

An account, OU, or root that manages its own Security Hub settings. A self-managed account uses account-specific operations to configure Security Hub for itself separately in each Region.

This is in contrast to centrally managed accounts, which are configurable only by the delegated administrator across Regions through configuration policies.

The delegated administrator account specifies whether an account is self-managed. The delegated administrator account can also change an account's status from self-managed to centrally managed, or the other way around.

The delegated administrator can apply self-managed behavior to an account or OU. Alternatively, an account or OU can inherit self-managed behavior from a parent. The delegated administrator account can itself be a self-managed account.

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, 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.

Start using central configuration

The AWS Security Hub delegated administrator account can use central configuration to configure Security Hub, standards, and controls for multiple accounts and organizational units (OUs) across AWS Regions.

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 with AWS Organizations and designate a home Region. If you use the Security Hub console, these prerequisites are included in the opt-in workflow for central configuration.

Integrate with Organizations

You must integrate Security Hub 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 delegated administrator account. For instructions, see Integrating Security Hub 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 console or Security Hub 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 delegated administrator, Security Hub 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.

To use central configuration, you must specify at least one linked Region that is configurable from the home Region.

> **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*.

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 home Region is also your Security Hub 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 resource that specifies the home Region and linked Regions.

To designate a home Region, follow 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.

Start central configuration

Choose your preferred method, and follow the steps to start using central configuration for your organization.

Security Hub console

To centrally configure your organization

1. Open the AWS Security Hub 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, choose **Go to Security Hub**.
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 Security Hub configuration policies.

Security Hub API

To centrally configure Security Hub

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 delegated administrator in all linked Regions.
   - Enables Security Hub in the delegated administrator account in all linked Regions.
   - Designates the calling account as the Security Hub delegated administrator for new and existing accounts that use Security Hub 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 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 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 in your organization. For instructions on creating a configuration policy, see Creating and associating Security Hub configuration policies.

Example API request:

```
{
    "AutoEnable": false,
    "OrganizationConfiguration": {
        "ConfigurationType": "CENTRAL"
    }
}
```

AWS CLI

To centrally configure Security Hub

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 delegated administrator in all linked Regions.
   - Enables Security Hub in the delegated administrator account in all linked Regions.
   - Designates the calling account as the Security Hub delegated administrator for new and existing accounts that use Security Hub 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 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 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 in your organization. For instructions on creating a configuration policy, see Creating and associating Security Hub configuration policies.

Example command:

```
aws securityhub --region us-east-1 update-organization-configuration \
--no-auto-enable \
--organization-configuration '{"ConfigurationType": "CENTRAL"}'
```

Management type of organization accounts and OUs

When you use central configuration, the AWS Security Hub delegated administrator can designate each organization account and organizational unit (OU) as *centrally managed* or *self-managed*. The management type of an account or OU determines how you can specify and change its Security Hub settings.

A self-managed account or OU can configure its own Security Hub settings separately in each AWS Region. The delegated administrator can't configure Security Hub settings for a self-managed account or OU, and configuration policies can't be associated with them. In contrast, only the delegated administrator can configure Security Hub settings for centrally managed accounts and OUs 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 an account or OU between self-managed and centrally managed. By default, all accounts and OU are self-managed when you start central configuration through the Security Hub 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 self-managed account, the policy overrides the self-managed designation. The account becomes centrally managed and adopts the settings reflected in the configuration policy.

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.

Specifying settings for self-managed accounts

Self-managed accounts must configure their own settings separately in each Region.

Owners of self-managed accounts can invoke the following APIs in each Region to configure their settings:

- `EnableSecurityHub` and `DisableSecurityHub` to enable or disable the Security Hub service
- `BatchEnableStandards` and `BatchDisableStandards` to enable or disable standards
- `BatchUpdateStandardsControlAssociations` or `UpdateStandardsControl` to enable or disable controls

For descriptions of Security Hub API actions, see the *AWS Security Hub API Reference*.

Self-managed accounts can also use the Security Hub console or AWS CLI to configure their settings in each Region.

Self-managed accounts can't invoke any APIs related to Security Hub 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 accounts and OUs

Choose your preferred method, and follow the steps to designate an account or OU as centrally managed or self-managed.

Security Hub console

To choose the management type of an account or OU

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated 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 API

To choose the management type of an account or OU

1. Invoke the StartConfigurationPolicyAssociation API from the Security Hub 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 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 Security Hub configuration policies work

The delegated administrator account can create AWS Security Hub configuration policies to configure Security Hub, security standards, and security controls in your organization. After creating a configuration policy, the delegated administrator can associate it with accounts, organizational units (OUs), or the root. The delegated administrator can also view, edit, or delete configuration policies.

Policy considerations

Before you create a configuration policy in Security Hub, 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 involve global resources.

  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 skips the control configuration in unavailable Regions but applies the configuration in Regions where the control is available.
- **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.
- 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 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 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 to generate control findings. For more information, see Configuring AWS Config.

If you use central configuration, Security Hub automatically disables controls that involve global resources in all Regions except the home Region. Other controls 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. For a list of controls that involve global resources, see Controls that deal with global resources.

Recommended configuration policy

When creating a configuration policy for the *first time in the Security Hub console*, you have the option to choose the Security Hub recommended policy.

The recommended policy enables Security Hub, 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 Security Hub 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 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 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.

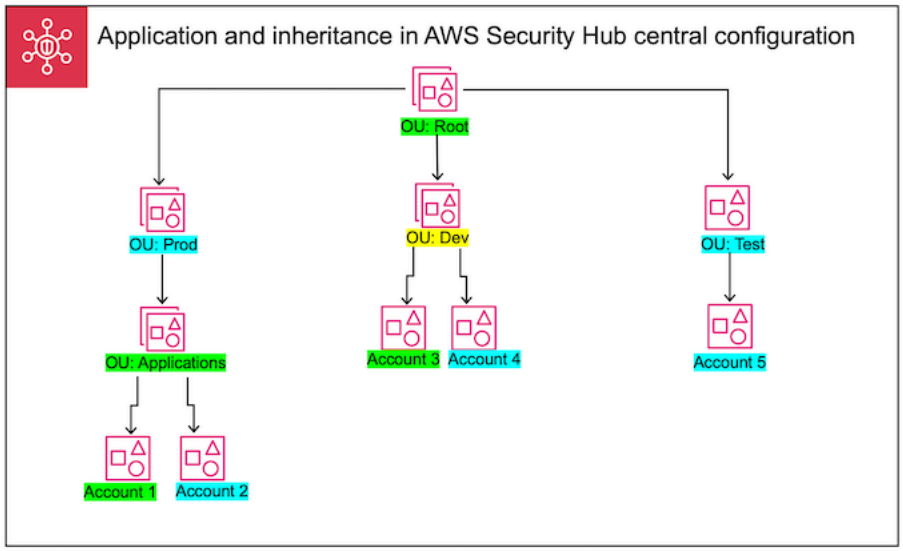
Application takes precedence over inheritance. In other words, inheritance doesn't override a configuration policy or self-managed designation that the delegated administrator has directly applied to an account or OU.

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 test the effect of a configuration policy, you can associate it with a single account or OU before associating it more widely throughout your organization.

To test a configuration policy

1. Create a custom configuration policy, but don't apply it to any accounts. Verify that the specified settings for Security Hub enablement, standards, and controls are correct.
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. For instructions on policy creation and association, see Creating and associating Security Hub configuration policies. The children of the applied accounts inherit the policy unless they're self-managed or a different configuration policy applies to them. You can also edit your configuration policies and create additional configuration policies as necessary.

Creating and associating Security Hub configuration policies

The delegated administrator account can create AWS Security Hub configuration policies and associate them with organization accounts, organizational units (OUs), or the root. You 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 Security Hub configuration policies work.

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 console, you can associate a configuration with multiple accounts or OUs at the same time. When using the Security Hub API or AWS CLI, you can associate a configuration with only one account or OU in each request.

Security Hub console

To create and associate configuration policies

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated 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 configuration across my entire organization** to use our recommended policy. The recommended policy enables Security Hub 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, 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 specify a maximum of 15 accounts or OUs to apply the policy to. To specify a larger number, edit your policy after creation, and apply it to additional accounts.

- 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 API

To create and associate configuration policies

1. Invoke the CreateConfigurationPolicy API from the Security Hub 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 to be enabled or disabled in this configuration policy.
4. For the `EnabledStandardIdentifiers` field, specify which Security Hub 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. Only select controls support custom parameter values. For more information, see Custom control parameters.

7. To apply your configuration policy to accounts or OUs, invoke the StartConfigurationPolicyAssociation API from the Security Hub 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 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 to be enabled or disabled in this configuration policy.
4. For the `EnabledStandardIdentifiers` field, specify which Security Hub 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. Only select controls support custom parameter values. For more information, see Custom control parameters.
7. To apply your configuration policy to accounts or OUs, run the start-configuration-policy-association command from the Security Hub 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 Association status of a configuration.

Viewing Security Hub configuration policies

The delegated administrator account can view AWS Security Hub configuration policies for an organization and their details.

Choose your preferred method, and follow the steps to view your configuration policies.

Console

To view configuration policies

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated administrator account in the home Region.
2. In the navigation pane, choose **Settings** and **Configuration**.
3. Choose the **Policies** tab to view an overview of your configuration policies.
4. Select a configuration policy, and choose **View details** to see additional details about it.

API

To view configuration policies

To view a summary list of all your configuration policies, invoke the `ListConfigurationPolicies` API from the Security Hub delegated administrator account in your home Region. You can provide optional pagination parameters

Example API request:

```
{
```

```
    "MaxResults": 5,
    "NextToken": "U2FsdGVkX19nUI2zoh+Pou9YyutlYJHWpn9xnG4hqSOhvw3o2JqjI23QDxdf"
}
```

To view details about a specific configuration policy, invoke the GetConfigurationPolicy API from the Security Hub delegated administrator account in your home Region. Provide the Amazon Resource Name (ARN) or ID of the configuration policy whose details you want to see.

Example API request:

```
{
    "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 associations, invoke the ListConfigurationPolicyAssociations API from the Security Hub delegated administrator account in your home Region. Optionally, you can provide pagination parameters or filter the results by a specific policy ID, association type, or association status.

Example API request:

```
{
    "AssociationType": "APPLIED"
}
```

To view associations for a specific account, OU, or the root, invoke the GetConfigurationPolicyAssociation or BatchGetConfigurationPolicyAssociations API from the Security Hub delegated administrator account in your home Region. For `Target`, provide the account number, OU ID, or root ID.

```
{
    "Target": {"AccountId": "123456789012"}
}
```

AWS CLI

To view configuration policies

To view a summary list of all your configuration policies, run the `list-configuration-policies` command from the Security Hub delegated administrator account in your home Region.

Example command:

```
aws securityhub --region us-east-1 list-configuration-policies \
--max-items 5 \
--starting-token U2FsdGVkX19nUI2zoh+Pou9YyutlYJHWpn9xnG4hqSOhvw3o2JqjI23QDxdf
```

To view details about a specific configuration policy, run the `get-configuration-policy` command from the Security Hub delegated administrator account in your home Region. Provide the Amazon Resource Name (ARN) or ID of the configuration policy whose details you want to see.

```
aws securityhub --region us-east-1 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, run the `list-configuration-policy-associations` command from the Security Hub delegated administrator account in your home Region. Optionally, you can provide pagination parameters or filter the results by a specific policy ID, association type, or association status.

```
aws securityhub --region us-east-1 list-configuration-policy-associations \
--association-type "APPLIED"
```

To view associations for a specific account, run the `get-configuration-policy-association` or `batch-get-configuration-policy-associations` command from the Security Hub delegated administrator account in your home Region. For `target`, provide the account number, OU ID, or root ID.

```
aws securityhub --region us-east-1 get-configuration-policy-association \
```

```
--target '{"AccountId": "123456789012"}'
```

Association status of a configuration

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. It can take up to 24 hours for the status to change from `PENDING` to `SUCCESS` or `FAILURE`. 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` also depends on all 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 replicates your existing associations that are in a `PENDING`, `SUCCESS`, or `FAILED` state in the new Region.

Common reasons for association failure

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 Security Hub 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 administrator account.
- **Must associate from home Region** – You can only associate a policy with target accounts and OUs when you're signed in to the 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 Security Hub configuration policies

The delegated administrator account can update AWS Security Hub configuration policies as needed. The delegated administrator can update policy settings, the accounts or OUs with which a policy is associated, or both. When policy settings are updated, accounts that are associated with the configuration policy automatically start using the updated policy.

Similar to when you created the configuration policy, you can update the following policy settings:

- Enable or disable Security Hub.

- 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 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 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.

Console

To update configuration policies

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated 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.
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 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 delegated administrator account in the home Region. To remove one or more current associations, invoke the StartConfigurationPolicyDisassociation API from the Security Hub 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 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 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 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 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 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 Association status of a configuration.

Deleting and disassociating Security Hub configuration policies

The delegated administrator account can delete an AWS Security Hub configuration policy. Alternatively, the delegated administrator account can retain the configuration policy, but disassociate it from specific accounts or organizational units (OUs).

The following section explains both of these options.

Deleting 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 Security Hub 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 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 console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated 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 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 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 accounts and OUs

From the delegated administrator account, you can disassociate a target account, OU, or the root from a configuration policy that currently applies to it or from a self-managed configuration. You can disassociate a target only from an applied configuration, not from 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. 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 console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated 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 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 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"}'
```

Central configuration in the context of a standard or control

You can use central configuration from the **Configuration** page of the AWS Security Hub console, or in the context of a specific security standard or security control. Using this feature in context lets you configure standards and controls across your organization in a way that's integrated with existing workflows. In addition, as you view 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 console. Programmatically, you must invoke the UpdateConfigurationPolicy API to change how specific standards or controls are configured in your organization.

Configuring a security standard in context

Follow the steps to configure a security standard in context through central configuration.

To configure a security standard in context (console only)

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated administrator account in the home Region.
2. In the navigation pane, choose **Security standards**.
3. For the standard you want to configure, choose **Configure**. You can also choose a specific standard and then choose **Configure** from the standard details page. The console lists your existing Security Hub configuration policies (configuration policies) and the status of this standard in each one.
4. Choose the options to enable or disable the standard in each configuration policy.
5. After making your changes, choose **Next**.

6. Review your changes, and choose **Apply**. This action affects all accounts and OUs that are associated with a configuration policy. Your configuration takes effect in the home Region and all linked Regions.

Configuring a security control in context

Follow the steps to configure a security control in context through central configuration.

To configure a security control in context (console only)

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated administrator account in the home Region.
2. In the navigation pane, choose **Controls**.
3. Choose a specific control, and then choose **Configure**. The console lists your current configuration policies and the status of this control in each one.
4. Choose the options to enable or disable the control in each configuration policy. You can also choose to customize control parameters.
5. After making your changes, choose **Next**.
6. Review your changes, and choose **Apply**. This action affects all accounts and OUs that are associated with a configuration policy. Your configuration takes effect in the home Region and all linked Regions.

Stop using central configuration

When you stop using central configuration in AWS Security Hub, the delegated administrator loses the ability to configure Security Hub, security standards, and security controls across multiple AWS accounts, organizational units (OUs), and AWS Regions. Instead, organization accounts must configure most of their own settings separately in each Region.

> **Important**
>
> Before you can stop using 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 stop using central configuration, you must also delete your configuration policies.

When you stop 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 settings must be configured separately in each organization account and Region. The delegated administrator can choose to automatically enable Security Hub, 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 console

To stop using central configuration

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated 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 API

To stop using central configuration

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 in new organization accounts, set the `AutoEnable` field to `true`. By default, the value of this field is `false`, and Security Hub 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 stop using central configuration

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 in new organization accounts, include the `auto-enable` parameter. By default, the value of this parameter is `no-auto-enable`, and Security Hub 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

If your AWS environment has multiple accounts, you can treat the accounts that use AWS Security Hub 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 with AWS Organizations or by manually sending and accepting membership invitations in Security Hub.

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, Amazon Macie, and Amazon GuardDuty. For more information, see the *AWS Organizations User Guide*.

When you integrate Security Hub and AWS Organizations, the Organizations management account designates a Security Hub delegated administrator. Security Hub 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 with central configuration. This is the most efficient way to customize Security Hub and ensure adequate security coverage for your organization.

Central configuration lets the delegated administrator customize Security Hub 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 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, called *local configuration*. Under local configuration, the delegated administrator can automatically enable Security Hub 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 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 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 administrator. The administrator account can view data in member accounts and take certain actions on member account findings. The Security Hub 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 by configuring variable settings for different accounts. Instead, each organization account must enable and configure Security Hub 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. 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.

Managing accounts with AWS Organizations

You can integrate AWS Security Hub with AWS Organizations, and then manage Security Hub for accounts in your organization.

To integrate Security Hub with AWS Organizations, you create an organization in AWS Organizations. The Organizations management account designates one account as the Security Hub delegated administrator for the organization. The delegated administrator can then enable Security Hub for other accounts in the organization, add those accounts as Security Hub member accounts, and take allowed actions on the member accounts. The Security Hub delegated administrator can enable and manage Security Hub 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 separately in each member account and AWS Region. The delegated administrator can enforce specific Security Hub settings in specified member accounts and organizational units (OUs) across Regions.

The Security Hub delegated administrator account can perform the following actions on member accounts:

- If using central configuration, centrally configure Security Hub for member accounts and OUs by creating Security Hub configuration policies. Configuration policies can be used to enable and disable Security Hub, enable and disable standards, and enable and disable controls.
- Automatically treat *new* accounts as Security Hub 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 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 with specific centrally managed member accounts.

For a full list of actions that the delegated administrator can perform on member accounts, see Allowed actions for accounts.

The topics in this section explain how to integrate Security Hub with AWS Organizations and how to manage Security Hub for accounts in an organization. Where relevant, each section identifies management benefits and differences for users of central configuration.

Topics

Integrating Security Hub with AWS Organizations

To integrate AWS Security Hub and AWS Organizations, you create an organization in Organizations and use the organization management account to designate a delegated Security Hub administrator account. The delegated administrator can then enable Security Hub 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 configuration policies that specify how the Security Hub 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 administrator

If you have an administrator account in place from the manual invitation process and are transitioning to account management with AWS Organizations, then Security Hub recommends that you designate that account as the delegated Security Hub administrator.

You shouldn't designate the organization management account as the delegated Security Hub administrator. 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 for security management.

We recommend using the same delegated administrator across Regions. If you opt in to central configuration, Security Hub automatically designates the same delegated administrator in your home Region and any linked Regions.

Verify permissions to configure the delegated Security Hub administrator

To designate and remove a delegated Security Hub administrator account, the organization management account must have permissions for the `EnableOrganizationAdminAccount` and `DisableOrganizationAdminAccount` actions in Security Hub. The Organizations management account must also have administrative permissions for Organizations.

To grant all of the required permissions, attach the following Security Hub managed policies to the IAM principal for the organization management account:

- AWSSecurityHubFullAccess
- AWSSecurityHubOrganizationsAccess

Designating the delegated Security Hub administrator

To designate the delegated Security Hub administrator account, you can use the Security Hub console, Security Hub API, or AWS CLI. Security Hub 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 automatically sets the same delegated administrator in the home Region and linked Regions.

The organization management account doesn't have to enable Security Hub in order to designate the delegated Security Hub administrator account.

We recommend that the organization management account is not the delegated Security Hub administrator account. However, if you do choose the organization management account as the Security Hub delegated administrator, the management account must have Security Hub enabled. If the management account does not have Security Hub enabled, you must enable Security Hub for it manually. Security Hub can't be enabled automatically for the organization management account.

> **Note**
>
> You must designate the delegated Security Hub administrator using one of the following methods. Designating the delegated Security Hub administrator with Organizations APIs doesn't reflect in Security Hub.
> Choose your preferred method, and follow the steps to designate the delegated Security Hub administrator account.

Security Hub console

To designate the delegated Security Hub administrator while onboarding

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. Choose **Go to Security Hub**. 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 Security Hub administrator from the Settings page

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. In the Security Hub navigation pane, choose **Settings**. Then choose **General**.
3. If a Security Hub 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** administrator account.

You must designate the same Security Hub 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 API

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

AWS CLI

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

Example command:

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

Removing the delegated Security Hub administrator

Warning

When you use central configuration, you can't use the Security Hub console or Security Hub 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 administrator, Security Hub 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.

Only the organization management account can remove the delegated Security Hub administrator account.

To change the delegated Security Hub administrator, you must first remove the current delegated administrator account and then designate a new one.

If you use the Security Hub console to remove the delegated administrator in one Region, it is automatically removed in all Regions.

The Security Hub API only removes the delegated Security Hub 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 administrator account, it is automatically removed in all Regions.

Removing the delegated Security Hub administrator (Organizations API, AWS CLI)

You can use Organizations to remove the delegated Security Hub 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 administrator account with AWS Organizations.

AWS Organizations API

To remove the delegated Security Hub administrator

Invoke the DeregisterDelegatedAdministrator API. Provide the account ID of the delegated administrator account, and the service principal for Security Hub, which is `securityhub.amazonaws.com`.

AWS CLI

To remove the delegated Security Hub administrator

Run the deregister-delegated-administrator command. Provide the account ID of the delegated administrator account, and the service principal for Security Hub, which is `securityhub.amazonaws.com`.

```
aws organizations deregister-delegated-administrator --account-id <admin account ID>
 --service-principal <Security Hub service principal>
```

Example

```
aws organizations deregister-delegated-administrator --account-id 123456789012 --
service-principal securityhub.amazonaws.com
```

Removing the delegated Security Hub administrator (Security Hub console)

You can use the Security Hub console to remove the delegated Security Hub administrator in all Regions.

When the delegated Security Hub administrator account is removed, the member accounts are disassociated from the removed delegated Security Hub administrator account.

Security Hub is still enabled in the member accounts. They become standalone accounts until a new Security Hub administrator enables them as member accounts.

If the organization management account isn't an enabled account in Security Hub, then use the option on the **Welcome to Security Hub** page.

To remove the delegated Security Hub administrator account from the Welcome to Security Hub page

1. Open the AWS Security Hub 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 administrator account from the Settings page

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. In the Security Hub navigation pane, choose **Settings**. Then choose **General**.
3. Under **Delegated Administrator**, choose **Remove**.

Removing the delegated Security Hub administrator (Security Hub API, AWS CLI)

You can use the Security Hub API or Security Hub operations for the AWS CLI to remove the delegated Security Hub 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 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 administrator account with Security Hub.

Security Hub API

To remove the delegated Security Hub administrator

Using the credentials of the organization management account, invoke the DisableOrganizationAdminAccount API. Provide the account ID of the delegated Security Hub administrator account.

AWS CLI

To remove the delegated Security Hub administrator

Using the credentials of the organization management account, run the disable-organization-admin-account command. Provide the account ID of the delegated Security Hub administrator account.

```
aws securityhub disable-organization-admin-account --admin-account-id <admin account
 ID>
```

Example

```
aws securityhub disable-organization-admin-account --admin-account-id 123456789012
```

Automatically enabling Security Hub 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 console. For organization accounts, **Type** is **By organization**. By default, new accounts don't become Security Hub 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 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 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 in new and existing organization accounts by creating a configuration policy in which Security Hub 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 is enabled with a specific OU, Security Hub 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 enabled. If you associate a configuration policy in which Security Hub is enabled with the root, Security Hub 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 Security Hub configuration policies.

Automatically enabling new organization accounts (local configuration)

When you use local configuration and turn on automatic enablement, Security Hub adds *new* organization accounts as members and enables Security Hub in them in the current Region. Other Regions aren't affected. In addition, turning on automatic enablement doesn't enable Security Hub in *existing* organization accounts unless they were already added as member accounts.

After turning on automatic enablement, default security standards are also enabled automatically for new 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 in new organization accounts. These instructions apply only if you use local configuration.

Security Hub console

To automatically enable new organization accounts as Security Hub members

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign is using the credentials of the delegated administrator account.
2. In the Security Hub navigation pane, under **Settings**, choose **Configuration**.
3. In the **Accounts** section, turn on **Auto-enable accounts**.

Security Hub API

To automatically enable new organization accounts as Security Hub members

Invoke the `UpdateOrganizationConfiguration` API from the delegated administrator account. Set the `AutoEnable` field to `true` to automatically enable Security Hub in new organization accounts.

AWS CLI

To automatically enable new organization accounts as Security Hub members

Run the `update-organization-configuration` command from the delegated administrator account. Include the `auto-enable` parameter to automatically enable Security Hub in new organization accounts.

```
aws securityhub update-organization-configuration --auto-enable
```

Manually enabling Security Hub in new organization accounts

If you don't automatically enable Security Hub in new organization accounts when they join the organization, then you can add those accounts as members and enable Security Hub in them manually after they join the organization. You must also manually enable Security Hub 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 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 in an account if it is already a member account within a different organization.

You also can't enable Security Hub 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 enabled, Security Hub 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 cannot be enabled automatically in the Organizations management account. You must manually enable Security Hub in the Organizations management account before you can add it as a member account.
- If the account already has Security Hub enabled, Security Hub doesn't make any other changes to the account. It only enables the membership.

In order for Security Hub 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 member account.

Security Hub console

To manually enable organization accounts as Security Hub members

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated administrator account.
2. In the Security Hub 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 API

To manually enable organization accounts as Security Hub 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 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 member accounts from your organization

To stop receiving and viewing findings from an AWS Security Hub 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 in one or more centrally managed member accounts. After that, these accounts are still part of the organization, but won't generate Security Hub 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 integration with AWS Organizations.

Choose your preferred method, and follow the steps to disassociate a member account from the organization.

Security Hub console

To disassociate a member account from the organization

1. Open the AWS Security Hub 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 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*.

Disabling Security Hub integration with AWS Organizations

After an AWS Organizations organization is integrated with AWS Security Hub, 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 in AWS Organizations.

When you disable trusted access for Security Hub, the following occurs:

- Security Hub loses its status as a trusted service in AWS Organizations.
- The Security Hub delegated administrator account loses access to Security Hub settings, data, and resources for all Security Hub member accounts in all AWS Regions.

- If you were using central configuration, Security Hub 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 member accounts become standalone accounts and retain their current settings. If Security Hub was enabled for a member account in one or more Regions, Security Hub 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. 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 in member accounts and to clean up Security Hub resources in those accounts.

Choose your preferred method, and follow the steps to disable trusted access for Security Hub.

Organizations console

To disable trusted access for Security Hub

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**.
5. Choose **Disable trusted access**.
6. Confirm that you want to disable trusted access.

Organizations API

To disable trusted access for Security Hub

Invoke the DisableAWSServiceAccess operation of the AWS Organizations API. For the `ServicePrincipal` parameter, specify the Security Hub service principal (`securityhub.amazonaws.com`).

AWS CLI

To disable trusted access for Security Hub

Run the disable-aws-service-access command of the AWS Organizations API. For the `service-principal` parameter, specify the Security Hub service principal (`securityhub.amazonaws.com`).

Example:

```
aws organizations disable-aws-service-access --service-principal
 securityhub.amazonaws.com
```

Managing accounts by invitation

You can centrally manage multiple AWS Security Hub accounts in two ways, by integrating Security Hub with AWS Organizations or by manually sending and accepting membership invitations. You must use the manual process if you have a standalone account or if you don't integrate with Organizations. In manual account management, the Security Hub administrator invites accounts to become members. The administrator-member relationship is established when a prospective member accepts the invitation. A Security Hub administrator account can manage Security Hub for up 1,000 invitation-based member accounts.

Tip

If you create an invitation-based organization in Security Hub, you can subsequently transition to using AWS Organizations instead. If you have more than one member account, we recommend managing accounts through 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 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 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 administrator account. The Security Hub administrator account must send invitations to accounts that belong to other organizations.

On the **Configuration** page of the Security Hub console, accounts that were added by invitation are listed in the **Invitation accounts** tab. If you use Central configuration in Security Hub, but also invite accounts outside of your organization, you can view findings from invitation-based accounts in this tab. However, the Security Hub 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

Your account becomes the AWS Security Hub administrator for accounts that accept your invitation.

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.

Add member accounts

From the Security Hub console, you can add accounts to your list of member accounts. In the Security Hub 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 console

To add accounts to your list of member accounts

1. Open the AWS Security Hub 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 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"}]'
```

Invite 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 console

To invite prospective member accounts

1. Open the AWS Security Hub 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 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 member account

You can accept or decline an invitation to be a member account.

After you accept an invitation, your account becomes an AWS Security Hub member account. The account that sent the invitation becomes your Security Hub administrator account. The administrator account user can view findings for your member account in Security Hub.

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.

Remember that all Security Hub 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.

Accept an invitation

Choose your preferred method, and follow the steps to accept an invitation to be a member account.

Security Hub console

To accept a membership invitation

1. Open the AWS Security Hub 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 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 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`.

Decline an invitation

You can decline an invitation to be a member account. When you decline an invitation in the Security Hub console, your account is marked as **Resigned** on the administrator account's list of member accounts.

When you decline an invitation, you must be signed 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 console

To decline a membership invitation

1. Open the AWS Security Hub 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 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

An AWS Security Hub 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 console

To disassociate a manually-invited member account

1. Open the AWS Security Hub 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 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

As an AWS Security Hub 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 console

To delete a manually-invited member account

1. Open the AWS Security Hub 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 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 your administrator account

If your account was added as a AWS Security Hub member account by invitation, you can disassociate the member account from the administrator account. Once you disassociate a member account, Security Hub 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 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 console

To disassociate from your administrator account

1. Open the AWS Security Hub 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 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 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 AWS Organizations for account management

When you manage accounts manually in AWS Security Hub, you must invite prospective member accounts and configure each member account separately in each AWS Region.

By integrating Security Hub and AWS Organizations, you can eliminate the need to send invitations and gain more control over how Security Hub is configured and customized in your organization.

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 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 with AWS Organizations

First, you must integrate Security Hub 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 delegated administrator account.

> **Note**
>
> The organization management account *cannot* be set as the DA account.

For detailed instructions, see Integrating Security Hub with AWS Organizations.

By completing the preceding steps, you grant trusted access for Security Hub in AWS Organizations. This also enables Security Hub in the current AWS Region for the delegated administrator account.

The delegated administrator can manage the organization in Security Hub, primarily by adding the organization's accounts as Security Hub member accounts. The administrator can also access certain Security Hub settings, data, and resources for those accounts.

When you transition to account management using Organizations, invitation-based accounts don't automatically become Security Hub members. Only the accounts that you add to your new organization can become Security Hub members.

Central configuration vs. local configuration

After activating the integration, you can manage accounts with Organizations. For information, see Managing accounts with AWS Organizations. Account management varies based on your organization's configuration type.

There are two possible configuration types for your organization, local and central. Your default configuration type is *local configuration*. To see your current configuration type, choose **Settings** on the navigation pane of the Security Hub console and then **Configuration**. You can also invoke the `DescribeOrganizationConfiguration` API to view your configuration type.

Under local configuration, the delegated administrator account can choose to automatically enable Security Hub and default security standards in new accounts as they join the organization. These new account settings take effect in the current Region. Other Security Hub settings must be configured separately by each member account in each Region.

We recommend using *central configuration* instead of local configuration. Under central configuration, the delegated administrator account can create Security Hub configuration policies that take effect across multiple Regions and specify Security Hub capabilities in your organization's various accounts and organizational units (OUs). You can apply a single configuration policy to your entire organization, or different configuration policies to different accounts and OUs. For example, you can enable one set of standards and controls in production accounts and a different set of standards and controls in test accounts. The DA can edit configuration policies as needed.

For more information about how central configuration works, see Central configuration in Security Hub.

For instructions on switching from local to central configuration, see Start using central configuration.

Allowed actions for accounts

Administrator and member accounts have access to AWS Security Hub 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 doesn't copy member account findings into the administrator account. In Security Hub, 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.

This table reflects the default permissions for administrator and member accounts. You can use custom IAM policies to further restrict access to Security Hub features and functions. For guidance and examples, see the blog post Aligning IAM policies to user personas for AWS Security Hub.

Allowed actions if you integrate with Organizations and use central configuration

Administrator and member accounts can access Security Hub actions as follows if you integrate with Organizations and use central configuration.

Action	Security Hub delegated administrator account	Centrally managed member account	Self-managed member account
Create and manage Security Hub configuration policies	For self and centrally managed accounts	–	–
View organization accounts	Any	–	–

Action	Security Hub delegated administrator account	Centrally managed member account	Self-managed member account
Disassociate member account	Any	–	–
Delete member account	Any non-organization account	–	–
Disable Security Hub	For current account and centrally managed accounts	–	Current
View findings and finding history	Any	Current	Current
Update findings	Any	Current	Current
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

Action	Security Hub delegated administrator account	Centrally managed member account	Self-managed member account
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 actions as follows if you integrate with Organizations and use local configuration.

Action	Security Hub delegated administrator account	Member account
Create and manage Security Hub configuration policies	–	–
View organization accounts	Any	–
Disassociate member account	Any	–
Delete member account	–	–

Action	Security Hub delegated administrator account	Member account
Disable Security Hub	–	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 and default standards in new organization accounts	For current account and 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 actions as follows if you use the invitation-based method to manually manage accounts instead of integrating with AWS Organizations.

Action	Security Hub administrator account	Member account
Create and manage Security Hub configuration policies	–	–
View organization accounts	Any	–
Disassociate member account	Any	Current
Delete member account	Any	–
Disable Security Hub	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 and default standards in new organization accounts	–	–

Action	Security Hub administrator account	Member account
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

Restrictions and recommendations on account management

The following section summarizes some restrictions and recommendations to keep in mind when managing member accounts in AWS Security Hub.

Maximum number of member accounts

If you use the integration with AWS Organizations, Security Hub supports up to 10,000 member accounts per delegated administrator account in each AWS Region. If you enable and manage Security Hub manually, Security Hub supports up to 1,000 member account invitations per administrator account in each Region.

Accounts and Regions

Membership by organization

If you integrate Security Hub with AWS Organizations, the Organizations management account can designate a delegated administrator (DA) account for Security Hub. The organization management account can't be set as the DA in Organizations. While this is permitted in Security Hub, 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 sets the same DA account in all Regions in which you configure Security Hub 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 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.

Restrictions on administrator-member relationships

> **Note**
>
> If you use the Security Hub integration with AWS Organizations, and haven't manually invited any member accounts, this section doesn't apply to you.

An account cannot 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 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 administrator account for the organization. It also cannot receive invitations from other accounts.

For the manual invitation process, accepting a membership invitation is optional.

Coordinating administrator accounts across services

Security Hub aggregates findings from various AWS services, such as Amazon GuardDuty, Amazon Inspector, and Amazon Macie. Security Hub 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. Security Hub 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, users from account A do *not* automatically see GuardDuty findings for accounts B and C in Security Hub. A Security Hub administrator-member relationship is also required for these accounts.

To do this, make account A the Security Hub administrator account and enable accounts B and C to become Security Hub member accounts.

Effect of account actions on Security Hub data

These account actions have the following effects on AWS Security Hub data.

Security Hub disabled

If you use central configuration, the delegated administrator (DA) can create Security Hub configuration policies that disable AWS Security Hub in specific accounts and organizational units (OUs). In this case, Security Hub is disabled in the specified accounts and OUs in your home Region and any linked Regions.

If don't use central configuration, you must disable Security Hub separately in each account and Region where you enabled it.

No new findings are generated for the administrator account if Security Hub is disabled in the administrator account. You also can't use central configuration if Security Hub is disabled in the DA account. Existing findings are deleted after 90 days.

Integrations with other AWS services are removed.

Enabled security standards and controls are disabled.

Other Security Hub data and settings, including custom actions, insights, and subscriptions to third-party products are retained.

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 is still enabled in both accounts.

If you use central configuration, the DA can't configure Security Hub 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 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 administrator account loses permission to view findings in the member account. However, Security Hub 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 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 administrator account, a removed account has a status of **Deleted**.

Account is suspended

When an account is suspended in AWS, the account loses permission to view their findings in Security Hub. No new findings are generated for that account. The administrator account for a suspended account can view the existing account findings.

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 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 responds to the closure as follows.

Security Hub retains the findings for the account for 90 days from the effective date of the account closure. At the end of the 90 day period, Security Hub permanently deletes all findings for the account.

- To retain findings for more than 90 days, you can use a custom action with an EventBridge rule to store the findings in an Amazon S3 bucket. As long as Security Hub retains the findings, when you reopen the closed account, Security Hub restores the findings for the account.
- If the account is a Security Hub administrator account, it is removed as an administrator 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 administrator account.
- For more information, see Closing an account in the *AWS Billing and Cost Management User Guide*.

> **Important**
>
> For customers in the AWS GovCloud (US) Regions:
>
> - Before closing your account, back up and then delete your policy data and other account resources. You will no longer have access to them after you close the account.

Cross-Region aggregation

With cross-Region aggregation, you can aggregate findings, finding updates, insights, control compliance statuses, and security scores from multiple Regions to a single aggregation Region. You can then manage all of this data from the aggregation Region.

> **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).

Suppose you set US East (N. Virginia) as an aggregation Region, and US West (Oregon) and US West (N. California) as your 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.

The enablement status of a control must be modified in each Region. If a control is enabled in a linked Region but disabled in the aggregation Region, you can see the compliance status of the control from the aggregation Region, but you cannot enable or disable that control from the aggregation Region.

To view cross-Region security scores and compliance statuses, add the following permissions to your IAM role that uses Security Hub:

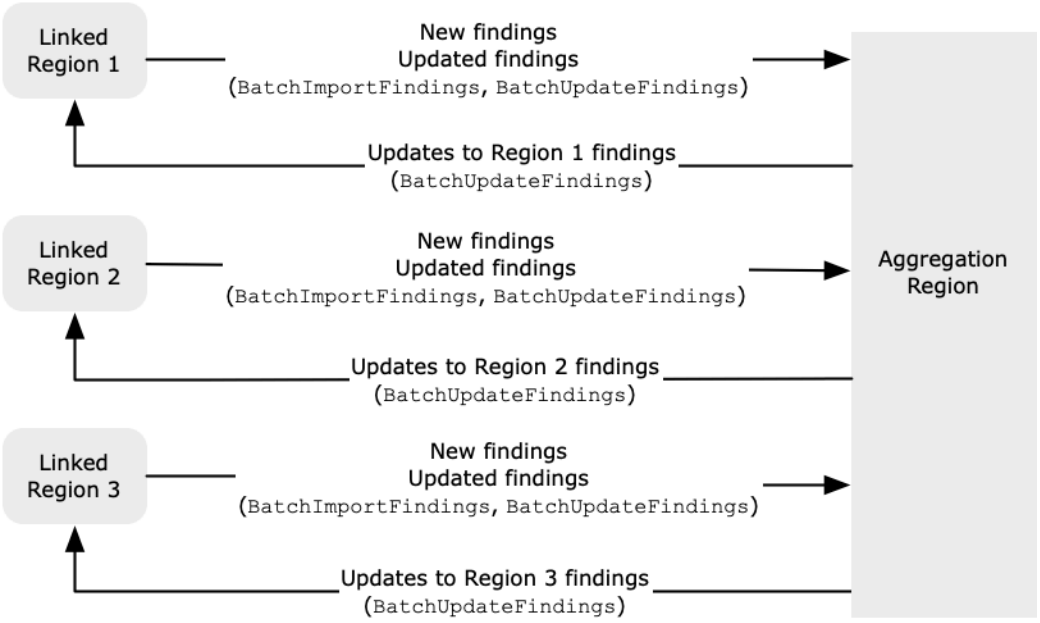
- `ListSecurityControlDefinitions`
- `BatchGetStandardsControlAssociations`
- `BatchUpdateStandardsControlAssociations`

How cross-Region aggregation works

When cross-Region aggregation is enabled, Security Hub replicates the following data from the linked Regions to the aggregation 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 also replicates updates to this data between the linked Regions and the aggregation Region. Updates that occur in a linked Region are replicated to the aggregation Region. Updates that occur in the aggregation Region are replicated back to the linked Region.



If there are conflicting updates in the aggregation Region and the linked Region, then the most recent update is used.

Cross-Region aggregation does not add to the cost of Security Hub. You are not charged when Security Hub replicates new data or updates.

In the aggregation 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 aggregation 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 aggregation 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 aggregation Region. The number of security checks includes findings from all linked Regions.

Security Hub only aggregates data from Regions where an account has Security Hub enabled. Security Hub is not automatically enabled for an account based on the cross-Region aggregation configuration.

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 replicates the data that the administrator account generates in all linked Regions to the aggregation Region. In addition, Security Hub 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 replicates the data that a member account generates in all linked Regions to the aggregation Region.

The administrator can access and manage security findings from all member accounts within the administered regions. However, as a Security Hub administrator, you must be signed in to the aggregation Region to view aggregated data from all member accounts and linked Regions.

As a Security Hub member account, you must be signed in to the aggregation 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 aggregation Region and all linked Regions in order for cross-Region aggregation to work. In addition, the member account must have Security Hub enabled in the aggregation Region and all linked Regions to give the administrator the ability to view findings from the member account. If you don't use the aggregation Region for other purposes, you can disable Security Hub 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 aggregation Region.

> **Note**
>
> To understand how cross-Region aggregation impacts central configuration, see Central configuration and cross-Region aggregation.

Central configuration and cross-Region aggregation

Central configuration is an opt-in feature in Security Hub that you can use if you integrate with AWS Organizations. If you use central configuration, the delegated administrator account can configure the Security Hub service, standards, and controls for accounts and organizational units (OU) in the organization. To configure accounts and OUs, the delegated administrator creates Security Hub configuration policies. Configuration policies can be used to define whether Security Hub 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 aggregation Region. In addition, configuration policies take effect in the aggregation Region and all linked Regions. You can't create a configuration policy that applies only in some linked Regions and not others. In central configuration, the aggregation Region is called the *home Region*. The same Region must serve as the home Region for purposes of central configuration and as the aggregation Region for purposes of cross-Region aggregation. For information about cross-Region aggregation, see Cross-Region aggregation.

To use central configuration, you must designate a home Region and at least one linked Region.

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 Central configuration in Security Hub.

Enabling cross-Region aggregation

You must enable cross-Region aggregation from the AWS Region that you want to designate as the aggregation Region.

You cannot use a Region that is disabled by default as your aggregation Region. For a list of Regions that are disabled by default, see Enabling a Region in the *AWS General Reference*.

Enabling cross-Region aggregation (console)

When you enable cross-Region aggregation, you choose your linked Regions. You also choose whether to automatically link new Regions when Security Hub begins to support them and you have opted into them.

To enable cross-Region aggregation

1. Open the AWS Security Hub 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 navigation menu, choose **Settings** and then **Regions**.
4. For **Finding aggregation**, choose **Configure finding aggregation**.

   By default, the aggregation Region is set to **No aggregation Region**.
5. Under **Aggregation Region**, select the option to designate the current Region as the aggregation 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 supports them and you opt into them, select **Link future Regions**.
8. Choose **Save**.

Enabling cross-Region aggregation (Security Hub API, AWS CLI)

You can use the Security Hub API to enable cross-Region aggregation.

To enable cross-Region aggregation from the Security Hub API, you create a finding aggregator. You must create the finding aggregator from the Region that you want to use as the aggregation Region.

To create the finding aggregator (Security Hub API, AWS CLI)

- **Security Hub API:** From the Region that you want to use as the aggregation Region, use the `CreateFindingAggregator` operation. For `RegionLinkingMode`, you choose from the following options:
  - `ALL_REGIONS` – Security Hub aggregates data from all Regions. Security Hub also aggregates data from new Regions as they are supported and you opt into them.
  - `ALL_REGIONS_EXCEPT_SPECIFIED` – Security Hub aggregates data from all Regions except for Regions that you want to exclude. Security Hub 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 aggregates data from a selected list of Regions. Security Hub does not aggregate data automatically from new Regions. Use `Regions` to provide the list of Regions to aggregate from.

- **AWS CLI:** At the command line, run the `create-finding-aggregator` command. Separate each Region with a space.

```
aws securityhub create-finding-aggregator --region <aggregation Region> --region-
linking-mode ALL_REGIONS | ALL_REGIONS_EXCEPT_SPECIFIED | SPECIFIED_REGIONS --
regions <Region list>
```

  In the following example, cross-Region aggregation is configured for selected Regions. The aggregation Region is US East (N. Virginia). The linked Regions are US West (N. California) and US West (Oregon).

```
aws securityhub create-finding-aggregator --region us-east-1 --region-linking-mode
 SPECIFIED_REGIONS --regions us-west-1 us-west-2
```

Viewing cross-Region aggregation settings

You can view the current cross-Region aggregation configuration from any Region. The configuration includes the aggregation Region, the linked Regions, and whether to automatically link new Regions.

Viewing the cross-Region aggregation configuration (console)

The **Regions** tab of the **Settings** page displays the current cross-Region aggregation configuration. You can view the configuration from any Region. Member accounts can also view the cross-Region configuration that the administrator account configured.

If cross-Region aggregation is not enabled, then the **Regions** tab displays the option to enable cross-Region aggregation. See the section called “Enabling 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 aggregation Region
- Whether to automatically aggregate findings, insights, control statuses, and security scores from new Regions that Security Hub supports and that you opt into

- The list of linked Regions

Viewing the current cross-Region aggregation configuration (Security Hub API, AWS CLI)

You can use the Security Hub API or AWS CLI to view the current cross-Region aggregation configuration. You can view the cross-Region aggregation configuration from any Region.

To view the current cross-Region aggregation configuration (Security Hub API, AWS CLI)

- **Security Hub API:** Use the `GetFindingAggregator` API. When you make the request, you must provide the finding aggregator ARN. To obtain the finding aggregator ARN, use `ListFindingAggregators`.
- **AWS CLI:** At the command line, run the `get-finding-aggregator` command. To obtain the finding aggregator ARN, use `list-finding-aggregators`.

```
aws securityhub get-finding-aggregator --finding-aggregator-arn <finding aggregator
 ARN>
```

Updating the cross-Region aggregation configuration

You can update the cross-Region aggregation configuration to change the linked AWS Regions for the current aggregation Region. You can also change whether to automatically aggregate findings, insights, control statuses, and security scores from new Regions.

Changes to cross-Region aggregation aren't implemented for an opt-in Region until the Region is enabled in an 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, Security Hub does not remove any existing aggregated data from the aggregation Region.

You cannot use the update process to change the aggregation Region. To change the aggregation Region, you must do the following:

1. Stop cross-Region aggregation. See the section called “Stopping cross-Region aggregation”.
2. Change to the Region that you want to be the new aggregation Region.

3. Enable cross-Region aggregation. See the section called “Enabling cross-Region aggregation”.

Updating the cross-Region aggregation configuration (console)

You must update the cross-Region aggregation configuration from the current aggregation Region.

In AWS Regions other than the aggregation Region, the **Finding aggregation** panel displays a message that you must edit the configuration in the aggregation Region. Choose this message to display a link to navigate to the aggregation Region.

To change the linked Regions for the current aggregation Region

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. Change to the current aggregation Region.
3. In the Security Hub navigation menu, choose **Settings**, then choose **Regions**.
4. Under **Finding aggregation**, choose **Edit**.
5. Under **Linked Regions**, update the selected linked Regions.
6. If needed, change whether **Link future Regions** is selected. This setting determines whether Security Hub automatically links new Regions as it adds support for them and you opt into them.
7. Choose **Save**.

Updating the cross-Region aggregation configuration (Security Hub API, AWS CLI)

You can use the Security Hub API or AWS CLI to update the cross-Region aggregation configuration. You must update cross-Region aggregation from the current aggregation Region.

You can change the Region linking mode. If the linking mode is `ALL_REGIONS_EXCEPT_SPECIFIED` or `SPECIFIED_REGIONS`, you can change the list of excluded or included Regions.

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). When you call `UpdateFindingAggregator`, you provide a `Regions` list that contains both US East (Ohio) and US West (Oregon).

To update cross-Region aggregation (Security Hub API, AWS CLI)

- **Security Hub API:** Use the `UpdateFindingAggregator` API operation. To identify the finding aggregator, you must provide the finding aggregator ARN. To obtain the finding aggregator ARN, use `ListFindingAggregators`.

  You provide the Region linking mode and the updated list of excluded or included Regions.
- **AWS CLI:** At the command line, run the `update-finding-aggregator` command. Separate each Region with a space.

  ```
  aws securityhub update-finding-aggregator --region <aggregation Region> --finding-
  aggregator-arn <finding aggregator ARN> --region-linking-mode ALL_REGIONS |
   ALL_REGIONS_EXCEPT_SPECIFIED | SPECIFIED_REGIONS --regions <Region list>
  ```

  In the following example, the cross-Region aggregation configuration is changed to aggregation for selected Regions. The command is run from the current aggregation Region, which is US East (N. Virginia). The linked Regions are US West (N. California) and US West (Oregon).

  ```
  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

Stop cross-Region aggregation if you no longer want to aggregate data or if you want to change the aggregation Region.

When you stop cross-Region aggregation, Security Hub stops aggregating data. It does not remove any existing aggregated data from the aggregation Region.

Stopping cross-Region aggregation (console)

You must stop cross-Region aggregation from the current aggregation Region.

In Regions other than the aggregation Region, the **Finding aggregation** panel displays a message that you must edit the configuration in the aggregation Region. Choose this message to display a link to switch to the aggregation Region.

To stop cross-Region aggregation

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. Change to the current aggregation Region.
3. In the Security Hub 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**.

Stopping cross-Region aggregation (Security Hub API, AWS CLI)

You can use the Security Hub API to stop cross-Region aggregation. You must stop cross-Region aggregation from the aggregation Region.

To stop cross-Region aggregation (Security Hub API, AWS CLI)

- **Security Hub API:** Use the `DeleteFindingAggregator` operation. To identify the finding aggregator to delete, you provide the finding aggregator ARN. To obtain the finding aggregator ARN, use `ListFindingAggregators`.
- **AWS CLI:** At the command line, run the `delete-finding-aggregator` command.

```
aws securityhub delete-finding-aggregator <finding aggregator ARN> --
region <aggregation Region>
```

Findings in AWS Security Hub

AWS Security Hub eliminates the complexity of addressing large volumes of findings from multiple providers. It reduces the effort required to manage and improve the security of all of your AWS accounts, resources, and workloads.

Security Hub receives findings from the following sources.

- Security Hub checks against enabled controls. See the section called "Generating and updating control findings".
- Integrations with AWS services that you enable. See the section called "AWS service integrations".
- Integrations with third-party products that you enable. See the section called "Third-party product integrations".
- Custom integrations that you configure. See the section called "Using custom product integrations".

Security Hub consumes findings using a standard findings format called the AWS Security Finding Format. For more information about the finding format, see the section called "Finding format".

Security Hub correlates the findings across integrated products to prioritize the most important ones.

Finding providers can update findings to reflect additional instances of the finding. You can update findings to provide details about your investigation and its results.

Security Hub also allows you to aggregate findings across Regions, so that you can view all of your findings from one place. See *Cross-Region aggregation*.

Topics

Creating and updating findings in AWS Security Hub

In AWS Security Hub, a finding can originate from one of the following types of finding providers.

- An enabled security control in Security Hub
- An enabled integration with another AWS service
- An enabled integration with a third-party product

After a finding is created, it can be updated by the finding provider or by the customer.

- The finding provider uses the `BatchImportFindings` API operation to update the general information about a finding. Finding providers can only update findings that they created.
- The customer uses the `BatchUpdateFindings` API operation to update the status of the investigation into a finding. `BatchUpdateFindings` can also be used by a ticketing, incident management, orchestration, remediation, or SIEM tool on behalf of the customer.

  From the Security Hub console, customers can manage the workflow status of findings and send findings to custom actions. See the section called “Taking action on findings”.

Security Hub also automatically updates and deletes findings. All findings are automatically deleted if they were not updated in the past 90 days.

If you enable cross-Region aggregation, then Security Hub automatically aggregates new findings from the linked Regions to the aggregation Region. Security Hub also replicates updates to findings. Updates that occur in the linked Regions are replicated to the aggregation Region. Updates that occur in the aggregation Region are replicated to the linked Region. For more information about cross-Region aggregation, see *Cross-Region aggregation*.

Topics

Using BatchImportFindings to create and update findings

Finding providers use the `BatchImportFindings` API operation to create new findings and to update information about the findings they created. They cannot update findings that they did not create.

Customers, SIEMs, ticketing tools, and SOAR tools use `BatchUpdateFindings` to make updates related to their investigation of findings from finding providers. See the section called "Using BatchUpdateFindings".

Whenever AWS Security Hub receives a `BatchImportFindings` request to either create or update a finding, it automatically generates a **Security Hub Findings - Imported** event in Amazon EventBridge. See the section called "Automated response and remediation".

Requirements for accounts and batch size

`BatchImportFindings` must be called by one of the following:

- The account that is associated with the findings. The identifier of the associated account is the value of the `AwsAccountId` attribute for the finding.
- An account that is allow-listed for an official Security Hub partner integration.

Security Hub can only accept finding updates for accounts that have Security Hub enabled. The finding provider also must be enabled. If Security Hub is disabled, or the finding provider integration is not enabled, then the findings are returned in the `FailedFindings` list, with an `InvalidAccess` error.

`BatchImportFindings` accepts up to 100 findings per batch, up to 240 KB per finding, and up to 6 MB per batch. The throttle rate limit is 10 TPS per account per Region, with a burst of 30 TPS.

Determining whether to create or update a finding

To determine whether to create or update a finding, Security Hub checks the `ID` field. If the value of `ID` does not match an existing finding, then a new finding is created.

If `ID` does match an existing finding, then Security Hub checks the `UpdatedAt` field for the update.

- If `UpdatedAt` on the update matches or occurs before `UpdatedAt` on the existing finding, then the update is ignored.

- If `UpdatedAt` on the update occurs after `UpdatedAt` on the existing finding, then the existing finding is updated.

Restricted attributes for BatchImportFindings

For an existing finding, finding providers can't use `BatchImportFindings` to update the following attributes and objects. These attributes can only be updated using `BatchUpdateFindings`.

- `Note`
- `UserDefinedFields`
- `VerificationState`
- `Workflow`

Security Hub ignores any content provided in a `BatchImportFindings` request for those attributes and objects. Customers, or other providers acting on their behalf, use `BatchUpdateFindings` to update them.

Using FindingProviderFields

Finding providers also shouldn't use `BatchImportFindings` to update the following attributes.

- `Confidence`
- `Criticality`
- `RelatedFindings`
- `Severity`
- `Types`

Instead, finding providers 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 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 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 uses the value to update the attribute in `FindingProviderFields`. It overwrites any existing value.

Security Hub 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 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 uses both values. However, it updates the top-level attribute value only if the attribute was not already updated by `BatchUpdateFindings`.

Using the batch-import-findings command from the AWS CLI

In the AWS Command Line Interface, you use the `batch-import-findings` command to create or update findings.

You provide each finding as a JSON object.

Example

```
aws securityhub batch-import-findings --findings
    [{
        "AwsAccountId": "123456789012",
        "CreatedAt": "2019-08-07T17:05:54.832Z",
        "Description": "Vulnerability in a CloudTrail trail",
        "GeneratorId": "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0/rule/2.2",
        "Id": "Id1",
        "ProductArn": "arn:aws:securityhub:us-west-1:123456789012:product/123456789012/
default",
        "Resources": [
            {
                "Id": "arn:aws:cloudtrail:us-west-1:123456789012:trail/TrailName",
                "Partition": "aws",
                "Region": "us-west-1",
                "Type": "AwsCloudTrailTrail"
            }
        ],
        "SchemaVersion": "2018-10-08",
        "Title": "CloudTrail trail vulnerability",
```

```
        "UpdatedAt": "2020-06-02T16:05:54.832Z",
        "Types": [
            "Software and Configuration Checks/Vulnerabilities/CVE"
        ],
        "Severity": {
            "Label": "INFORMATIONAL",
            "Original": "0"
        }
    }]'
```

Using BatchUpdateFindings to update a finding

The `BatchUpdateFindings` action is used to update information related to a customer's processing of findings from finding providers. It can be used by a customer or by a SIEM, ticketing, incident management, or SOAR tool that works on behalf of a customer. You can use `BatchUpdateFindings` to update specific fields in the AWS Security Finding Format (ASFF).

You can't use `BatchUpdateFindings` to create new findings. You can use it to update up to 100 findings at a time.

Whenever Security Hub receives a `BatchUpdateFindings` request to update a finding, it automatically generates a **Security Hub Findings - Imported** event in Amazon EventBridge. See the section called “Automated response and remediation”.

`BatchUpdateFindings` doesn't change the `UpdatedAt` field for the finding. `UpdatedAt` only reflects the most recent update from the finding provider.

Available fields for BatchUpdateFindings

Administrator accounts can use >`BatchUpdateFindings` to update findings for their account or for their member accounts. Member accounts can use >`BatchUpdateFindings` to update findings for their account.

Customers can only use >`BatchUpdateFindings` to update the following fields and objects.

- `Confidence`
- `Criticality`
- `Note`
- `RelatedFindings`
- `Severity`

- `Types`
- `UserDefinedFields`
- `VerificationState`
- `Workflow`

By default, administrator and member accounts have access to all of the above fields and field values. Security Hub also provides context keys to allow you to restrict access to fields and field values.

For example, you might only allow member accounts to set `Workflow.Status` to `RESOLVED`. Or you might not want to allow member accounts to change `Severity.Label`.

Configuring access to BatchUpdateFindings

You can configure IAM policies to restrict access to using `BatchUpdateFindings` to update 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.

```
{
    "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"
        ]
    }
}
```

Using the batch-update-findings command from the AWS CLI

In the AWS Command Line Interface, you use the `batch-update-findings` command to update the findings.

For each finding to update, you provide both the finding ID and the ARN of the product that generated the finding.

```
--finding-identifiers ID="<findingID1>",ProductArn="<productARN>"
 ID="<findingID2>",ProductArn="<productARN2>"
```

When you provide the attributes to update, you can either use a JSON format or a shortcut format.

Here is an example of an update to the `Note` object that uses the JSON format:

```
--note '{"Text": "Known issue that is not a risk.", "UpdatedBy": "user1"}'
```

Here is the same update that uses the shortcut format:

```
--note Text="Known issue that is not a risk.",UpdatedBy="user1"
```

The AWS CLI Command Reference provides the JSON and shortcut syntax for each field.

The following >`batch-update-findings` example updates two findings to add a note, change the severity label, and resolve them.

```
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-2::product/aws/securityhub" Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" --note '{"Text": "Known issue that is not a
 risk.", "UpdatedBy": "user1"}' --severity '{"Label": "LOW"}' --workflow '{"Status":
 "RESOLVED"}'
```

This is the same example, but uses the shortcuts instead of JSON.

```
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" Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" --note Text="Known issue that is not a
 risk.",UpdatedBy="user1" --severity Label="LOW" --workflow Status="RESOLVED"
```

Managing and reviewing finding details and history

There are multiple ways to view finding lists on the AWS Security Hub console:

- **Findings page** – Displays a comprehensive list of findings from all enabled controls and product integrations. By default, active findings with a `NEW` or `NOTIFIED` workflow status are shown.
- **Control details page** – Displays a list of findings that were generated in the last 24 hours for a specific control.
- **Insights page** – Displays a list of findings for a matching insight. An insight is a collection specific findings. For more information, see the section called “Viewing insight results and findings”.
- **Integrations page** – Displays a list of findings generated by an integrated AWS service or third-party product.

You can filter and group findings on these lists to focus on specific types of findings. You can also select a specific finding on the preceding pages to view details about it.

To view a list of findings programmatically, use the `GetFindings` operation of the Security Hub API. You can include filters to retrieve specific types of findings.

If you enable cross-Region aggregation, you can retrieve control statuses, security scores, insights, and findings from across Regions. In the aggregation Region, finding data includes data from the aggregation Region and the linked Regions. In other Regions, finding data is specific to that Region only. For information about configuring cross-Region aggregation, see *Cross-Region aggregation*.

Filtering and grouping findings (console)

When you display a list of findings on the **Findings** page, **Integrations** page, or **Insights** page of the Security Hub console, the list is pre-filtered based on the record state and workflow status. This is in addition to the filters for an insight or integration.

Record state indicates whether a finding is active or archived. By default, a finding list only shows active findings.A finding can be archived by the finding provider. AWS Security Hub 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.

If you enabled finding aggregation and are signed in to the aggregation Region, you can filter findings by Region on the **Findings** and **Insights** pages.

For information about working with control findings, see the section called "Filtering and sorting findings". The information on this page applies to finding lists on the **Findings**, **Insights**, and **Integrations** pages.

Adding filters

To change the scope of the list, you can add filters to it.

You can filter by up to 10 attributes. For each attribute, you can provide up to 20 filter values.

When filtering the finding list, Security Hub applies AND logic to the set of filters. In other words, a finding only matches 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, then matching findings must match both of these criteria.

However, Security Hub applies OR logic to filters that use the same attribute but different values. For example, you add both GuardDuty and Amazon Inspector as filter values for product name. In that case, a finding matches if it was generated by either GuardDuty or Amazon Inspector.

To add a filter to the finding list

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. To display a finding list, do one of the following:
   - In the Security Hub navigation pane, choose **Findings**.
   - In the Security Hub navigation pane, choose **Insights**. Choose an insight. Then on the results list, choose an insight result.
   - In the Security Hub navigation pane, choose **Integrations**. Choose **See findings** for an integration.
3. In the **Add filters** box, for **Filters**, choose a filter.

   When you filter by **Company name** or **Product name**, the console uses the top-level `CompanyName` and `ProductName` fields. The API uses the values that are in `ProductFields`.
4. Choose the filter match type.

   For a string filter, you can choose from the following comparison 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 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.

   For example, for findings from Security Hub, **Product name** is Security Hub. If you use the **EQUALS** operator to see findings from Security Hub, you must enter **`Security Hub`** as the filter value. If you enter **`security hub`**, no findings are displayed.

   Similarly, if you use the **PREFIX** operator, and enter **`Sec`**, Security Hub findings are displayed. If you enter **`sec`**, no Security Hub findings are displayed.

6. Choose **Apply**.

Grouping findings

In addition to changing the filters, you can group the findings 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 findings that match the other filter criteria.

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.

Note that Security Hub can only display 100 values. If there are more than 100 grouping values, you only see the first 100.

When you choose an attribute value, the list of matching findings for that value is displayed.

To group the findings in a findings list

1. On the finding list, choose the **Add filters** box.
2. For **Grouping**, choose **Group by**.
3. In the list, choose the attribute to use for the grouping.
4. Choose **Apply**.

Changing a filter value or grouping attribute

For an existing filter, you can change the filter value. You can also change the grouping attribute.

For example, you can change the **Record state** filter to look for `ARCHIVED` findings instead of `ACTIVE` findings.

To edit a filter or grouping attribute

1. On a filtered finding list, choose the filter or grouping attribute.
2. For **Group by**, choose the new attribute, then choose **Apply**.
3. For a filter, choose the new value, and then choose **Apply**.

Deleting a filter or grouping attribute

To delete a filter or grouping attribute, choose the **x** icon.

The list is updated automatically to reflect the change. When you remove the grouping attribute, the list changes from the list of field values back to a list of findings.

Available finding information

You can get a variety of findings details on the Security Hub console or by calling the GetFindings operation of the Security Hub API. Here is a partial list of the types of finding details you can get.

- **Application metadata** – Provides the name and Amazon Resource Name (ARN) of the application involved in a finding if you created an application. and added the AWS application tag to it. We recommend creating applications in AWS Service Catalog AppRegistry.
- **Finding history** – Provides the history of the finding in the last 90 days.
- **Finding investigation in Detective (console only)** – Provides a link to further investigate a finding in Detective using using automated log collection, security analytics, and AWS service resource exploration tools. This information is only included for Security Hub findings received from other AWS services if you enable Detective.
- **Finding provider fields** – displays the values from the finding provider for confidence, criticality, related findings, severity, and finding type.
- **Parameters** – Shows the current parameter values for a security control. Security Hub uses these parameter values when conducting security checks of the control.
- **Remediation** – Provides a link to the instructions for remediating failed control findings.
- **Resource** – Provides information about the AWS resource involved in a finding.
- **Resource tags** – Provides tag key and value information for the resources involved in a finding. You can tag resources that are supported by the `GetResources` operation of the AWS Resource Groups Tagging API. For more information about the inclusion of resource tags in findings, see Tags.
- **Types and related findings** – Contains information about the finding type.
- **Vulnerability details** – Information about a vulnerability that's detected in a finding and affected packages. These details are available if you enable Amazon Inspector for findings that Amazon Inspector sends to Security Hub.

Review the following sections to understand how to access these details for a finding.

Reviewing finding history

Finding history is a Security Hub feature that lets you track changes made to a finding during the last 90 days. It's available for active and archived findings. Finding history provides an immutable trail of changes made to a finding over time, including what the change was, when it occurred, and by which user.

In particular, you can track changes made to fields in the AWS Security Finding Format (ASFF). Security Hub tracks changes that you make manually and with automation rules.

Finding history is available in the Security Hub console, API, and AWS CLI.

If you're signed in to a Security Hub administrator account, you can get finding history for the administrator account and all member accounts.

Choose your preferred method, and follow the steps to get finding history.

Security Hub console

Reviewing finding history

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. In the left navigation pane, choose **Findings**.
3. Select a finding. In the panel that appears, choose the **History** tab.

Security Hub API

Reviewing finding history

1. Run GetFindings, or if you're using the AWS CLI, run the get-findings command. using appropriate filters as needed, to identify the finding that you want to view history for. The API response will give you the `ProductArn` and `Id` for the finding. You need the values for these fields in the third step.
2. Run GetFindingHistory, or if you're using the AWS CLI, run the get-finding-history command.
3. Identify the finding that you want to get history for with the `ProductArn` and `Id` fields. For more information about these fields, see AwsSecurityFindingIdentifier. You can only get history for one finding per request.

4. Provide values for `StartTime`. and `EndTime` to limit finding history to a specific period of time.
5. Provide a value for `MaxResults` to limit finding history to a specific number of results. If not provided, the API response returns the first 100 results of finding history.
6. Provide a value for `NextToken` to view the next 100 results (if applicable) for a finding. In your initial API request, the value of `NextToken` should be `NULL`.

The following CLI command retrieves 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"
```

Reviewing finding details

Follow the steps to view finding details on the Security Hub console.

Security Hub console

Reviewing finding details

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. To display a finding list, take one of the following actions:
   - In the Security Hub navigation pane, choose **Findings**. Add search filters as necessary to narrow down the finding list.
   - In the Security Hub navigation pane, choose **Insights**. Choose an insight. Then on the results list, choose an insight result.
   - In the Security Hub navigation pane, choose **Integrations**. Choose **See findings** for an integration.

3. Select a finding title.
4. From the finding details panel, you can take additional actions as follows:

   - To display the complete JSON for the finding, choose the finding ID. From **Finding JSON**, download the finding JSON.
   - For findings that are based on AWS Config rules, to display a list of the applicable rules, choose **Rules**.
   - Choose **Investigate with Macie** to investigate sensitive data that's discovered in the finding in the Macie console. This option is only available if you enable Amazon Macie and its automated sensitive data discovery feature.
   - Choose **Resources** to view information about the resource involved in a finding.
   - Choose **Investigate in Amazon Detective** to investigate the finding in the Detective console. This option is only available if you enable Amazon Detective.
   - Choose the **History** tab to view up to 90 days of finding history.

> **Note**
>
> The top of the finding details panel contains overview information about the finding, including the account, severity, dates, and status. If you integrate with AWS Organizations and the account you're signed in to is an organization member account, then the details panel includes the account name. For member accounts that are invited manually rather than through the Organizations integration, the details panel only includes the account ID.

Security Hub API

Reviewing finding details

Use the `GetFindings` operation of the Security Hub 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 that you want 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.

If you've enabled cross-Region aggregation and invoke this operation from the aggregation Region, the results include findings from the aggregation and linked Regions.

The following CLI command retrieves the findings that match the provided filters and sorts them in descending order of 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
```

PowerShell

Reviewing finding details

1. Use the `Get-SHUBFinding` cmdlet.
2. Optionally, populate the `Filter` parameter to narrow the findings that you want to retrieve.

Example

```
Get-SHUBFinding -Filter @{AwsAccountId =
 [Amazon.SecurityHub.Model.StringFilter]@{Comparison = "EQUALS"; Value =
 "XXX"};ComplianceStatus = [Amazon.SecurityHub.Model.StringFilter]@{Comparison =
 "EQUALS"; Value = 'FAILED'}}
```

Note

When you filter findings by `CompanyName` or `ProductName`, Security Hub uses the values that are part of the `ProductFields` ASFF object. Security Hub doesn't use the top-level `CompanyName` and `ProductName` fields.

Taking action on findings in AWS Security Hub

AWS Security Hub allows you to track the current status of your investigation into a finding.

You can also send findings to custom actions for processing.

Topics

Setting the workflow status of findings

Workflow status tracks the progress of your investigation into a finding. The workflow status is specific to an individual finding. It doesn't affect the generation of new findings. For example, setting the workflow status of a finding to SUPPRESSED or RESOLVED doesn't prevent AWS Security Hub from generating a new finding for the same issue.

Workflow status can have 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 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`, then Security Hub automatically sets the workflow status to `RESOLVED`.

Setting the workflow status of findings

Choose your preferred method, and follow the steps to set the workflow status of one or more findings.

To automatically update the workflow status of specific findings, see Automation rules.

Security Hub console

To set the workflow status of findings

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. To display a finding list, do one of the following:
   - In the Security Hub navigation pane, choose **Findings**.
   - In the Security Hub navigation pane, choose **Insights**. Choose an insight. Then on the results list, choose an insight result.

- In the Security Hub navigation pane, choose **Integrations**. Choose **See findings** for an integration.
- In the Security Hub navigation pane, choose **Security standards**. Choose **View results** to display a list of controls. Then, select a control to see a list of findings for that control.

3. In the finding list, select the check box for each finding that you want to update.
4. At the top of the list, for **Workflow status**, choose the status.

Security Hub API

Invoke the `BatchUpdateFindings` API. Provide both the finding ID and the ARN of the product that generated the finding. You can get these details by invoking the `GetFindings` API.

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 action

You can create AWS Security Hub custom actions to automate Security Hub with Amazon EventBridge. For custom actions, the event type is **Security Hub Findings - Custom Action**.

For more information and detailed steps on creating custom actions, see the section called "Automated response and remediation".

After you set up a custom action, you can send findings to it.

To send findings to a custom action (console)

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. To display a finding list, do one of the following:
   - In the Security Hub navigation pane, choose **Findings**.
   - In the Security Hub navigation pane, choose **Insights**. Choose an insight. Then on the results list, choose an insight result.
   - In the Security Hub navigation pane, choose **Integrations**. Choose **See findings** for an integration.
   - In the Security Hub 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 consumes, aggregates, organizes, and prioritizes findings from AWS security services and from the third-party product integrations. Security Hub 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. Then it correlates ingested findings across products to prioritize the most important ones.

Topics

AWS Security Finding Format (ASFF) syntax

This page provides a complete outline of the JSON for a finding in the AWS Security Finding Format (ASFF). The format is derived from JSON Schema. Choose a linked object name to view an

example finding for that object. You can compare your Security Hub findings with the resources and examples shown here to help you interpret your findings.

To view descriptions of the required ASFF attributes, see the section called "Required top-level attributes".

To view descriptions of the other top-level ASFF attributes, see the section called "Optional top-level 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",
    "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"
      },
      "AwsEc2VpnConnection": {
       "Category": "string",
       "CustomerGatewayConfiguration": "string",
       "CustomerGatewayId": "string",
       "Options": {
        "StaticRoutesOnly": boolean,
        "TunnelOptions": [{
         "DpdTimeoutSeconds": number,
         "IkeVersions": ["string"],
         "OutsideIpAddress": "string",
         "Phase1DhGroupNumbers": [number],
         "Phase1EncryptionAlgorithms": ["string"],
         "Phase1IntegrityAlgorithms": ["string"],
         "Phase1LifetimeSeconds": number,
         "Phase2DhGroupNumbers": [number],
         "Phase2EncryptionAlgorithms": ["string"],
         "Phase2IntegrityAlgorithms": ["string"],
         "Phase2LifetimeSeconds": number,
         "PreSharedKey": "string",
         "RekeyFuzzPercentage": number,
         "RekeyMarginTimeSeconds": number,
         "ReplayWindowSize": number,
         "TunnelInsideCidr": "string"
        }]
       },
```

```
        "Routes": [{
         "DestinationCidrBlock": "string",
         "State": "string"
        }],
        "State": "string",
        "TransitGatewayId": "string",
        "Type": "string",
        "VgwTelemetry": [{
         "AcceptedRouteCount": number,
         "CertificateArn": "string",
         "LastStatusChange": "string",
         "OutsideIpAddress": "string",
         "Status": "string",
         "StatusMessage": "string"
        }],
        "VpnConnectionId": "string",
        "VpnGatewayId": "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"
      },
      "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

Security Hub offers two types of consolidation:

- **Consolidated controls view (always on; can't be turned off)** – Each control has a single identifier across standards. The **Controls** page of the Security Hub console displays all your controls across standards.
- **Consolidated control findings (can be turned on or off)** – When consolidated control findings is turned on, Security Hub produces a single finding for a security check even when a check is shared across multiple standards. This is intended to reduce finding noise. Consolidated control findings is turned *on* for you by default if you enabled Security Hub on or after February 23, 2023. Otherwise, it's turned off by default. However, consolidated control findings is turned on in Security Hub member accounts only if it's turned on in the administrator account. If the feature is turned off in the administrator account, it's turned off in member accounts. For instructions on turning on this feature, see Turning on consolidated control findings.

Both features bring changes to control finding fields and values in the AWS Security Finding Format (ASFF). This section summarizes those changes.

Consolidated controls view – ASFF changes

The consolidated controls view feature introduced the following changes to control finding fields and values in the ASFF.

If your workflows don't rely on the values of these control finding fields, no action is required.

If you have workflows that rely on the specific values of these control finding fields, update your workflows to use the current values.

ASFF field	Sample value before consolidated controls view	Sample value after consolidated controls view, plus description of 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

ASFF field	Sample value before consolidated controls view	Sample value after consolidated controls view, plus description of change
		-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 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 has archived existing findings.
ProductFields.ArchivalReasons:0/ReasonCode	Not applicable (new field)	"CONSOLIDATED_CONTROL_FINDINGS_UPDATE" Provides the reason why Security Hub has archived existing findings.

ASFF field	Sample value before consolidated controls view	Sample value after consolidated controls view, plus description of change
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 PCI DSS documentation."	"For directions on how to correct this issue, consult the AWS Security Hub controls documentation." This field no longer references a standard.
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 turn on consolidated control findings, you may be impacted by the following changes to control finding fields and values in the ASFF. These changes are in addition to the changes previously described for consolidated controls view.

If your workflows don't rely on the values of these control finding fields, no action is required.

If you have workflows that rely on the specific values of these control finding fields, update your workflows to use the current values.

> **Note**
>
> Automated Security Response on AWS v2.0.0 supports consolidated control findings. If you use this version of the solution, you can maintain your workflows when turning on consolidated control findings.

ASFF field	Example value before turning on consolidated control findings	Example value after turning on consolidated control findings, and description of 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 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

ASFF field	Example value before turning on consolidated control findings	Example value after turning on consolidated control findings, and description of change
		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 turning on consolidated control findings	Example value after turning on consolidated control findings, and description of change

Severity	"Severity": {
	"Product": 90,
	"Label": "CRITICAL",
	"Normalized": 90,
	"Original": "CRITICAL"
	}
		"Severity": {
		"Label": "CRITICAL",
		"Normalized": 90,
		"Original": "CRITICAL"
		}
		Security Hub 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"]	["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 turning on consolidated control findings	Example value after turning on consolidated control findings, and description of change
CreatedAt	2022-05-05T08:18:13.138Z	2022-09-25T08:18:13.138Z Format remains the same, but value resets when you turn on 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 turn on 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 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 generates one finding for a security check across standards.

ASFF field	Example value before turning on consolidated control findings	Example value after turning on consolidated control findings, and description of change
ProductFields.StandardsSubscriptionArn	arn:aws:securityhub:us-east-1:123456789012:subscription/aws-foundational-security-best-practices/v/1.0.0	Removed. Security Hub 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 turn on consolidated control findings, Security Hub generates one finding across standards and archives the original findings (separate findings for each standard). To view archived findings, you can visit the **Findings** page of the Security Hub console with the **Record state** filter set to **ARCHIVED**, or use the `GetFindings` API action. Updates you've made to the original findings in the Security Hub console or using the BatchUpdateFindings API won't be preserved in the new findings (if needed, you can recover this data by referring to the archived findings).

Customer-provided ASFF field	Description of change after turning on consolidated control findings
Confidence	Resets to empty state.
Criticality	Resets to empty state.
Note	Resets to empty state.

Customer-provided ASFF field	Description of change after turning on consolidated control findings
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 turning on consolidated control findings

Here's a list of generator ID changes for controls when you turn on consolidated control findings. These apply to controls that Security Hub supported as of February 15, 2023.

GeneratorID before turning on consolidated control findings	GeneratorID after turning on 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
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.12	security-control/IAM.4

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.8	security-control/IAM.14

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.2	security-control/CloudWatch.3

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/3.14	security-control/CloudWatch.14

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
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

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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.18	security-control/RDS.18
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
aws-foundational-security-best-practices/v/1.0.0/RDS.22	security-control/RDS.22

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
aws-foundational-security-best-practices/v/1.0.0/Redshift.2	security-control/Redshift.2

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
aws-foundational-security-best-practices/v/1.0.0/S3.5	security-control/S3.5

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
aws-foundational-security-best-practices/v/1.0.0/SSM.1	security-control/SSM.1

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
pci-dss/v/3.2.1/PCI.CloudTrail.1	security-control/CloudTrail.2

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
pci-dss/v/3.2.1/PCI.IAM.3	security-control/IAM.1

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
pci-dss/v/3.2.1/PCI.SSM.1	security-control/SSM.2

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/Autoscaling.5	security-control/Autoscaling.5

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/DynamoDB.1	security-control/DynamoDB.1

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/EC2.16	security-control/EC2.16

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/ECS.3	security-control/ECS.3

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/ELB.3	security-control/ELB.3

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/EMR.1	security-control/EMR.1

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/IAM.1	security-control/IAM.1

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/KMS.3	security-control/KMS.3

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/Opensearch.5	security-control/Opensearch.5

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/RDS.10	security-control/RDS.10

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/Redshift.2	security-control/Redshift.2

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/S3.9	security-control/S3.9

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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
service-managed-aws-control-tower/v/1.0.0/SSM.4	security-control/SSM.4

GeneratorID before turning on consolidated control findings	GeneratorID after turning on consolidated control findings
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 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 unchanged.

The Security Hub console displays security control IDs and security control titles, regardless of whether consolidated control findings is turned on or off in your account. However, Security Hub findings contain security control IDs and security control titles only if consolidated control findings is turned on in your account. If consolidated control findings is turned off in your account, Security Hub findings contain standard-specific control IDs and titles. For more information about how consolidation impacts control findings, see Sample control findings.

For controls that are part of Service-Managed Standard: AWS Control Tower, the prefix `CT.` is removed from the control ID and title in findings when consolidated control findings is turned on.

To run your own scripts on this table, 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

Standard	Standard control ID and title	Security control ID and title
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 AWS 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 privilege s
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
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

Standard	Standard control ID and title	Security control ID and title
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
CIS v1.2.0	2.5 Ensure AWS Config is enabled	[Config.1] AWS Config should be enabled

Standard	Standard control ID and title	Security control ID and title
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
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

Standard	Standard control ID and title	Security control ID and title
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 AWS 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
CIS v1.2.0	3.9 Ensure a log metric filter and alarm exist for AWS Config configura tion changes	[CloudWatch.9] Ensure a log metric filter and alarm exist for AWS Config configuration changes

Standard	Standard control ID and title	Security control ID and title
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 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 v1.4.0	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 privilege s
CIS v1.4.0	1.17 Ensure a support role has been created to manage incidents with AWS 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
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

Standard	Standard control ID and title	Security control ID and title
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
CIS v1.4.0	3.2 Ensure CloudTrail log file validation is enabled	[CloudTrail.4] CloudTrail log file validation should be enabled

Standard	Standard control ID and title	Security control ID and title
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
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 configuration changes	[CloudWatch.5] Ensure a log metric filter and alarm exist for CloudTrail AWS 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 authentication failures
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

Standard	Standard control ID and title	Security control ID and title
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
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

Standard	Standard control ID and title	Security control ID and title
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 Classic Load Balancer should use load balancer 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] CloudTrail 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
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

Standard	Standard control ID and title	Security control ID and title
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
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

Standard	Standard control ID and title	Security control ID and title
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 AWS 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
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

Standard	Standard control ID and title	Security control ID and title
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 AWS 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
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 installation	[SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation

Standard	Standard control ID and title	Security control ID and title
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 control finding fields, no action is required.

If your workflows rely on the specific format of any control finding fields noted in the tables, you should update your workflows. For example, If you created a Amazon CloudWatch Events 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, the `Compliance.SecurityControlId` field for that control.

If you created custom insights using any of the control finding fields or values that changed, update those insights to use the current fields or values.

ASFF examples

The following sections contain examples of required and optional attributes in the AWS Security Finding Format (ASFF), as well as examples of each resource that ASFF supports.

Topics

Required top-level attributes

The following top-level attributes in the AWS Security Finding Format (ASFF) are required for all findings in Security Hub. For more information about these required 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 captured by a finding was created.

Example

```
"CreatedAt": "2017-03-22T13:22:13.933Z"
```

> **Note**
>
> Security Hub deletes findings 90 days after the most recent update or 90 days after the creation date if no update occurs. To store findings for longer than 90 days, you can configure a rule in Amazon EventBridge that routes findings to your S3 bucket.

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 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 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 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 that uniquely identifies a third-party findings product after the product is registered with Security Hub.

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, 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 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` object provides a set of resource data types that describe the AWS resources that the finding refers to.

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 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 only be updated by 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`, then Security Hub automatically populates the value of the other field.
The `Product` field under `FindingProviderFields` is retired and isn't populated in current findings. Instead, use the `Original` field.

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 only be updated using `BatchUpdateFindings`.

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` API operation. You can only update it by using `BatchImportFindings`.

Example

```
"UpdatedAt": "2017-04-22T13:22:13.933Z"
```

> **Note**
>
> Security Hub deletes findings 90 days after the most recent update or 90 days after the creation date if no update occurs. To store findings for longer than 90 days, you can configure a rule in Amazon EventBridge that routes findings to your S3 bucket.

Optional top-level attributes

These top-level attributes are optional in the AWS Security Finding Format (ASFF). 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 that 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 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 “Using custom product integrations”.

When you use the Security Hub console to filter findings by company name, you use this attribute. When you use the Security Hub API to filter findings by company name, you use the `aws/securityhub/CompanyName` attribute under `ProductFields`. Security Hub does not synchronize those two attributes.

Example

```
"CompanyName": "AWS"
```

Compliance

The `Compliance` object provides finding details related to a control. This attribute is returned for findings generated from a Security Hub control and for findings that AWS Config sends to Security Hub.

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 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 "Using 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 "Using 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
```

FindingProviderFields

`FindingProviderFields` includes the following attributes:

- `Confidence`
- `Criticality`
- `RelatedFindings`
- `Severity`
- `Types`

You can update`FindingProviderFields` by using the`BatchImportFindings` API operation. You cannot update it with `BatchUpdateFindings`.

For details on how Security Hub handles updates from `BatchImportFindings` to `FindingProviderFields` and to the corresponding top-level attributes, see the section called "Using 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 captured by a finding was first observed.

This timestamp reflects the time of when the event or vulnerability was first observed. Consequently, it can differ from the `CreatedAt` timestamp, which reflects the time this finding record was created.

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 that was captured by a finding was most recently observed by the security findings product.

This timestamp reflects the time 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 provide 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 received a finding and begins to process it.

This differs from `CreatedAt` and `UpdatedAt`, which are required time stamps that relate to the finding provider's interaction with the security issue and finding. The `ProcessedAt` time stamp indicates when Security Hub starts to process a finding. A finding appears in a user's account after processing is completed.

```
"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 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 archives an existing finding. For example, Security Hub 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.

Security Hub 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 “Using custom product integrations”.

When you use the Security Hub console to filter findings by product name, you use this attribute.

When you use the Security Hub API to filter findings by product name, you use the `aws/securityhub/ProductName` attribute under `ProductFields`.

Security Hub 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 immediately deleted. You can search, review, and report on them. Security Hub 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 only be updated by `BatchImportFindings`. You cannot update it using `BatchUpdateFindings`.

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 either `NOTIFIED` or `RESOLVED`, then Security Hub automatically sets the workflow status to `NEW`.

Example

```
"RecordState": "ACTIVE"
```

Region

Specifies the AWS Region from which the finding was generated.

Security Hub 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 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" }
]
```

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
 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

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) syntax.

For examples of each resource object, select from the following list.

Topics

- Resource attributes
- AwsAmazonMQ
- AwsApiGateway
- AwsAppSync
- AwsAthena
- AwsAutoScaling
- AwsBackup
- AwsCertificateManager
- AwsCloudFormation
- AwsCloudFront
- AwsCloudTrail
- AwsCloudWatch
- AwsCodeBuild
- AwsDms
- AwsDynamoDB
- AwsEc2

- AwsEcr
- AwsEcs
- AwsEfs
- AwsEks
- AwsElasticBeanstalk
- AwsElasticSearch
- AwsElb
- AwsEventBridge
- AwsGuardDuty
- AwsIam
- AwsKinesis
- AwsKms
- AwsLambda
- AwsMsk
- AwsNetworkFirewall
- AwsOpenSearchService
- AwsRds
- AwsRedshift
- AwsRoute53
- AwsS3
- AwsSageMaker
- AwsSecretsManager
- AwsSns
- AwsSqs
- AwsSsm
- AwsStepFunctions
- AwsWaf
- AwsXray
- Container
- Other

Resource attributes

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:::example-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

You can add resource tags to findings that are ingested into Security Hub, including findings from integrated AWS services and third-party products. You can tag resources that the `GetResources` operation of the AWS Resource Groups Tagging API supports. For a list of supported resources, see Services that support the Resource Groups Tagging API.

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

The following are examples of the AWS Security Finding Format (ASFF) for `AwsAmazonMQ` resources.

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

The following are examples of the AWS Security Finding Format for `AwsApiGateway` resources.

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

The following are examples of the AWS Security Finding Format (ASFF) for `AwsAppSync` resources.

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

The following are examples of the AWS Security Finding Format (ASFF) for `AwsAthena` resources.

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

The following are examples of the AWS Security Finding Format for `AwsAutoScaling` resources.

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

The following are examples of the AWS Security Finding Format for `AwsBackup` resources.

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

The following are examples of the AWS Security Finding Format for `AwsCertificateManager` resources.

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

The following are examples of the AWS Security Finding Format for `AwsCloudFormation` resources.

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

The following are examples of the AWS Security Finding Format for `AwsCloudFront` resources.

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": "my-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

The following are examples of the AWS Security Finding Format for `AwsCloudTrail` resources.

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

The following are examples of the AWS Security Finding Format for `AwsCloudWatch` resources.

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": {
 "ActonsEnabled": 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

The following are examples of the AWS Security Finding Format for `AwsCodeBuild` resources.

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

The following are examples of the AWS Security Finding Format for `AwsDms` resources.

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

The following are examples of the AWS Security Finding Format for `AwsDynamoDB` resources.

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

The following are examples of the AWS Security Finding Format for `AwsEc2` resources.

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"
}
```

AwsEc2VpnConnection

The `AwsEc2VpnConnection` object provides details about an Amazon EC2 VPN connection.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2VpnConnection` object. To view descriptions of `AwsEc2VpnConnection` attributes, see AwsEc2VpnConnectionDetails in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2VpnConnection": {
    "VpnConnectionId": "vpn-205e4f41",
    "State": "available",
    "CustomerGatewayConfiguration": "",
    "CustomerGatewayId": "cgw-5699703f",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-2ccb2245",
    "Category": "VPN"
    "TransitGatewayId": "tgw-09b6f3a659e2b5elf",
    "VgwTelemetry": [
        {
            "OutsideIpAddress": "92.0.2.11",
            "Status": "DOWN",
            "LastStatusChange": "2016-11-11T23:09:32.000Z",
            "StatusMessage": "IPSEC IS DOWN",
            "AcceptedRouteCount": 0
        },
        {
            "OutsideIpAddress": "92.0.2.12",
            "Status": "DOWN",
            "LastStatusChange": "2016-11-11T23:10:51.000Z",
            "StatusMessage": "IPSEC IS DOWN",
            "AcceptedRouteCount": 0
        }
    ],
    "Routes": [{
        "DestinationCidrBlock": "10.24.34.0/24",
        "State": "available"
   }],
    "Options": {
        "StaticRoutesOnly": true
        "TunnelOptions": [{
            "DpdTimeoutSeconds": 30,
            "IkeVersions": ["ikev1", "ikev2"],
            "Phase1DhGroupNumbers": [14, 15, 16, 17, 18},
            "Phase1EncryptionAlgorithms": ["AES128", "AES256"],
            "Phase1IntegrityAlgorithms": ["SHA1", "SHA2-256"],
```

```
            "Phase1LifetimeSeconds": 28800,
            "Phase2DhGroupNumbers": [14, 15, 16, 17, 18],
            "Phase2EncryptionAlgorithms": ["AES128", "AES256"],
            "Phase2IntegrityAlgorithms": ["SHA1", "SHA2-256"],
            "Phase2LifetimeSeconds": 28800,
            "PreSharedKey": "RltXC3REhTw1RAdiM2s1uMfkkSDLyGJoe1QEWeGxqkQ=",
            "RekeyFuzzPercentage": 100,
            "RekeyMarginTimeSeconds": 540,
            "ReplayWindowSize": 1024,
            "TunnelInsideCidr": "10.24.34.0/23"
        }]
   }
}
```

AwsEcr

The following are examples of the AWS Security Finding Format for `AwsEcr` resources.

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

The following are examples of the AWS Security Finding Format for `AwsEcs` resources.

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

The following are examples of the AWS Security Finding Format for `AwsEfs` resources.

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

The following are examples of the AWS Security Finding Format for `AwsEks` resources.

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

The following are examples of the AWS Security Finding Format for `AwsElasticBeanstalk` resources.

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

The following are examples of the AWS Security Finding Format for `AwsElasticSearch` resources.

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

The following are examples of the AWS Security Finding Format for `AwsElb` resources.

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": "doc-example-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

The following are examples of the AWS Security Finding Format for `AwsEventBridge` resources.

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

The following are examples of the AWS Security Finding Format for `AwsGuardDuty` resources.

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

The following are examples of the AWS Security Finding Format for `AwsIam` resources.

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

The following are examples of the AWS Security Finding Format for `AwsKinesis` resources.

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

The following are examples of the AWS Security Finding Format for `AwsKms` resources.

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 for `AwsLambda` resources.

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": "DOC-EXAMPLE-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

The following are examples of the AWS Security Finding Format for `AwsMsk` resources.

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

The following are examples of the AWS Security Finding Format for `AwsNetworkFirewall` resources.

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

The following are examples of the AWS Security Finding Format for `AwsOpenSearchService` resources.

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

The following are examples of the AWS Security Finding Format for `AwsRds` resources.

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

```
"AwsRdsDbClusterSnaphot": {
    "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

The following are examples of the AWS Security Finding Format for `AwsRedshift` resources.

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": "test-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

The following are examples of the AWS Security Finding Format for `AwsRoute53` resources.

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

The following are examples of the AWS Security Finding Format for AwsS3 resources.

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": "DOC-EXAMPLE-BUCKET1",
        "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-858726136312",
 "LogFilePrefix": "buckettestreadwrite23435/"
},
"BucketName": "DOC-EXAMPLE-BUCKET1",
"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

The following are examples of the AWS Security Finding Format for `AwsSageMaker` resources.

AwsSageMakerNotebookInstance

The `AwsSageMakerNotebookInstance` object provides information about a Amazon SageMaker 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

The following are examples of the AWS Security Finding Format for `AwsSecretsManager` resources.

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

The following are examples of the AWS Security Finding Format for `AwsSns` resources.

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

The following are examples of the AWS Security Finding Format for `AwsSqs` resources.

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

The following are examples of the AWS Security Finding Format for `AwsSsm` resources.

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

The following are examples of the AWS Security Finding Format for `AwsStepFunctions` resources.

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

The following are examples of the AWS Security Finding Format for `AwsWaf` resources.

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

The following are examples of the AWS Security Finding Format for `AwsXray` resources.

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"
}
```

Container

Container details that are related to a finding.

The following example shows the AWS Security Finding Format (ASFF) for the `Container` object. To view descriptions of `Container` attributes, see ContainerDetails in the *AWS Security Hub API Reference*.

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

The `Other` object allows you to provide custom fields and values. You use the `Other` object in the following cases.

- The resource type does not have a corresponding `Details` object. To provide details for the resource, you use the `Other` object.
- The `Details` object for the resource type does not include all of the attributes that you want to populate. In this case, use the `Details` object for the resource type to populate the available attributes. Use the `Other` object to populate the attributes that are not in the type-specific object.
- The resource type is not one of the provided types. In this case, you set `Resource.Type` to `Other`, and use the `Other` object to populate 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.

Insights in AWS Security Hub

An AWS Security Hub insight is a collection of related findings. It identifies a 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 offers several built-in managed insights. You cannot modify or delete managed insights.

To track security issues that are unique to your AWS environment and usage, you can create custom insights.

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 the CIS AWS Foundations standard.

Topics

Viewing and filtering the list of insights

The **Insights** page 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 managed insights**.
- To display only custom insights, choose **Custom insights**.

You also can filter the insight list based on text in the insight name.

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.

Viewing and taking action on insight results and findings

For each insight, AWS Security Hub first determines the findings that match the filter criteria, and then uses the grouping attribute to group the matching findings.

From the **Insights** console page, you can view and take action on the results and findings.

If you enable cross-Region aggregation, then in the aggregation Region, the results for managed insights include findings from the aggregation Region and the linked Regions. For custom insight results, if the insight does not filter by Region, then the results include findings from the aggregation Region and linked Regions.

In other Regions, the insight results are only for that Region.

For information on how to configure cross-Region aggregation, see *Cross-Region aggregation*.

Viewing and taking action on insight results (console)

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.

Note that if the findings are grouped by resource identifier or resource type, then 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, then the insight results list both of those resources.

The results list is sorted from most to fewest matching findings.

Security Hub 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 a CloudWatch rule for the `Security Hub Insight Results` event type. See the section called “Automated response and remediation”.

If you have not configured custom actions, then the **Actions** menu is disabled.

To display and take action on the list of insight results

1. Open the AWS Security Hub 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.

Viewing insight results (Security Hub API, AWS CLI)

To view insight results, you can use an API call or the AWS Command Line Interface.

To view insight results (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `GetInsightResults` operation. To identify the insight to return results for, you need the insight ARN. To obtain the insight ARNs for custom insights, use the `GetInsights` operation.
- **AWS CLI** – At the command line, run the `get-insight-results` command.

```
aws securityhub get-insight-results --insight-arn <insight ARN>
```

Example:

```
aws securityhub get-insight-results --insight-arn "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Viewing findings for an insight result (console)

From the insight results list, you can display the list of findings for each result.

To display and take action on insight findings

1. Open the AWS Security Hub 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.

- Change the filters and grouping for the list
- View details for individual findings
- Update the workflow status of findings
- Send findings to custom actions

Managed insights

AWS Security Hub provides several managed insights.

You cannot edit or delete Security Hub 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 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, then the insight results include both resources.

Security Hub 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`

Custom insights

In addition to the AWS Security Hub managed insights, you can create custom insights in Security Hub to track issues that are specific to your environment. Custom insights provide a way to 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 is 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**, then 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.

  When querying your findings, Security Hub applies Boolean AND logic to the set of filters. In other words, a finding only matches if it matches all of the provided filters. For example, if the filters are "Product name is GuardDuty" and "Resource type is `AwsS3Bucket`," then matching findings must match both of these criteria.

  However, Security Hub 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," then a finding matches if it was generated by either GuardDuty or Amazon Inspector.

Note that if you use the resource identifier or resource type as the grouping attribute, then 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.

Creating a custom insight (console)

From the console, you can create a completely new insight.

To create a custom insight

1. Open the AWS Security Hub 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. (Optional) 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**, then choose **Create insight**.

Creating a custom insight (programmatic)

Choose your preferred method, and follow the steps to programmatically create a custom insight in Security Hub. You can specify filters to narrow down the collection of findings in the insight to a specific subset.

The following tabs include instructions in a few languages for creating a custom insight. For support in additional languages, see Tools to Build on AWS.

Security Hub API

1. Run the `CreateInsight` operation.
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.

   If you've enabled cross-region aggregation and call this API from the aggregation Region, the insight applies to matching findings in the aggregation and linked Regions.

AWS CLI

1. At the command line, 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 `group-by-attribute` 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 run this command from the aggregation Region, the insight applies to matching findings from the aggregation and linked Regions.

```
aws securityhub create-insight --name <insight name> --filters <filter values> --
group-by-attribute <attribute name>
```

Example

```
aws securityhub create-insight --name "Critical role findings" --filters
 '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],
 "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' --group-by-
attribute "ResourceId"
```

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
```

Modifying a custom insight (console)

You can modify 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.

To modify an insight

1. Open the AWS Security Hub 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**.

Modifying a custom insight (programmatic)

To modify a custom insight, choose your preferred method, and follow the instructions.

Security Hub API

1. Run the `UpdateInsight` operation.
2. To identify the custom insight, provide the insight's Amazon Resource Name (ARN). To get the ARN of a custom insight, run the `GetInsights` operation.
3. Update the `Name`, `Filters`, and `GroupByAttribute` parameters as needed.

AWS CLI

1. At the command line, run the `update-insight` command.
2. To identify the custom insight, provide the insight's Amazon Resource Name (ARN). To get the ARN of a custom insight, run the `get-insights` command.
3. Update the `name`, `filters`, and `group-by-attribute` parameters as needed.

```
aws securityhub update-insight --insight-arn <insight ARN> [--name <new name>] [--
filters <new filters>] [--group-by-attribute <new grouping attribute>]
```

Example

```
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

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"
```

Creating a new custom insight from a managed insight (console)

You cannot save changes to or delete a managed insight. You can use a managed insight as the basis for a new custom insight.

To create a new custom insight from a managed insight

1. Open the AWS Security Hub 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**.

Deleting a custom insight (console)

When you no longer want a custom insight, you can delete it. You cannot delete managed insights.

To delete a custom insight

1. Open the AWS Security Hub 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**.

Deleting a custom insight (programmatic)

To delete a custom insight, choose your preferred method, and follow the instructions.

Security Hub API

1. Run the `DeleteInsight` operation.
2. To identify the custom insight to delete, provide the insight's ARN. To get the ARN of a custom insight, run the `GetInsights` operation.

AWS CLI

1. At the command line, run the `delete-insight` command.
2. To identify the custom insight, provide the insight's ARN. To get the ARN of a custom insight, run the `get-insights` command.

```
aws securityhub delete-insight --insight-arn <insight ARN>
```

Example

```
aws securityhub delete-insight --insight-arn "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
```

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"
```

Automations

Security Hub automations can help you quickly modify and remediate findings based on your specifications.

Security Hub 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 EventBridge rules that define automatic actions to take against specific findings and insights.

Automation rules apply before EventBridge rules. That is, automation rules are triggered and update a finding before it's sent to EventBridge. 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 occasionally updates control titles and descriptions, control IDs stay the same.

Topics

Automation rules

Automation rules can be used to automatically update findings in Security Hub. As findings are ingested, Security Hub can apply a variety of rule actions, such as suppressing findings, changing their severity, and adding notes to findings. Such rule actions take effect when findings match your specified criteria, such as which resource or account ID the finding is associated with or its title.

Examples of use cases for automation rules include:

- 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.

Automation rules can be used to update select finding fields in the AWS Security Finding Format (ASFF). 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. If you use a rule template, you can modify it as needed for your use case.

How automation rules work

The Security Hub administrator can create an automation rule by defining rule *criteria*. When a finding matches the defined criteria, Security Hub applies the rule *action* to it. For more information about available criteria and actions, see Available rule criteria and rule actions.

Only the Security Hub administrator account can create, delete, edit, and view automation rules. A rule that an administrator creates applies to findings in the administrator account and all member accounts. By providing member account IDs as rule criteria, Security Hub administrators can also use automation rules to update findings or take action on findings in specific member accounts.

> **Important**
>
> An automation rule applies only in the AWS Region in which it's created. To apply a rule in multiple Regions, the delegated administrator must create the rule in each Region. This can be done through the Security Hub console, Security Hub API, or AWS CloudFormation. You can also use a multi-Region deployment script.

To get a history of how automation rules have changed your findings, see Reviewing finding history.

Automation rules apply to new and updated findings that Security Hub generates or ingests after you create the rule. Security Hub updates control findings every 12-24 hours or when the associated resource changes state. For more information, see Schedule for running security checks.

Security Hub currently supports a maximum of 100 automation rules for an administrator account.

Rule order

When creating automation rules, you assign each rule an 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 most recently created rule has the lowest numerical value for rule order and therefore applies first. Security Hub applies subsequent rules in ascending order.

When you create a rule through the Security Hub 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`
  - `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 findings that match Rule A criteria. Next, 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.

Available rule criteria and rule actions

The following ASFF fields are currently supported as criteria for automation rules.

ASFF field	Filters	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,`	String

ASFF field	Filters	Field type
	NOT_EQUALS, PREFIX_NOT_EQUALS	
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)
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

ASFF field	Filters	Field type
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)
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

ASFF field	Filters	Field type
RelatedFindingsProductArn	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
ResourceApplicationArn	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
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

ASFF field	Filters	Field type
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]
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

ASFF field	Filters	Field type
VerificationState	CONTAINS, EQUALS, PREFIX, NOT_CONTAINS, NOT_EQUALS, PREFIX_NOT_EQUALS	String
WorkflowStatus	Is, Is Not	Select: [NEW, NOTIFIED, RESOLVED, SUPPRESSED]

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 and ASFF examples.

Tip

If you want Security Hub 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 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 Enabling and disabling controls in all standards.

Creating automation rules

You can create a custom rule from scratch or use a pre-populated Security Hub rule template.

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 console, Security Hub shows you a preview of the findings to which your rule applies. The preview 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 rule from a template (console only)

Currently, only the Security Hub console supports rule templates. These templates reflect common use cases for automation rules and can help you get started with the feature. Complete the following steps to create an automation rule from a template in the console.

Console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in to the Security Hub administrator account.
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 preview 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**.

Creating a custom rule

Choose your preferred method, and complete the following steps to create a custom automation rule.

Console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in to the Security Hub administrator account.
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 preview 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

1. Run `CreateAutomationRule` from the Security Hub 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 applies rules with a lower numerical value for this parameter first.
5. For the `RuleStatus` parameter, specify if you want Security Hub 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 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 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.

Example API request:

```
{
    "Actions": [{
        "Type": "FINDING_FIELDS_UPDATE",
        "FindingFieldsUpdate": {
            "Workflow": {
                "Status": "SUPPRESSED"
            },
            "Note": {
                "Text": "Known issue that is not a risk.",
                "UpdatedBy": "sechub-automation"
            }
        }
    }],
```

```
    "Criteria": {
        "ProductName": [{
            "Value": "Security Hub",
            "Comparison": "EQUALS"
        }],
        "ComplianceStatus": [{
            "Value": "FAILED",
            "Comparison": "EQUALS"
        }],
        "RecordState": [{
            "Value": "ACTIVE",
            "Comparison": "EQUALS"
        }],
        "WorkflowStatus": [{
            "Value": "NEW",
            "Comparison": "EQUALS"
        }],
        "GeneratorId": [{
            "Value": "aws-foundational-security-best-practices/v/1.0.0/IAM.1",
            "Comparison": "EQUALS"
        }]
    },
    "Description": "Sample rule description",
    "IsTerminal": false,
    "RuleName": "sample-rule-name",
    "RuleOrder": 1,
    "RuleStatus": "ENABLED",
}
```

AWS CLI

1. Run the `create-automation-rule` command from the Security Hub administrator account. This command creates a rule with a specific Amazon Resource Name (ARN).
2. Provide a name and description for the rule.
3. Include the `is-terminal` parameter if you want this rule to be the last rule applied to findings that match the rule criteria. Otherwise, include the `no-is-terminal` parameter.
4. For the `rule-order` parameter, provide the order of the rule. Security Hub applies rules with a lower numerical value for this parameter first.
5. For the `rule-status` parameter, specify if you want Security Hub 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 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 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.

Example command:

```
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
```

Viewing automation rules

Choose your preferred method, and follow the steps to view your automation rules and the details of each rule.

Console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in to the Security Hub administrator account.
2. In the navigation pane, choose **Automations**.
3. Choose a rule name. Alternatively, select a rule.
4. Choose **Actions** and **View**.

API

1. To view the automation rules for your account, run `ListAutomationRules` from the Security Hub 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`.

   Example API request:

   ```
   {
    "MaxResults": 50,
    "NextToken": "cVpdnSampleTokenYcXgTockBW44c"
   }
   ```
2. For additional rule details, including the criteria and actions for a rule, run `BatchGetAutomationRules` from the Security Hub administrator account.

   Example API request:

   ```
   {
       "AutomationRulesArns": [
         "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",
         "arn:aws:securityhub:us-east-1:123456789012:automation-
   rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
         "arn:aws:securityhub:us-east-1:123456789012:automation-
   rule/a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa"
       ]
   ```

```
}
```

AWS CLI

1. To view the automation rules for your account, run the `list-automation-rules` command from the Security Hub administrator account. This command returns the rule ARNs and other metadata for your rules. No input parameters are required for this command, but you can optionally provide `max-results` to limit the number of results and `next-token` as a pagination parameter.

   Example command:

   ```
   aws securityhub list-automation-rules  \
   --max-results 5 \
   --next-token cVpdnSampleTokenYcXgTockBW44c \
   --region us-east-1
   ```

2. For additional rule details, including the criteria and actions for a rule, run the `batch-get-automation-rules` command from the Security Hub administrator account.

   Example command:

   ```
   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

When you edit an automation rule, the changes apply to new and updated findings that Security Hub 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 rule order.

Console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in to the Security Hub administrator account.
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

1. Run `BatchUpdateAutomationRules` from the Security Hub 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`.

Example API request:

```
{
    "UpdateAutomationRulesRequestItems": [
        {
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-
rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "RuleOrder": 15,
            "RuleStatus": "Enabled"
        },
        {
            "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-
rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "RuleStatus": "Disabled"
        }
    ]
}
```

AWS CLI

1. Run the `batch-update-automation-rules` command from the Security Hub 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`.

Example command:

```
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 rule order

In some cases, you might want to keep the rule criteria and actions as is, but change the order in which Security Hub applies an automation rule. Choose your preferred method, and follow the steps to edit rule order.

Console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in to the Security Hub administrator account.
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 console, Security Hub 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

1. Run `BatchUpdateAutomationRules` from the Security Hub 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 applies a rule with an earlier value for the `UpdatedAt` field first (that is, the rule which was most recently edited applies last).

AWS CLI

1. Run the `batch-update-automation-rules` command from the Security Hub 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 applies a rule with an earlier value for the `UpdatedAt` field first (that is, the rule which was most recently edited applies last).

Deleting automation rules

When you delete an automation rule, Security Hub removes it from your account and no longer applies the rule to findings.

Choose your preferred method, and follow the steps to delete an automation rule. You can delete one or more rules in a single request.

> **Tip**
>
> As an alternative to deletion, you can disable a rule. This retains the rule for future use, but Security Hub won't apply the rule to any matching findings until you enable it.

Console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in to the Security Hub administrator account.
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

1. Run `BatchDeleteAutomationRules` from the Security Hub 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).

Example API request:

```
{
    "AutomationRulesArns": [
        "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",
        "arn:aws:securityhub:us-east-1:123456789012:automation-
rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "arn:aws:securityhub:us-east-1:123456789012:automation-
rule/a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa"
    ]
}
```

AWS CLI

1. Run the `batch-delete-automation-rules` command from the Security Hub administrator account.
2. For the `automation-rules-arns` 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).

Example command:

```
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
```

Automation rule examples

This section include some example automation rules for common use cases. These examples correspond to rule templates in the Security Hub 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",
            "Comparison": "EQUALS"
        }],
        "ComplianceStatus": [{
            "Value": "FAILED",
            "Comparison": "EQUALS"
        }],
        "RecordState": [{
            "Value": "ACTIVE",
            "Comparison": "EQUALS"
        }],
        "WorkflowStatus": [{
            "Value": "NEW",
            "Comparison": "EQUALS"
        }],
        "ResourceId": [{
            "Value": "arn:aws:s3:::examplebucket/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",
"Comparison": "EQUALS"
}],
"ComplianceStatus": [{
"Value": "FAILED",
"Comparison": "EQUALS"
}],
"RecordState": [{
"Value": "ACTIVE",
"Comparison": "EQUALS"
}],
"WorkflowStatus": [{
"Value": "NEW",
"Comparison": "EQUALS"
}],
"ResourceId": [{
"Value": "arn:aws:s3:::examplebucket/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",
            "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",
```

```
"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 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
```

Automated response and remediation

With Amazon EventBridge, you can automate your AWS services to respond automatically to system events such as application availability issues or resource changes. Events from AWS services are delivered to EventBridge in near-real time and on a guaranteed basis. 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 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 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 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

Types of Security Hub integration with EventBridge

Security Hub uses the following EventBridge event types to support the following types of integration with EventBridge.

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

All findings (Security Hub Findings - Imported)

Security Hub 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 *Cross-Region aggregation*.

You can define rules in EventBridge that automatically route findings to an Amazon S3 bucket, a remediation workflow, or a third-party tool. 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 a rule for automatically sent findings”.

Findings for custom actions (Security Hub Findings - Custom Action)

Security Hub 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 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 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 and use the custom action in Security Hub to manually send findings to your ticketing system.

For examples of how to send Security Hub findings to EventBridge for further processing, see How to Integrate AWS Security Hub Custom Actions with PagerDuty and How to Enable Custom Actions in AWS Security Hub 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, 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

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 sends an event to EventBridge.

Security Hub Findings - Imported

Security Hub Findings - Imported events that are sent from Security Hub 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 a rule for automatically sent findings".

Security Hub Findings - Custom Action

Security Hub Findings - Custom Action events that are sent from Security Hub 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 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 automatically sent findings

You can create a rule in 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 `BatchImportFindings` and `BatchUpdateFindings`.

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 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 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*.

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 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

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 Security Hub custom actions to send findings or insight results to EventBridge, you first create the custom action in Security Hub. Then, define rules in EventBridge that apply to your custom actions.

You can create up to 50 custom actions.

If you enabled cross-Region aggregation, and manage findings from the aggregation Region, then create custom actions in the aggregation Region.

The rule in EventBridge uses the ARN from the custom action.

Creating a custom action (console)

When you create a custom action, you specify the name, description, and a unique identifier.

To create a custom action in Security Hub (console)

1. Open the AWS Security Hub 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.

Creating a custom action (Security Hub API, AWS CLI)

To create a custom action, you can use an API call or the AWS Command Line Interface.

To create a custom action (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `CreateActionTarget` operation. When you create a custom action, you provide the name, description, and custom action identifier.
- **AWS CLI** – At the command line, run the `create-action-target` command.

```
create-action-target --name <customActionName> --
description <customActionDescription> --id <customActionidentifier>
```

Example

```
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 process the custom action, you must create a corresponding rule in EventBridge. The rule definition includes the 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

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 your Security Hub custom actions and EventBridge rules, you can send findings and insight results to EventBridge for further 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

1. Open the AWS Security Hub 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 selected control. See the section called “Viewing details for a control”.
   - From **Integrations**, you can navigate to a list of findings generated by an enabled integration. See the section called “Viewing the findings from an integration”.
   - From **Insights**, you can navigate to a list of findings for an insight result. See the section called “Viewing insight results and findings”.
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 sends a separate **Security Hub Findings - Custom Action** event for each finding.

To send insight results to EventBridge

1. Open the AWS Security Hub 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.

Product integrations in AWS Security Hub

AWS Security Hub can aggregate security finding data from several AWS services and from supported AWS Partner Network (APN) security solutions. This aggregation provides a comprehensive view of security and compliance across your AWS environment.

You can also send findings that are generated from your own custom security products.

Important

From the supported AWS and partner product integrations, Security Hub receives and consolidates only findings that are generated after you enable Security Hub in your AWS accounts.
The service does not retroactively receive and consolidate security findings that were generated before you enabled Security Hub.

For details on how Security Hub charges for ingested findings, see Security Hub pricing.

Topics

Managing product integrations

The **Integrations** page in the AWS Management Console provides access to all of the available AWS and third-party product integrations. The AWS Security Hub API also provides operations to allow you to manage integrations.

Note

Some integrations are not available in all Regions. If an integration is not supported in the current Region, it is not listed on the **Integrations** page.

See also the section called "Integrations that are supported in China (Beijing) and China (Ningxia)" and the section called "Integrations that are supported in AWS GovCloud (US-East) and AWS GovCloud (US-West)".

Viewing and filtering the list of integrations (console)

From the **Integrations** page, you can view and filter the list of integrations.

To view the list of integrations

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

On the **Integrations** page, the integrations with other AWS services are listed first, followed by the 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

Viewing information about product integrations (Security Hub API, AWS CLI)

To view information about product integrations, you can use an API call or the AWS Command Line Interface. You can display information about all product integrations, or information about the product integrations that you have enabled.

To view information about all available product integrations (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DescribeProducts` operation. To identify a specific product integration to return, use the `ProductArn` parameter to provide the integration ARN.
- **AWS CLI** – At the command line, run the `describe-products` command. To identify a specific product integration to return, provide the integration ARN.

```
aws securityhub describe-products --product-arn "<integrationARN>"
```

Example

```
aws securityhub describe-products --product-arn "arn:aws:securityhub:us-
east-1::product/3coresec/3coresec"
```

To view information about product integrations you have enabled (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `ListEnabledProductsForImport` operation.
- **AWS CLI** – At the command line, run the `list-enabled-products-for-import` command.

```
aws securityhub list-enabled-products-for-import
```

Enabling an integration

On the **Integrations** page, each integration provides the required steps to enable the integration.

For most of the integrations with other AWS services, the only required step is to enable the other service. The integration information includes a link to the service home page. When you enable the other service, a resource-level permission that allows Security Hub 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 perform those tasks.

If more than one version of a product is available in AWS Marketplace, select the version 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 needs to receive findings from that product.

Disabling and enabling the flow of findings from an integration (console)

On the **Integrations** page, for integrations that send findings, the **Status** information indicates whether you are currently accepting findings.

To stop accepting findings, choose **Stop accepting findings**.

To resume accepting findings, choose **Accept findings**.

Disabling the flow of findings from an integration (Security Hub API, AWS CLI)

To disable the flow of findings from an integration, you can use an API call or the AWS Command Line Interface.

To disable the flow of findings from an integration (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DisableImportFindingsForProduct` operation. To identify the integration to disable, you need the ARN of your subscription. To obtain the subscription ARNs for your enabled integrations, use the `ListEnabledProductsForImport` operation.
- **AWS CLI** – At the command line, run the `disable-import-findings-for-product` command.

```
aws securityhub disable-import-findings-for-product --product-subscription-
arn <subscription ARN>
```

Example

```
aws securityhub disable-import-findings-for-product --product-subscription-arn
 "arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/
crowdstrike-falcon"
```

Enabling the flow of findings from an integration (Security Hub API, AWS CLI)

To enable the flow of findings from an integration, you can use an API call or the AWS Command Line Interface.

To enable the flow of findings from an integration (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `EnableImportFindingsForProduct` operation. 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.
- AWS CLI**:** At the command line, run the `enable-import-findings-for-product` command.

```
aws securityhub enable-import-findings-for-product --product-arn <integration ARN>
```

Example

```
aws securityhub enable-import-findings-for product --product-arn
 "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/crowdstrike-falcon"
```

Viewing the findings from an integration

For integrations that you accept findings for (**Status** is **Accepting findings**), to view a list of findings, 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 *Cross-Region aggregation*.

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 AWS Security Hub

AWS Security Hub supports integrations with several other AWS services.

> **Note**
>
> Some integrations are only available in select AWS Regions.
> If an integration is not supported in a specific Region, it is not listed on the **Integrations** page of the Security Hub console.
> For more information, see Integrations that are supported in China (Beijing) and China (Ningxia) and Integrations that are supported in AWS GovCloud (US-East) and AWS GovCloud (US-West).

Unless indicated below, AWS service integrations that send findings to Security Hub are automatically activated after you enable Security Hub. Integrations that receive Security Hub findings may require additional steps for activation. Review the information about each integration to learn more.

Overview of AWS service integrations with Security Hub

Here is an overview of AWS services that send findings to Security Hub or receive findings from Security Hub.

Integrated AWS service	Direction
AWS Config	Sends findings
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
AWS Systems Manager Patch Manager	Sends findings
AWS Audit Manager	Receives findings
AWS Chatbot	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

The following AWS services integrate with Security Hub by sending findings to Security Hub. Security Hub transforms the findings into 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. These findings can be viewed alongside other Security Hub findings, providing a comprehensive overview of your security posture.

AWS Config uses Amazon EventBridge to send AWS Config rule evaluations to Security Hub. Security Hub transforms the rule evaluations into findings that follow the AWS Security Finding Format. Security Hub then enriches the findings on a best effort basis by getting more information about the impacted resources, such as the Amazon Resource Name (ARN) and creation date. Resource tags in AWS Config rule evaluations aren't included in Security Hub findings.

For more information about this integration, see the following sections.

How AWS Config sends findings to Security Hub

All findings in Security Hub 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 via EventBridge. Security Hub 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

Once the integration is activated, AWS Config sends evaluations of all AWS Config managed rules and custom rules to Security Hub. Only evaluations from service-linked AWS Config rules, such as those used to run checks on security controls, are excluded.

Sending AWS Config findings to Security Hub

When the integration is activated, Security Hub will automatically assign the permissions necessary to receive findings from AWS Config. Security Hub 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 within five minutes.

Retrying when Security Hub is not available

AWS Config sends findings to Security Hub on a best-effort basis through EventBridge. When an event isn't successfully delivered to Security Hub, EventBridge retries delivery for up to 24 hours or 185 times, whichever comes first.

Updating existing AWS Config findings in Security Hub

After AWS Config sends a finding to Security Hub, it can send updates to the same finding to Security Hub 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. Security Hub deletes findings 90 days after the most recent update or 90 days after creation if no update occurs.

Regions in which AWS Config findings exist

AWS Config findings occur on a Regional basis. AWS Config sends findings to Security Hub in the same Region or Regions where the findings occur.

Viewing AWS Config findings in Security Hub

To view your AWS Config findings, choose **Findings** from the Security Hub 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

Security Hub 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.

Config rule evaluation finding type	ASFF finding type	Hardcoded value
detail.awsAccountId	AwsAccountId	

Config rule evaluation finding type	ASFF finding type	Hardcoded value
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}` "
Configuration Item "ARN" or Security Hub computed ARN	Resources[i].id	
detail.resourceType	Resources[i].Type	"AwsS3Bucket"
	Resources[i].Partition	"aws"
	Resources[i].Region	"eu-central-1"

Config rule evaluation finding type	ASFF finding type	Hardcoded value
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 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 1024 characters, it will be truncated to 1024 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:::config-integration-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, this integration is activated automatically. AWS Config immediately begins to send findings to Security Hub.

Stopping the publication of findings to Security Hub

To stop sending findings to Security Hub, you can use the Security Hub console, the Security Hub API, or the AWS CLI.

See Disabling and enabling the flow of findings from an integration (console) or Disabling the flow of findings from an integration (Security Hub API, AWS CLI).

AWS Firewall Manager (Sends findings)

Firewall Manager sends findings to Security Hub 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, this integration is automatically activated. Firewall Manager immediately begins to send findings to Security Hub.

To learn more about the integration, view the **Integrations** page in the Security Hub console.

To learn more about Firewall Manager, see the *AWS WAF Developer Guide*.

Amazon GuardDuty (Sends findings)

GuardDuty sends all of the findings it generates to Security Hub.

New findings from GuardDuty are sent to Security Hub 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 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.

After you enable Security Hub, this integration is automatically activated. GuardDuty immediately begins to send findings to Security Hub.

For more information about the GuardDuty integration, see Integration with AWS Security Hub 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 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.

For more information about the integration, see the following sections.

How AWS Health sends findings to Security Hub

In Security Hub, 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 also has a set of rules that it uses to detect security issues and generate findings.

Security Hub 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 Managing and reviewing finding details and history. You can also track the status of an investigation into a finding. See Taking action on findings in AWS Security Hub.

All findings in Security Hub 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.

Types of findings that AWS Health sends to Security Hub

Once the integration is enabled, AWS Health sends all security-related findings it generates to Security Hub. The findings are sent to Security Hub using the AWS Security Finding Format (ASFF). Security-related findings are defined as the following:

- Any finding associated with an AWS security service
- Any finding with the words `security`,`abuse`, or `certificate` in the AWS Health **typeCode**
- Any finding where the AWS Health service is `risk` or `abuse`

Sending AWS Health findings to Security Hub

When you choose to accept findings from AWS Health, Security Hub will automatically assign the permissions necessary to receive the findings from AWS Health. Security Hub 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 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 within five minutes.

Retrying when Security Hub is not available

AWS Health sends findings to Security Hub on a best-effort basis through EventBridge. When an event isn't successfully delivered to Security Hub, EventBridge retries sending the event for 24 hours.

Updating existing findings in Security Hub

After AWS Health sends a finding to Security Hub, it can send updates to the same finding to reflect additional observations of the finding activity to Security Hub.

Regions in which findings exist

For global events, AWS Health sends findings to Security Hub 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 in the same Region or Regions where the events occur.

Viewing AWS Health findings in Security Hub

To view your AWS Health findings in Security Hub, 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

AWS Health sends the findings to Security Hub using the AWS Security Finding Format (ASFF). AWS Health finding uses a different event pattern compared to Security Hub ASFF format. The table below details all the AWS Health finding fields with their ASFF counterpart as they appear in Security Hub.

Health finding type	ASFF finding type	Hardcoded value
account	AwsAccountId	
detail.startTime	CreatedAt	
detail.eventDescription.latestDescription	Description	
detail.eventTypeCode	GeneratorId	

Health finding type	ASFF finding type	Hardcoded value
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
“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 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, this integration is automatically activated. AWS Health immediately begins to send findings to Security Hub.

Stopping the publication of findings to Security Hub

To stop sending findings to Security Hub, you can use the Security Hub console, Security Hub API, or AWS CLI.

See Disabling and enabling the flow of findings from an integration (console) or Disabling the flow of findings from an integration (Security Hub API, AWS CLI).

AWS Identity and Access Management Access Analyzer (Sends findings)

With IAM Access Analyzer, all findings are sent to Security Hub.

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 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 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, this integration is automatically activated. Amazon Inspector immediately begins to send all of the findings that it generates to Security Hub.

For more information about the integration, see Integration with AWS Security Hub in the *Amazon Inspector User Guide*.

Security Hub can also receive findings from Amazon Inspector Classic. Amazon Inspector Classic sends findings to Security Hub that are generated through assessment runs for all supported rules packages.

For more information about the integration, see Integration with AWS Security Hub 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",
```

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, visit the Integrations page of the Security Hub 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.

AWS IoT Device Defender Audit sends check summaries to Security Hub, 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. Audit also sends finding updates to Security Hub.

For more information about this integration, see Integration with AWS Security Hub in the *AWS IoT Developer Guide*.

Amazon Macie (Sends findings)

A finding from Macie can indicate that there is a potential policy violation or that sensitive data, such as personally identifiable information (PII), is present in data that your organization stores in Amazon S3.

After you enable Security Hub, Macie automatically starts sending policy findings to Security Hub. You can configure the integration to also send sensitive data findings to Security Hub.

In Security Hub, 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.

Macie also sends generated sample findings to Security Hub. 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 Amazon Macie integration with AWS Security Hub in the *Amazon Macie User Guide*.

AWS Systems Manager Patch Manager (Sends findings)

AWS Systems Manager Patch Manager sends findings to Security Hub 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, this integration is automatically activated. Systems Manager Patch Manager immediately begins to send findings to Security Hub.

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

The following AWS services are integrated with Security Hub and receive findings from Security Hub. 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.

AWS Audit Manager (Receives findings)

AWS Audit Manager receives findings from Security Hub. 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 checks supported by AWS Audit Manager lists the controls for which Security Hub sends findings to Audit Manager.

AWS Chatbot (Receives findings)

AWS Chatbot is an interactive agent that helps you to monitor and interact with your AWS resources in your Slack channels and Amazon Chime chat rooms.

AWS Chatbot receives findings from Security Hub.

To learn more about the AWS Chatbot integration with Security Hub, see the Security Hub integration overview in the *AWS Chatbot 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 integration with Detective allows you to pivot from Amazon GuardDuty findings in Security Hub 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 or Detective.

For findings received from other AWS services, the finding details panel on the Security Hub 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 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 as a source in the Security Lake console, Security Lake API, or AWS CLI. Once you complete these steps, Security Hub begins to send all findings to Security Lake.

Security Lake automatically normalizes Security Hub 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 findings.

For more information about this integration, including instructions on adding Security Hub as a source and creating subscribers, see Integration with AWS Security Hub 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, and update those findings in Security Hub.

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, the integration is updated automatically.

Security Hub sends the results of its AWS Foundational Security Best Practices checks to Trusted Advisor.

For more information about the Security Hub integration with Trusted Advisor, see Viewing AWS Security Hub controls in AWS Trusted Advisor in the *AWS Support User Guide*.

Available third-party partner product integrations

AWS Security Hub integrates with multiple third-party partner products. An integration may perform one or more of the following actions:

- Send findings that it generates to Security Hub.
- Receive findings from Security Hub.
- Update findings in Security Hub.

All integrations that send findings to Security Hub have an Amazon Resource Name (ARN).

> **Note**
>
> Some integrations are only available in select AWS Regions.
> The **Integrations** page of the Security Hub console lists all supported integrations for the current Region.
> For more information, see Integrations that are supported in China (Beijing) and China (Ningxia) and Integrations that are supported in AWS GovCloud (US-East) and AWS GovCloud (US-West).

If you have a security solution and are interested in becoming a Security Hub partner, email `<securityhub-partners@amazon.com>`. For more information, see the *AWS Security Hub Partner Integration Guide*.

Overview of third-party integrations with Security Hub

Here's an overview of the third party integrations that send findings to Security Hub or receive findings from Security Hub.

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/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/barra

Integration	Direction	ARN (if applicable)
		cuda/cloudsecurity guardian
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
Capitis Solutions – C2VS	Sends findings	arn:aws:securityhub: *<REGION>*::product/capitis/c2vs
Check Point – CloudGuard IaaS	Sends findings	arn:aws:securityhub: *<REGION>*:758245563457:product/checkpoint/cloudguard-iaas
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

Integration	Direction	ARN (if applicable)
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
Drata	Sends findings	arn:aws:securityhub: *<REGION>*::product/drata/drata-integration

Integration	Direction	ARN (if applicable)
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
Guardicore – Centra 4.0	Sends findings	arn:aws:securityhub: *<REGION>*::product/guardicore/guardicore

Integration	Direction	ARN (if applicable)
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
NETSCOUT – NETSCOUT Cyber Investigator	Sends findings	arn:aws:securityhub:us-east-1::product/netscout/netscout-cyber-investigator

Integration	Direction	ARN (if applicable)
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
Rapid7 – InsightVM	Sends findings	arn:aws:securityhub: *<REGION>*:336818582268:product/rapid7/insightvm
SecureCloudDB – SecureCloudDB	Sends findings	arn:aws:securityhub: *<REGION>*::product/securecloudddb/securecloudddb

Integration	Direction	ARN (if applicable)
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
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

Integration	Direction	ARN (if applicable)
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
Atlassian - Jira Service Management Cloud	Receives and updates findings	Not applicable
Atlassian – Opsgenie	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

Integration	Direction	ARN (if 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
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`

Integration	Direction	ARN (if applicable)
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

The following third party partner product integrations send findings to Security Hub. Security Hub 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 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

Capitis Solutions – C2VS

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/capitis/c2vs`

C2VS is a customizable compliance solution designed to automatically identify your application-specific misconfigurations and their root cause.

Product 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 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.

Product 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 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

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 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 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

Partner documentation

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 in real time, enhancing visibility into your AWS resources and Trend Micro Cloud One event details in Security Hub.

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 to visualize and respond to issues that Wiz detects from the Security Hub console.

AWS Marketplace link

Partner documentation

Third-party integrations that receive findings from Security Hub

The following third party partner product integrations receive findings from Security Hub. Where noted, the products may also update findings. In this case, finding updates that you make in the partner product will also be reflected in Security Hub.

Atlassian - Jira Service Management

Integration type: Receive and update

The AWS Service Management Connector for Jira sends findings from Security Hub to Jira. Jira issues are created based on the findings. When the Jira issues are updated, the corresponding findings are updated in Security Hub.

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 – 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 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.

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 ensures that mission critical security-related incidents are routed to the appropriate teams for immediate resolution.

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 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, 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 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 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 allows security findings from Security Hub 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.

Any updates to these incidents and problems result in updates to the findings in Security Hub.

For an overview of the integration and how it works, watch the video AWS Security Hub - 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 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, 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

The following third party partner product integrations send findings to and receive findings from Security Hub.

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, 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

Using custom product integrations to send findings to AWS Security Hub

In addition to findings generated by the integrated AWS services and third-party products, Security Hub can consume findings that are generated by other custom security products.

You can send these findings to Security Hub manually by using the `BatchImportFindings` API operation.

When setting up the custom integration, use the guidelines and checklists provided in the *Security Hub Partner Integration Guide*.

Requirements and recommendations for sending findings from custom security products

Before you can successfully invoke the `BatchImportFindings` API operation, you must enable Security Hub.

You must provide the finding details using the the section called “Finding format”. For the findings from your custom integration, use the following requirements and recommendations.

Setting the product ARN

When you enable Security Hub, a default product Amazon Resource Name (ARN) for Security Hub 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.

Defining the company and product name

You can use `BatchImportFindings` to set a preferred company name and product name for the custom integration that is sending findings to Security Hub.

Your specified names replace the preconfigured company name and product name, called personal name and default name respectively, and appear in the Security Hub console and the JSON of each finding. See Using BatchImportFindings to create and update findings.

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 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 solution.

Sending findings from Chef InSpec scans to Security Hub

You can create an AWS CloudFormation template that runs a Chef InSpec compliance scan and then sends findings to Security Hub.

For more details, see Continuous compliance monitoring with Chef InSpec and AWS Security Hub.

Sending container vulnerabilities detected by Trivy to Security Hub

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.

For more details, see How to build a CI/CD pipeline for container vulnerability scanning with Trivy andAWS Security Hub.

Security controls and standards in AWS Security Hub

AWS Security Hub consumes, aggregates, and analyzes security findings from various supported AWS and third-party products.

Security Hub also generates its own findings by running automated and continuous security checks against rules. The rules are represented by *security controls*. The controls may, in turn, be enabled in one or more *security standards*. The controls help you determine whether the requirements in a standard are being met.

Security checks against controls generate findings that you can use to monitor your security posture and identify specific AWS accounts or resources that require attention. Each control is related to an AWS service and resource. For example, security checks against the CloudTrail.4 control determine whether you have configured log file validation on your AWS CloudTrail logs. For more information about controls, see Viewing and managing security controls.

You can enable a control in one or more enabled Security Hub standards. When you enable a standard, Security Hub automatically enables the controls that apply to the standard. Security standards allow you to focus on a specific compliance framework. Security Hub defines the controls that apply to each standard. For more information about security standards, see Viewing and managing security standards.

Based on the results of security checks, Security Hub calculates an overall security score and standard-specific security scores. These scores help you understand your security posture. For more information about scores, see How security scores are calculated.

For information about Security Hub pricing for security checks, see Security Hub pricing.

Topics

IAM permissions to configure standards and controls

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 needs permissions to call the following API actions. Without adding permissions for these actions, you won't be able to call these APIs. To get the necessary permissions, you can use Security Hub managed policies. Alternatively, you can update custom IAM policies to include permissions for these actions. Custom policies should also include permissions for the `DescribeStandardsControls` and `UpdateStandardsControl` APIs.

- **`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` API if an administrator doesn't want to allow member accounts to enable or disable 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 console. Because only the console calls **`BatchGetControlEvaluations`**, this IAM permission doesn't directly correspond to publicly documented Security Hub APIs or AWS CLI commands.

For more information about APIs related to controls and standards, see the AWS Security Hub API Reference.

Security checks and security scores in Security Hub

For each control that you enable, AWS Security Hub runs security checks. A security check determines whether your AWS resources are 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 the section called “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. For more information, see the section called “AWS Config rules and security checks”. Others use custom Lambda functions, which are managed by Security Hub and are not visible to customers.

As Security Hub runs security checks, it generates findings and assigns them a compliance status. For more information about compliance status, see Values for compliance status of a finding.

Security Hub uses the compliance status of control findings to determine an overall control status. Security Hub 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 “Determining security scores”.

If you've turned on consolidated control findings, Security Hub generates a single finding even when a control is associated with more than one standard. For more information, see Consolidated control findings.

Topics

How Security Hub uses AWS Config rules to run security checks

To run security checks on your environment's resources, AWS Security Hub either uses steps specified by the standard, or uses specific AWS Config rules. Some rules are managed rules, which are managed by AWS Config. Other rules are custom rules that Security Hub develops.

AWS Config rules that Security Hub uses for controls are referred to as service-linked rules, because they are enabled and controlled by the Security Hub service.

To enable checks against these AWS Config rules, you must first enable AWS Config for your account and enable resource recording for required resources. For information about how to enable AWS Config, see Configuring AWS Config. For information about required resource recording, see AWS Config resources required to generate control findings

How Security Hub generates the service-linked rules

For every control that uses an AWS Config service-linked rule, Security Hub creates instances of the required rules in your AWS environment.

These service-linked rules are specific to Security Hub. It 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 original AWS Config managed rule `vpc-flow-logs-enabled`, the service-linked rule name would be something like `securityhub-vpc-flow-logs-enabled-12345`.

There are limits on the number of AWS Config rules that can be used to evaluate controls. Custom AWS Config rules that Security Hub creates don't count towards that limit. You can enable a security standard even if you've already reached the AWS Config limit for managed rules in your account. To learn more about AWS Config rule limits, see Service Limits in the *AWS Config Developer Guide*.

Viewing details about the AWS Config rules for controls

For controls that use AWS Config managed rules, the control description includes a link to the AWS Config rule details. Custom rules aren't linked from the control description. For control descriptions, see Security Hub controls reference. Select a control from the list to see its description.

For findings generated from those controls, the finding details include a link to the associated AWS Config rule. Note that to navigate to the AWS Config rule from finding details, you must also have an IAM permission in the selected account to navigate to AWS Config.

The finding details on the **Findings** page, **Insights** page, and **Integrations** page include a **Rules** link to the AWS Config rule details. See Reviewing finding details.

On the control details page, the **Investigate** column of the finding list contains a link to the AWS Config rule details. See Viewing the AWS Config rule for a finding resource.

AWS Config resources required to generate control findings

AWS Security Hub generates control findings by performing security checks against Security Hub controls. Some controls use AWS Config rules that evaluate compliance with specific resources. For Security Hub to generate findings for controls that have a *change triggered* schedule type, you must turn on recording for required resources in AWS Config. You don't need to record resources for most controls that have a *periodic* schedule type. However, some periodic controls require resource recording to detect changes in compliance.

This page provides a list of required resources across standards and a list of required resources divided by standard. The first table also lists which Security Hub controls use each resource.

If a finding is generated by a security check that is based on an AWS Config rule, the finding details include a **Rules** link to the associated AWS Config rule. To navigate to the AWS Config rule, your account must have IAM permissions to view AWS Config rules.

> **Note**
>
> In AWS Regions where a control isn't available, the corresponding resource isn't available in AWS Config. For a list of Regional limits on Security Hub controls, see Availability of controls by Region.

AWS Config resources required for all controls

For Security Hub to generate findings for enabled Security Hub change triggered controls that use a AWS Config rule, you must record these resources in AWS Config. This table also indicates which controls require a particular resource. A control may require more than one resource.

Service	Required resource	Related controls
Amazon API Gateway	`AWS::ApiGateway::Stage`	APIGateway.1 APIGateway.2 APIGateway.3 APIGateway.4 APIGateway.5

Service	Required resource	Related controls
	`AWS::ApiGatewayV2::Stage`	APIGateway.1 APIGateway.9
AWS AppSync	`AWS::AppSync::GraphQLApi`	AppSync.2 AppSync.5
AWS Backup (AWS Backup)	`AWS::Backup::RecoveryPoint`	Backup.1
AWS Certificate Manager (ACM)	`AWS::ACM::Certificate`	ACM.1 ACM.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
Amazon CloudWatch	`AWS::CloudWatch::Alarm`	CloudWatch.15 CloudWatch.17

Service	Required resource	Related controls
AWS CodeBuild	`AWS::CodeBuild::Project`	CodeBuild.1 CodeBuild.2 CodeBuild.3 CodeBuild.4
AWS Database Migration Service (AWS DMS)	`AWS::DMS::Endpoint`	DMS.9
	`AWS::DMS::ReplicationInstance`	DMS.6
	`AWS::DMS::ReplicationTask`	DMS.7 DMS.8
Amazon DynamoDB	`AWS::DynamoDB::Table`	DynamoDB.2 DynamoDB.6
Amazon Elastic Compute Cloud (EC2)	`AWS::EC2::ClientVpnEndpoint`	EC2.51
	`AWS::EC2::EIP`	EC2.12

Service	Required resource	Related controls
	AWS::EC2::Instance	EC2.4 EC2.8 EC2.9 EC2.17 EC2.24 EMR.1 SSM.1
	AWS::EC2::LaunchTemplate	EC2.25
	AWS::EC2::NetworkAcl	EC2.16 EC2.21
	AWS::EC2::NetworkInterface	EC2.22
	AWS::EC2::SecurityGroup	EC2.2 EC2.13 EC2.14 EC2.18 EC2.19

Service	Required resource	Related controls
	AWS::EC2::Subnet	EC2.15 ElastiCache.7 Lambda.5
	AWS::EC2::TransitGateway	EC2.23
	AWS::EC2::VPNConnection	EC2.20
	AWS::EC2::Volume	EC2.3
Amazon EC2 Auto Scaling	AWS::AutoScaling::AutoScalingGroup	AutoScaling.1 AutoScaling.2 AutoScaling.6 AutoScaling.9
	AWS::AutoScaling::LaunchConfiguration	AutoScaling.3 Autoscaling.5
Amazon EC2 Systems Manager (SSM)	AWS::SSM::AssociationCompliance	SSM.3
	AWS::SSM::ManagedInstanceInventory	SSM.1

Service	Required resource	Related controls
	`AWS::SSM::PatchCompliance`	SSM.2
Amazon Elastic Container Registry (Amazon ECR)	`AWS::ECR::Repository`	ECR.2 ECR.3
Amazon Elastic Container Service (Amazon ECS)	`AWS::ECS::Cluster`	ECS.12
	`AWS::ECS::Service`	ECS.2 ECS.10
	`AWS::ECS::TaskDefinition`	ECS.1 ECS.3 ECS.4 ECS.5 ECS.8 ECS.9
Amazon Elastic File System (Amazon EFS)	`AWS::EFS::AccessPoint`	EFS.3 EFS.4
Amazon Elastic Kubernetes Service (Amazon EKS)	`AWS::EKS::Cluster`	EKS.2

Service	Required resource	Related controls
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::LoadBalancer`	ELB.4 ELB.5 ELB.6 ELB.12 ELB.13 ELB.16

Service	Required resource	Related controls
ElasticSearch	`AWS::Elasticsearch::Domain`	ES.3 ES.4 ES.5 ES.6 ES.7 ES.8
Amazon EventBridge	`AWS::Events::EventBus`	EventBridge.3
	`AWS::Events::Endpoint`	EventBridge.4
`Amazon FSx`	`AWS::FSx::FileSystem`	FSx.1
AWS Identity and Access Management (IAM)	`AWS::IAM::Group`	IAM.18 KMS.2
	`AWS::IAM::Policy`	IAM.1 IAM.21 KMS.1
	`AWS::IAM::Role`	IAM.18 KMS.2
	`AWS::IAM::User`	IAM.2 IAM.18 KMS.2

Service	Required resource	Related controls
AWS Key Management Service (AWS KMS)	`AWS::KMS::Key`	KMS.3
Amazon Kinesis	`AWS::Kinesis::Stream`	Kinesis.1
AWS Lambda	`AWS::Lambda::Function`	Lambda.1 Lambda.2 Lambda.3 Lambda.5
Amazon MSK	`AWS::MSK::Cluster`	MSK.1 MSK.2
Amazon MQ	`AWS::AmazonMQ::Broker`	MQ.5 MQ.6
AWS Network Firewall	`AWS::NetworkFirewall::Firewall`	NetworkFirewall.1 NetworkFirewall.9
	`AWS::NetworkFirewall::FirewallPolicy`	NetworkFirewall.3 NetworkFirewall.4 NetworkFirewall.5
	`AWS::NetworkFirewall::RuleGroup`	NetworkFirewall.6

Service	Required resource	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.10

Service	Required resource	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.34 RDS.35

Service	Required resource	Related controls
	AWS::RDS::DBClusterSnapshot	DocumentDB.3 Neptune.3 Neptune.6 RDS.1 RDS.4
	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
	AWS::RDS::DBSnapshot	DocumentDB.3 RDS.1 RDS.4

Service	Required resource	Related controls
	`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
Amazon Route 53	`AWS::Route53::HostedZone`	Route53.2
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::AccessPoint`	S3.19

Service	Required resource	Related controls
	`AWS::S3::Bucket`	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
Amazon Simple Notification Service (Amazon SNS)	`AWS::SNS::Topic`	SNS.1
Amazon Simple Queue Service (Amazon SQS)	`AWS::SQS::Queue`	SQS.1

Service	Required resource	Related controls
Amazon SageMaker	`AWS::SageMaker::NotebookInstance`	SageMaker.2 SageMaker.3
AWS Secrets Manager	`AWS::SecretsManager::Secret`	SecretsManager.1 SecretsManager.2
AWS Step Functions	`AWS::StepFunctions::StateMachine`	StepFunctions.1
AWS WAF	`AWS::WAF::Rule`	WAF.6
	`AWS::WAF::RuleGroup`	WAF.7
	`AWS::WAF::WebACL`	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

AWS Config resources required for FSBP standard

For Security Hub to accurately report findings for enabled AWS Foundational Security Best Practices (FSBP) change triggered controls that use a AWS Config rule, you must record these resources in AWS Config. For more information about this standard, see AWS Foundational Security Best Practices (FSBP) standard.

Service	Required resources
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`
Amazon CloudFront	`AWS::CloudFront::Distribution`
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 EC2 Systems Manager (SSM)	`AWS::SSM::AssociationCompliance`
	`AWS::SSM::ManagedInstanceInventory`
	`AWS::SSM::PatchCompliance`

Amazon Elastic Compute Cloud (EC2)	`AWS::EC2::ClientVpnEndpoint` `AWS::EC2::Instance`

Service	Required resources
	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 EKS	AWS::EKS::Cluster
ElasticBeanstalk	AWS::ElasticBeanstalk::Environment

Service	Required resources
Elastic Load Balancing	`AWS::ElasticLoadBalancing::LoadBalancer` `AWS::ElasticLoadBalancingV2::LoadBalancer`
ElasticSearch	`AWS::Elasticsearch::Domain`
Amazon FSx	`AWS::FSx::FileSystem`

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::Key`
Amazon Kinesis	`AWS::Kinesis::Stream`
AWS Lambda	`AWS::Lambda::Function`
Amazon MSK	`AWS::MSK::Cluster`

AWS Network Firewall	`AWS::NetworkFirewall::Firewall`
	`AWS::NetworkFirewall::FirewallPolicy`
	`AWS::NetworkFirewall::RuleGroup`

Amazon OpenSearch Service	`AWS::OpenSearch::Domain`

Service	Required resources

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 Route 53	`AWS::Route53::HostedZone`
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::AccessPoint` `AWS::S3::Bucket`
Amazon Simple Queue Service (Amazon SQS)	`AWS::SQS::Queue`
Amazon SageMaker	`AWS::SageMaker::NotebookInstance`
AWS Secrets Manager	`AWS::SecretsManager::Secret`
AWS Step Functions	`AWS::StepFunctions::StateMachine`

Service	Required resources

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`

AWS Config resources required for CIS AWS Foundations Benchmark

To run security checks for enabled controls that apply to the Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0 and v1.4.0, Security Hub either runs through the exact audit steps prescribed for the checks in Securing Amazon Web Services or uses specific AWS Config managed rules.

For more information about this standard, see Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0 and v1.4.0.

Required AWS Config resources for CIS v1.4.0

For Security Hub to accurately report findings for enabled CIS v1.4.0 change triggered controls that use a AWS Config rule, you must record these resources in AWS Config.

Service	Required resources
Amazon Elastic Compute Cloud (EC2)	`AWS::EC2::NetworkAcl` `AWS::EC2::SecurityGroup`
AWS Identity and Access Management (IAM)	`AWS::IAM::Policy`

Service	Required resources
	`AWS::IAM::User`
Amazon Relational Database Service (Amazon RDS)	`AWS::RDS::DBInstance`
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::Bucket`

Required AWS Config resources for CIS v1.2.0

For Security Hub to accurately report findings for enabled CIS v1.2.0 change triggered controls that use a AWS Config rule, you must record these resources in AWS Config.

Service	Required resources
Amazon Elastic Compute Cloud (EC2)	`AWS::EC2::SecurityGroup`
AWS Identity and Access Management (IAM)	`AWS::IAM::Policy` `AWS::IAM::User`

AWS Config resources required for NIST SP 800-53 Rev. 5

For Security Hub to accurately report findings for enabled National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5 change triggered controls that use a AWS Config rule, you must record these resources in AWS Config. You only have to record resources for controls that have a schedule type of *change triggered*. For more information about this standard, see National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5.

Service	Required resources
Amazon API Gateway	`AWS::ApiGateway::Stage` `AWS::ApiGatewayV2::Stage`
AWS AppSync	`AWS::AppSync::GraphQLApi`

Service	Required resources
AWS Backup	`AWS::Backup::RecoveryPoint`
AWS Certificate Manager (ACM)	`AWS::ACM::Certificate`
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 (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`

Service	Required resources
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`
Amazon EventBridge	`AWS::Events::Endpoint` `AWS::Events::EventBus`
Amazon FSx	`AWS::FSx::FileSystem`

Service	Required resources

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::Key`
Amazon Kinesis	`AWS::Kinesis::Stream`
AWS Lambda	`AWS::Lambda::Function`
Amazon MSK	`AWS::MSK::Cluster`
Amazon MQ	`AWS::AmazonMQ::Broker`

AWS Network Firewall	`AWS::NetworkFirewall::Firewall`
	`AWS::NetworkFirewall::Firew allPolicy`
	`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 Route 53	`AWS::Route53::HostedZone`

Service	Required resources
Amazon Simple Storage Service (Amazon S3)	`AWS::S3::AccessPoint` `AWS::S3::Bucket`
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	`AWS::SageMaker::NotebookInstance`
AWS Secrets Manager	`AWS::SecretsManager::Secret`

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`

AWS Config resources required for PCI DSS

For Security Hub to accurately report findings for enabled Payment Card Industry Data Security Standard (PCI DSS) controls that use a AWS Config rule, you must record these resources in AWS Config. For more information about this standard, see Payment Card Industry Data Security Standard (PCI DSS).

Service	Required resources
AWS CodeBuild	`AWS::CodeBuild::Project`

Amazon Elastic Compute Cloud (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`

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::Bucket`
Amazon EC2 Systems Manager (SSM)	`AWS::SSM::AssociationCompliance`

Service	Required resources
	`AWS::SSM::ManagedInstanceInventory` `AWS::SSM::PatchCompliance`

AWS Config resources required for Service-Managed Standard: AWS Control Tower

For Security Hub to accurately report findings for enabled Service-Managed Standard: AWS Control Tower change triggered controls that use a AWS Config rule, you must record the following resources in AWS Config. For more information about this standard, see Service-Managed Standard: AWS Control Tower.

Service	Required resources
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::EC2::Volume`

Service	Required resources
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 Key Management Service (AWS KMS)	`AWS::KMS::Key`

Service	Required resources
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::Bucket`
Amazon Simple Queue Service (Amazon SQS)	`AWS::SQS::Queue`

Amazon EC2 Systems Manager (SSM)	`AWS::SSM::AssociationCompliance`
	`AWS::SSM::ManagedInstanceInventory`
	`AWS::SSM::PatchCompliance`

AWS Secrets Manager	`AWS::SecretsManager::Secret`

Service	Required resources

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 begins to run all checks within two hours. Most checks begin to run within 25 minutes. Security Hub 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, Security Hub may take up to 24 hours 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 Lambda.1 in the AWS Foundational Security Best Practices (FSBP) standard, Security Hub will create the service-linked rule and typically generate findings in minutes. After this, if you enable Lambda.1 in the Payment Card Industry Data Security Standard (PCI DSS), Security Hub may take up to 24 hours to generate findings for this control because it uses the same service-linked rule as Lambda.1.

After the initial check, the schedule for each control can be either periodic or change triggered.

- **Periodic checks** – These checks run automatically within 12 or 24 hours after the most recent run. Security Hub 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 **KMS.4 - AWS KMS key rotation should be enabled**, and then remediate the finding, Security Hub 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`.
- **Change-triggered checks** – These 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 findings until a 24-hour period is complete. Regardless of your chosen recording period, Security Hub checks every 18 hours to ensure no resource updates from AWS Config were missed.

In general, Security Hub uses change-triggered rules whenever possible. For a resource to use a change-triggered rule, it must support AWS Config configuration items.

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 includes whether the rule is change triggered or periodic.

Checks that use Security Hub custom Lambda functions are periodic.

Generating and updating control findings

AWS Security Hub generates findings by running checks against security controls. These findings use the AWS Security Finding Format (ASFF). Note that if the finding size exceeds the maximum of 240 KB, then the `Resource.Details` object is removed. For controls that are backed by AWS Config resources, you can view the resource details on the AWS Config console.

Security Hub normally charges for each security check for a control. However, if multiple controls use the same AWS Config rule, then Security Hub only charges once for each check against the AWS Config rule. If you turn on consolidated control findings, Security Hub generates a single finding for a security check even when the control is included in multiple enabled standards.

For example, the AWS Config rule `iam-password-policy` is used by multiple controls in the Center for Internet Security (CIS) AWS Foundations Benchmark standard and the Foundational Security Best Practices standard. Each time Security Hub runs a check against that AWS Config rule, it generates a separate finding for each related control, but only charges once for the check.

Consolidated control findings

When consolidated control findings is turned on in your account, Security Hub generates a single new finding or finding update for each security check of a control, even if a control applies to multiple enabled standards. To see a list of controls and the standards they apply to, see Security Hub controls reference. You can turn consolidated control findings on or off. We recommend turning it on to reduce finding noise.

If you enabled Security Hub for an AWS account before February 23, 2023, you must turn on consolidated control findings by following the instructions later in this section. If you enable

Security Hub on or after February 23, 2023, consolidated control findings is automatically turned on in your account. However, if you use the Security Hub integration with AWS Organizations or invited member accounts through a manual invitation process, consolidated control findings is turned on in member accounts only if it's turned on in the administrator account. If the feature is turned off in the administrator account, it's turned off in member accounts. This behavior applies to new and existing member accounts.

If you turn off consolidated control findings in your account, Security Hub generates a separate finding per security check for each enabled standard that includes a control. For example, if four enabled standards share a control with the same underlying AWS Config rule, you receive four separate findings after a security check of the control. If you turn on consolidated control findings, you receive only one finding. For more information about how consolidation affects your findings, see Sample control findings.

When you turn on consolidated control findings, Security Hub creates new standard-agnostic findings and archives the original standard-based findings. Some control finding fields and values will change and may impact existing workflows. For more information about these changes, see Consolidated control findings – ASFF changes.

Turning on consolidated control findings may also affect findings that third-party integrations receive from Security Hub. Automated Security Response on AWS v2.0.0 supports consolidated control findings.

Turning on consolidated control findings

To turn on consolidated control findings, you must be signed in to an administrator account or a standalone account.

> **Note**
>
> After turning on consolidated control findings, it may take up to 24 hours for Security Hub for generate new, consolidated findings and archive the original, standard-based findings. During that time, you may see a mix of standard-agnostic and standard-based findings in your account.

Security Hub console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

2. In the navigation pane, choose **Settings**.
3. Choose the **General** tab.
4. For **Controls**, turn on **Consolidated control findings**.
5. Choose **Save**.

Security Hub API

1. Run UpdateSecurityHubConfiguration.
2. Set `ControlFindingGenerator` equal to `SECURITY_CONTROL`.

   Example request:

```
{
   "ControlFindingGenerator": "SECURITY_CONTROL"
}
```

AWS CLI

1. Run the `update-security-hub-configuration` command.
2. Set `control-finding-generator` equal to `SECURITY_CONTROL`.

```
aws securityhub  --region us-east-1 update-security-hub-configuration --control-finding-generator SECURITY_CONTROL
```

Turning off consolidated control findings

To turn off consolidated control findings, you must be signed in to an administrator account or a standalone account.

> **Note**
>
> After turning off consolidated control findings, it may take up to 24 hours for Security Hub for generate new, standard-based findings and archive the consolidated findings. During that time, you may see a mix of standard-based and consolidated findings in your account.

Security Hub console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. In the navigation pane, choose **Settings**.
3. Choose the **General** tab.
4. For **Controls**, choose **Edit** and turn off **Consolidated control findings**.
5. Choose **Save**.

Security Hub API

1. Run `UpdateSecurityHubConfiguration`.
2. Set `ControlFindingGenerator` equal to `STANDARD_CONTROL`.

   Example request:

```
{
   "ControlFindingGenerator": "STANDARD_CONTROL"
}
```

AWS CLI

1. Run the `update-security-hub-configuration` command.
2. Set `control-finding-generator` equal to `STANDARD_CONTROL`.

```
aws securityhub  --region us-east-1 update-security-hub-configuration --control-finding-generator STANDARD_CONTROL
```

`Compliance` details for control findings

For findings generated by security checks of controls, the `Compliance` field in the AWS Security Finding Format (ASFF) contains details related to control findings. The `Compliance` field includes the following information.

`AssociatedStandards`

The enabled standards that a control is enabled in.

`RelatedRequirements`

The list of related requirements for the control in all enabled standards. The requirements are from the third-party security framework for the control, such as the Payment Card Industry Data Security Standard (PCI DSS).

`SecurityControlId`

The identifier for a control across security standards that Security Hub supports.

`Status`

The result of the most recent check that Security Hub ran for a given control. The results of the previous checks are kept in an archived state for 90 days.

`StatusReasons`

Contains a list of reasons for the value of `Compliance.Status`. For each reason, `StatusReasons` includes the reason code and a description.

The following table lists the available status reason codes and descriptions. The remediation steps depend on which control generated a finding with the reason code. Choose a control from the Security Hub controls reference to see remediation steps for that control.

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.
`CLOUDTRAIL_REGION_INVAILD`	`WARNING`	Multi-Region CloudTrail trails are not in the current Region.

Reason code	Compliance.Status	Description
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.

Reason code	Compliance.Status	Description
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 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

When Security Hub runs security checks and generates control findings, the `ProductFields` attribute in ASFF includes the following fields:

`ArchivalReasons:0/Description`

Describes why Security Hub has archived existing findings.

For example, Security Hub archives existing findings when you disable a control or standard and when you turn consolidated control findings on or off.

`ArchivalReasons:0/ReasonCode`

Provides the reason why Security Hub has archived existing findings.

For example, Security Hub archives existing findings when you disable a control or standard and when you turn consolidated control findings on or off.

`StandardsGuideArn` or `StandardsArn`

The ARN of the standard associated with the control.

For the CIS AWS Foundations Benchmark standard, the field is `StandardsGuideArn`.

For PCI DSS and AWS Foundational Security Best Practices standards, the field is `StandardsArn`.

These fields are removed in favor of `Compliance.AssociatedStandards` if you turn on 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 turn on consolidated control findings.

`RuleId` or `ControlId`

The identifier of the control.

For the CIS AWS Foundations Benchmark standard, the field is `RuleId`.

For other standards, the field is `ControlId`.

These fields are removed in favor of `Compliance.SecurityControlId` if you turn on consolidated control findings.

`RecommendationUrl`

The URL to the remediation information for the control. This field is removed in favor of `Remediation.Recommendation.Url` if you turn on 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 turn on consolidated control findings.

`aws/securityhub/ProductName`

For control-based findings, the product name is Security Hub.

`aws/securityhub/CompanyName`

For control-based findings, the company name is AWS.

`aws/securityhub/annotation`

A description of the issue uncovered by the control.

`aws/securityhub/FindingId`

The identifier of the finding. This field doesn't reference a standard if you turn on consolidated control findings.

Assigning severity to control findings

The severity assigned to a Security Hub control identifies 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 nonproduction 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 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 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.

Rules for updating control findings

A subsequent check against a given rule might generate a new result. For example, the status of "Avoid the use of the root user" could change from `FAILED` to `PASSED`. In that case, a new finding is generated that contains the most recent result.

If a subsequent check against a given rule generates a result that is identical to the current result, the existing finding is updated. No new finding is generated.

Security Hub automatically archives findings from controls if the associated resource is deleted, the resource does not exist, or the control is disabled. A resource might no longer exist because the associated service is not currently used. The findings are archived automatically based on one of the following criteria:

- The finding is not updated for three to five days (note that this is best effort and not guaranteed).

- The associated AWS Config evaluation returned `NOT_APPLICABLE`.

Compliance status and control status

The `Compliance.Status` field of the AWS Security Finding Format describes the result of a control finding. Security Hub 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 console.

For an administrator account, the control status reflects the control status in 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 typically generates the initial control status within 30 minutes after your first visit to the **Summary** page or **Security standards** page of the Security Hub 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 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**
>
> 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) Region.

Values for compliance status of a finding

The compliance status for each finding is assigned one of the following values:

- `PASSED` – Automatically sets the Security Hub `Workflow.Status` to `RESOLVED`.

  If `Compliance.Status` for a finding changes from `PASSED` to `FAILED`, `WARNING`, or `NOT_AVAILABLE`; and `Workflow.Status` was either `NOTIFIED` or `RESOLVED`; then Security Hub automatically sets `Workflow.Status` to `NEW`.

If you don't have resources corresponding to a control, Security Hub produces a PASSED finding at the account level. If you have a resource corresponding to a control but then delete the resource, Security Hub creates a NOT_AVAILABLE finding and archives it immediately. After 18 hours, you receive a PASSED finding since you no longer have resources corresponding to the control.

- FAILED – Indicates that the control didn't pass the security check for this finding.
- WARNING – Indicates that the check was completed, but Security Hub can't determine whether the resource is in a PASSED or FAILED state.
- 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 automatically archives the finding.

Values for control status

Security Hub derives an overall control status from the compliance status of the control findings. When determining control status, Security Hub 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 starts to generate findings for it. A control also has this status if all of the findings are SUPPRESSED or if it's unavailable in the current 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.

Determining security scores

The **Summary** page and **Controls** page of the Security Hub console display a summary security score across all of your enabled standards. On the **Security standards** page, Security Hub also displays a security score from 0–100 percent for each enabled standard.

When you first enable Security Hub, Security Hub calculates the summary security score and standard security scores within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub console. Scores are only generated for standards that are enabled when you visit those pages. To view a list of standards that are currently enabled, invoke the `GetEnabledStandards` API operation. In addition, AWS Config resource recording must be configured for scores to appear. The summary security score is the average of the standard security scores.

After first-time score generation, Security Hub updates security scores every 24 hours. Security Hub displays a timestamp to indicate when a security score was last updated.

> **Note**
>
> It may take up to 24 hours for first-time security scores to be generated in the China Regions and AWS GovCloud (US) Region.

If you turn on consolidated control findings, it may 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 may take up to 24 hours for Security Hub to generate new security scores that include data from the updated Regions.

How security scores are calculated

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 calculates a summary security score across all of your enabled standards. Security Hub 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 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 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%
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 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 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 may have a summary score that differs from the standard security score even if you've only enabled one standard in your account in the current Region. This may occur if you're signed in to an administrator account and member accounts have additional standards or different standards enabled. This may 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.

Security Hub standards reference

AWS Security Hub currently supports the security standards detailed in this section.

Choose a standard to view more details about it and the controls that apply to it.

Security Hub standards and controls don't guarantee compliance with any regulatory frameworks or audits. Rather, the controls provide a way to monitor the current state of your AWS accounts and resources.

Supported standards

- AWS Foundational Security Best Practices (FSBP) standard
- Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0 and v1.4.0
- National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5
- Payment Card Industry Data Security Standard (PCI DSS)
- Service-managed standards

AWS Foundational Security Best Practices (FSBP) standard

The AWS Foundational Security Best Practices standard is a set of controls that detect when your AWS accounts and resources deviate from security best practices.

The standard lets you continuously evaluate all of your AWS accounts and workloads to quickly identify areas of deviation from best practices. It provides actionable and prescriptive guidance about how to improve and maintain your organization's security posture.

The controls include security best practices for resources from multiple AWS services. Each control is also assigned a category that reflects the security function that it applies to. For more information, see the section called "Control categories".

Controls that apply to the FSBP standard

[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.2] AWS AppSync should have field-level logging enabled

[AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys

[AutoScaling.1] Auto Scaling groups associated with a Classic Load Balancer should use load balancer 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

[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

[Config.1] AWS Config should be 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

[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.6] DynamoDB tables should have deletion protection enabled

[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

[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

[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

[EKS.8] EKS clusters should have audit logging enabled

[ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled

[ElastiCache.2] ElastiCache for Redis cache clusters should have auto minor version upgrade enabled

[ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled

[ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest

[ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit

[ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH

[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 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

[EMR.2] Amazon EMR block public access setting should be enabled

[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

[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

[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

[MSK.1] MSK clusters should be encrypted in transit among broker nodes

[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

[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 AWS 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] RDS DB clusters should be configured to copy tags to 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.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

[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

[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

[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

[StepFunctions.1] Step Functions state machines should have logging turned on

[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

Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0 and v1.4.0

The 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 supports CIS AWS Foundations Benchmark v1.2.0 and v1.4.0.

This page lists security control IDs and titles. In the AWS GovCloud (US) Region and China Regions, standard-specific control IDs and titles are used. For a mapping of security control IDs and titles to standard-specific control IDs and titles, see How consolidation impacts control IDs and titles.

Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0

Security Hub 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 v1.2.0

[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 AWS 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

[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

Center for Internet Security (CIS) AWS Foundations Benchmark v1.4.0

Security Hub supports v1.4.0 of the CIS AWS Foundations Benchmark.

Controls that apply to CIS AWS Foundations Benchmark v1.4.0

[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 AWS 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

[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 v1.2.0 compared to v1.4.0

This section summarizes the differences between the Center for Internet Security (CIS) AWS Foundations Benchmark v1.4.0 and v1.2.0. Security Hub supports both versions of this standard.

> **Note**
>
> We recommend upgrading to CIS AWS Foundations Benchmark v1.4.0 to stay current on security best practices, but you may have both v1.4.0 and v1.2.0 enabled at the same time. For more information, see Enabling and disabling security standards. If you want to upgrade to v1.4.0, it's best to enable v1.4.0 first before disabling v1.2.0. If you use the Security Hub integration with AWS Organizations to centrally manage multiple accounts

and you want to batch enable v1.4.0 across all of them (and optionally disable v1.2.0), you can run a Security Hub multi-account script from the administrator account.

Controls that exist in CIS AWS Foundations Benchmark v1.4.0, but not in v1.2.0

The following controls were added in CIS AWS Foundations Benchmark v1.4.0. These controls are *not* included in CIS AWS Foundations Benchmark v1.2.0.

Security control ID	CISv1.4.0 requirement	Control title
EC2.7	2.2.1	Ensure EBS volume encryption is enabled
EC2.21	5.1	Ensure no Network ACLs allow ingress from 0.0.0.0/0 to remote server administration ports
IAM.22	1.12	Ensure credentials unused for 45 days or greater are disabled
RDS.3	2.3.1	Ensure that encryption is enabled for RDS Instances
S3.1	2.1.5.1	S3 Block Public Access setting should be enabled
S3.5	2.1.2	Ensure S3 bucket policy is set to deny HTTP requests
S3.8	2.1.5.2	S3 Block Public Access setting should be enabled at the bucket level
S3.20	2.1.3	S3 general purpose buckets should have MFA delete enabled

Controls that exist in CIS AWS Foundations Benchmark v1.2.0, but not in v1.4.0

The following controls exist only in CIS AWS Foundations Benchmark v1.2.0. These controls are *not* included in CIS AWS Foundations Benchmark v1.4.0.

Security control ID	CISv1.2.0 requirement	Control title	Reason not included in v1.4.0
CloudWatch.2	3.1	Ensure a log metric filter and alarm exist for unauthorized API calls	Automated check that Security Hub doesn't support
CloudWatch.3	3.2	Ensure a log metric filter and alarm exist for AWS Management Console sign-in without MFA	Automated check that Security Hub doesn't support
EC2.13	4.1	Ensure no security groups allow ingress from 0.0.0.0/0 to port 22	See instead, [EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389
EC2.14	4.2	Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389	Automated check that Security Hub doesn't support
IAM.2	1.16	IAM users should not have IAM policies attached	Automated check that Security Hub doesn't support
IAM.8	1.3	Ensure credentials unused for 90 days or greater are disabled	See instead, [IAM.22] IAM user credentials unused for 45 days should be removed

Security control ID	CISv1.2.0 requirement	Control title	Reason not included in v1.4.0
IAM.11	1.5	Ensure IAM password policy requires at least one uppercase letter	Not a requirement in CISv1.4.0
IAM.12	1.6	Ensure IAM password policy requires at least one lowercase letter	Not a requirement in CISv1.4.0
IAM.13	1.7	Ensure IAM password policy requires at least one symbol	Not a requirement in CISv1.4.0
IAM.14	1.8	Ensure IAM password policy requires at least one number	Not a requirement in CISv1.4.0
IAM.17	1.11	Ensure IAM password policy expires passwords within 90 days or less	Not a requirement in CISv1.4.0
IAM.20	1.1	Avoid the user of the root user	See instead, [CloudWatch.1] A log metric filter and alarm should exist for usage of the "root" user

Controls that exist in CIS AWS Foundations Benchmark v1.2.0 and v1.4.0

The following controls exist in both CIS AWS Foundations Benchmark v1.2.0 and v1.4.0. However, the controls IDs and some of the control titles differ in each version.

Security control ID	CISv1.2.0 requirement	Control title in CISv1.2.0	CISv1.4.0 requirement	Control title in CISv1.4.0
CloudTrail.1	2.1	Ensure CloudTrail is enabled in all Regions	3.1	Ensure CloudTrail is enabled in all Regions
CloudTrail.2	2.7	Ensure CloudTrail logs are encrypted at rest using AWS KMS keys	3.7	Ensure CloudTrail logs are encrypted at rest using AWS KMS keys
CloudTrail.4	2.2	Ensure CloudTrail log file validation is enabled	3.2	Ensure CloudTrail log file validation is enabled
CloudTrail.5	2.4	Ensure CloudTrail trails are integrated with CloudWatch Logs	3.4	Ensure CloudTrail trails are integrated with CloudWatch Logs
CloudTrail.6	2.3	Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible	3.3	Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
CloudTrail.7	2.6	Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket	3.6	Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket

Security control ID	CISv1.2.0 requirement	Control title in CISv1.2.0	CISv1.4.0 requirement	Control title in CISv1.4.0
CloudWatch.1	1.1 3.3	**1.1** – Avoid the use of the root user **3.3** – Ensure a log metric filter and alarm exist for usage of root user	1.7	Eliminate use of the root user for administrative and daily tasks
CloudWatch.4	3.4	Ensure a log metric filter and alarm exist for IAM policy changes	4.4	Ensure a log metric filter and alarm exist for IAM policy changes
CloudWatch.5	3.5	Ensure a log metric filter and alarm exist for CloudTrail configuration change	4.5	Ensure a log metric filter and alarm exist for CloudTrail configuration change
CloudWatch.6	3.6	Ensure a log metric filter and alarm exist for AWS Managemen t Console authentication failures	4.6	Ensure a log metric filter and alarm exist for AWS Managemen t Console authentication failures

Security control ID	CISv1.2.0 requirement	Control title in CISv1.2.0	CISv1.4.0 requirement	Control title in CISv1.4.0
CloudWatch.7	3.7	Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys	4.7	Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys
CloudWatch.8	3.8	Ensure a log metric filter and alarm exist for S3 bucket policy changes	4.8	Ensure a log metric filter and alarm exist for S3 bucket policy changes
CloudWatch.9	3.9	Ensure a log metric filter and alarm exist for AWS Config configuration changes	4.9	Ensure a log metric filter and alarm exist for AWS Config configuration changes
CloudWatch.10	3.10	Ensure a log metric filter and alarm exist for security group changes	4.10	Ensure a log metric filter and alarm exist for security group changes

Security control ID	CISv1.2.0 requirement	Control title in CISv1.2.0	CISv1.4.0 requirement	Control title in CISv1.4.0
CloudWatch.11	3.11	Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)	4.11	Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)
CloudWatch.12	3.12	Ensure a log metric filter and alarm exist for changes to network gateways	4.12	Ensure a log metric filter and alarm exist for changes to network gateways
CloudWatch.13	3.13	Ensure a log metric filter and alarm exist for route table changes	4.13	Ensure a log metric filter and alarm exist for route table changes
CloudWatch.14	3.14	Ensure a log metric filter and alarm exist for VPC changes	4.14	Ensure a log metric filter and alarm exist for VPC changes
Config.1	2.5	Ensure AWS Config is enabled	3.5	Ensure AWS Config is enabled in all Regions

Security control ID	CISv1.2.0 requirement	Control title in CISv1.2.0	CISv1.4.0 requirement	Control title in CISv1.4.0
EC2.2	4.3	Ensure the default security group of every VPC restricts all traffic	5.3	Ensure the default security group of every VPC restricts all traffic
EC2.6	2.9	Ensure VPC flow logging is enabled in all VPCs	3.9	Ensure VPC flow logging is enabled in all VPCs
IAM.1	1.22	Ensure IAM policies that allow full "*:*" administrative privileges are not created	1.16	Ensure IAM policies that allow full "*:*" administrative privileges are not attached
IAM.3	1.4	Ensure access keys are rotated every 90 days or less	1.14	Ensure access keys are rotated every 90 days or less
IAM.4	1.12	Ensure no root user access key exists	1.4	Ensure no root user account access key exists
IAM.5	1.2	Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password	1.10	Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password

Security control ID	CISv1.2.0 requirement	Control title in CISv1.2.0	CISv1.4.0 requirement	Control title in CISv1.4.0
IAM.6	1.14	Ensure hardware MFA is enabled for the root user	1.6	Ensure hardware MFA is enabled for the root user account
IAM.9	1.13	Ensure MFA is enabled for the root user	1.5	Ensure MFA is enabled for the root user account
IAM.15	1.9	Ensure IAM password policy requires minimum password length of 14 or greater	1.8	Ensure IAM password policy requires minimum length of 14 or greater
IAM.16	1.10	Ensure IAM password policy prevents password reuse	1.9	Ensure IAM password policy prevents password reuse
IAM.18	1.20	Ensure a support role has been created to manage incidents with AWS Support	1.17	Ensure a support role has been created to manage incidents with AWS Support
KMS.4	2.8	Ensure rotation for customer-created KMS keys is enabled	3.8	Ensure rotation for customer-created KMS keys is enabled

Finding fields format for CIS AWS Foundations Benchmark v1.4.0

When you enable CIS AWS Foundations Benchmark v1.4.0, you'll begin receiving findings in the AWS Security Finding Format (ASFF). For these findings, standard-specific fields will reference v1.4.0. For CIS AWS Foundations Benchmark v1.4.0, note the following format for `GeneratorID` and any ASFF fields that reference the standard Amazon Resource Name (ARN).

- **Standard ARN** – `arn:partition:securityhub:region:account-id:standards/cis-aws-foundations-benchmark/v/1.4.0`
- **`GeneratorID`** – `cis-aws-foundations-benchmark/v/1.4.0/control ID`

You can call the GetEnabledStandards API operation to find out the ARN of a standard.

> **Note**
>
> When you enable CIS AWS Foundations Benchmark v1.4.0, Security Hub may take up to 18 hours to generate findings for controls that use the same AWS Config service-linked rule as enabled controls in other enabled standards. For more information, see Schedule for running security checks.

Finding fields will differ if you've turned on consolidated control findings. For more information about these differences, see Impact of consolidation on ASFF fields and values. For sample CIS control findings with consolidation turned on and off, see Sample control findings.

CIS AWS Foundations Benchmark security checks that aren't supported in Security Hub

This section summarizes CIS requirements that are not currently supported in Security Hub. The Center for Internet Security (CIS) is an independent, nonprofit organization that establishes these requirements.

CIS AWS Foundations Benchmark v1.2.0 security checks that aren't supported in Security Hub

The following CIS AWS Foundations Benchmark v1.2.0 requirements are *not* currently supported in Security Hub.

Manual checks that aren't supported

Security Hub focuses on automated security checks. As a result, Security Hub doesn't support the following requirements of CIS AWS Foundations Benchmark v1.2.0 because they require manual checks of your resources:

- 1.15 – Ensure security questions are registered in the AWS account
- 1.17 – Maintain current contact details
- 1.18 – Ensure security contact information is registered
- 1.19 – Ensure IAM instance roles are used for AWS resource access from instances
- 1.21 – Do not setup access keys during initial user setup for all IAM users that have a console password
- 4.4 – Ensure routing tables for VPC peering are "least access"

Security Hub supports all automated checks for CIS AWS Foundations Benchmark v1.2.0.

CIS AWS Foundations Benchmark v1.4.0 security checks that aren't supported in Security Hub

The following CIS AWS Foundations Benchmark v1.4.0 requirements are *not* currently supported in Security Hub.

Manual checks that aren't supported

Security Hub focuses on automated security checks. As a result, Security Hub doesn't support the following requirements of CIS AWS Foundations Benchmark v1.4.0 because they require manual checks of your resources:

- 1.1 – Maintain current contact details
- 1.2 – Ensure security contact information is registered
- 1.3 – Ensure security questions are registered in the AWS account
- 1.11 – Do not setup access keys during initial user setup for all IAM users that have a console password
- 1.18 – Ensure IAM instance roles are used for AWS resource access from instances
- 1.21 – Ensure IAM users are managed centrally via identity federation or AWS Organizations for multi-account environments
- 2.1.4 – Ensure all data in Amazon S3 has been discovered, classified, and secured when required

- 5.4 – Ensure routing tables for VPC peering are "least access"

Automated checks that aren't supported

Security Hub doesn't support the following requirements of CIS AWS Foundations Benchmark v1.4.0 that rely on automated checks:

- 1.13 – Ensure there is only one active access key available for any single IAM user
- 1.15 – Ensure IAM users receive permissions only through groups
- 1.19 – Ensure that all the expired SSL/TLS certificates stored in IAM are removed
- 1.20 – Ensure that IAM Access Analyzer is enabled for all Regions
- 3.10 – Ensure that Object-level logging for write events is enabled for S3 buckets
- 3.11 – Ensure that Object-level logging for read events is enabled for S3 buckets
- 4.1 – Ensure a log metric filter and alarm exist for unauthorized API calls
- 4.2 – Ensure a log metric filter and alarm exist for Management Console sign-in without MFA
- 4.3 – Ensure a log metric filter and alarm exist for usage of root account (this is similar to automated requirement, **1.7 - Eliminate use of the root user for administrative and daily tasks**, which is supported in Security Hub)
- 4.15 – Ensure a log metric filter and alarm exists for AWS Organizations changes
- 5.2 – Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports

National Institute of Standards and Technology (NIST) SP 800-53 Rev. 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 is part of the U.S. Department of Commerce. This compliance framework helps you protect the availability, confidentiality, and integrity of your information systems and critical resources. U.S. federal government agencies and contractors must comply with NIST SP 800-53 to protect their systems, but private companies may voluntarily use it as a guiding framework for reducing cybersecurity risk.

Security Hub provides controls that support select NIST SP 800-53 requirements. These controls are evaluated through automated security checks. Security Hub controls don't support NIST SP 800-53 requirements that require manual checks. In addition, Security Hub controls only support the automated NIST SP 800-53 requirements that are listed as **Related requirements** in the details

for each control. Choose a control from the following list to see its details. Related requirements not mentioned in the control details are currently not supported by Security Hub.

Unlike other frameworks, NIST SP 800-53 isn't prescriptive about how its requirements should be evaluated. Instead, the framework provides guidelines, and the Security Hub NIST SP 800-53 controls represent the service's understanding of them.

If you use the Security Hub integration with AWS Organizations to centrally manage multiple accounts and you want to batch enable NIST SP 800-53 across all of them, you can run a Security Hub multi-account script from the administrator account.

For more information about NIST SP 800-53 Rev. 5, see the NIST Computer Security Resource Center.

Controls that apply to NIST SP 800-53 Rev. 5

[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 Classic Load Balancer should use load balancer 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

[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

[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

[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

[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

[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

[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

[EKS.8] EKS clusters should have audit logging enabled

[ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled

[ElastiCache.2] ElastiCache for Redis cache clusters should have auto minor version upgrade enabled

[ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled

[ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest

[ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit

[ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH

[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 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

[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.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

[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

[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.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

[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

[RDS.1] RDS snapshot should be private

[RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible AWS 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] RDS DB clusters should be configured to copy tags to 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.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs

[RDS.35] RDS DB clusters should have automatic minor version upgrade 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

[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

[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

[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

[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

Payment Card Industry Data Security Standard (PCI DSS)

The Payment Card Industry Data Security Standard (PCI DSS) in Security Hub provides a set of AWS security best practices for handling cardholder data. You can use this standard to discover security vulnerabilities in resources that handle cardholder data. Security Hub currently scopes the controls at the account level. We recommend that you enable these controls in all of your accounts that have resources that store, process, or transmit cardholder data.

This standard was validated by AWS Security Assurance Services LLC (AWS SAS), which is a team of Qualified Security Assessors (QSAs) certified to provide PCI DSS guidance, and assessments by the PCI DSS Security Standards Council (PCI SSC). AWS SAS has confirmed that the automated checks can assist a customer in preparing for a PCI DSS assessment.

This page lists security control IDs and titles. In the AWS GovCloud (US) Region and China Regions, standard-specific control IDs and titles are used. For a mapping of security control IDs and titles to standard-specific control IDs and titles, see How consolidation impacts control IDs and titles.

Controls that apply to PCI DSS

[AutoScaling.1] Auto Scaling groups associated with a Classic Load Balancer should use load balancer health checks

[CloudTrail.2] CloudTrail should have encryption at-rest enabled

[CloudTrail.3] CloudTrail 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

[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 AWS 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 AWS 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

Service-managed standards

A service-managed standard is a security standard that another AWS service manages. 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 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 console and isn't accessible by the Security Hub API or AWS CLI.
- **No automatic enablement of controls** – When you create a service-managed standard, Security Hub and the managing service don't automatically enable the controls that apply to the standard. In addition, when Security Hub releases new controls for the standard, they're not automatically enabled. This is a departure from standards that Security Hub manages. For more information about the usual way of configuring controls in Security Hub, see Viewing and managing security controls.
- **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 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 console, Security Hub 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 and AWS Control Tower. It lets you configure the proactive controls of AWS Control Tower alongside the detective controls of Security Hub 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 manages. For example, you must create and delete a service-managed standard in the managing service. For more information, see Service-managed standards.

In the Security Hub console and API, you can view Service-Managed Standard: AWS Control Tower alongside other Security Hub 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 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, AWS Control Tower also enables Security Hub for you.

If you haven't set up AWS Control Tower, you can't view or access this standard in the Security Hub console, Security Hub API, or AWS CLI. Even if you have set up AWS Control Tower, you can't view or access this standard in Security Hub 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 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.

However, disabling a control in Security Hub 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 (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 console, Security Hub 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 unless you explicitly enable that control in Security Hub.

Security Hub 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 Viewing details for a control.

Based on control statuses, Security Hub calculates a security score for Service-Managed Standard: AWS Control Tower. This score is only available in Security Hub. In addition, you can only view control findings in Security Hub. 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 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. 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 console, and you can no longer access it by using the Security Hub API or AWS CLI.

> **Note**
>
> Disabling all controls from the standard in Security Hub doesn't disable or delete the standard.

Disabling the Security Hub 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. Security Hub 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 Sample 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 from the following table 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, 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 Classic Load Balancer should use load balancer 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 for Redis replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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 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.4] Neptune DB clusters should have deletion protection enabled
- [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 AWS 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 controls in the *AWS Control Tower User Guide*.

Viewing and managing security standards

Security standards include a set of requirements to determine compliance with regulatory frameworks, industry best practices, or company policies. AWS Security Hub maps these requirements to controls and runs security checks on controls to assess whether the requirements of a standard are being met. A control may be enabled in one or more standards. If you turn on consolidated control findings, Security Hub generates a single finding per security check even when a control is part of multiple enabled standards. For more information, see Consolidated control findings.

For a list of available standards and the controls that apply to them, see Standards reference. The **Security standards** page on the Security Hub console also shows all of the supported security standards in Security Hub and their enablement status. For each security standard that's enabled in your account (or if you use the integration with AWS Organizations, in at least one account in your organization), you can view the following information:

- The enablement status of the standard in different Security Hub configuration policies if you use central configuration
- A description of any disabled standards
- A list of controls that are currently enabled in the standard and the overall status of those controls based on the compliance status of their findings
- a list of controls that apply to the standard but are currently disabled
- A security score for the standard

Security Hub generates a security score for each standard. Administrator accounts see aggregated security scores and control statuses across their member accounts. If you have set an aggregation Region, your security scores reflect the compliance status of controls across all linked Regions. For more information, see How security scores are calculated.

Topics

- Enabling and disabling security standards
- Viewing details for a standard

- Enabling and disabling controls in specific standards

Enabling and disabling security standards

You can enable or disable each security standard that's available in Security Hub.

Before you enable any security standards, make sure that you have enabled AWS Config and configured resource recording. Otherwise, Security Hub may not be able to generate findings for the controls that apply to a standard. For more information, see Configuring AWS Config.

> **Note**
>
> The instructions for enabling and disabling standards vary based on whether or not you use central configuration. This section describes the differences. Central configuration is available to users who integrate Security Hub and AWS Organizations. We recommend using central configuration to simplify the process of enabling and disabling standards in multi-account, multi-Region environments.

Enabling a security standard

When you enable a security standard, all of the controls that apply to the standard are automatically enabled in it. Security Hub also starts generating findings for controls that apply to the standard.

You can choose which controls to enable and disable in each standard. Disabling a control stops findings for the control from being generated, and the control is ignored when calculating security scores.

When you enable Security Hub, Security Hub calculates the initial security score for a standard within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub console. It can take up to 24 hours for first-time security scores to be generated in the China Regions and AWS GovCloud (US) Region. Scores are only generated for standards that are enabled when you visit those pages. In addition, AWS Config resource recording must be configured for scores to appear. After first-time score generation, Security Hub updates the security score every 24 hours. Security Hub displays a timestamp to indicate when a security score was last updated. To view a list of standards that are currently enabled in your account, invoke the GetEnabledStandards API.

Enabling a standard across multiple accounts and Regions

To enable a security standard across multiple accounts and AWS Regions, you must use central configuration.

When you use central configuration, the delegated administrator can create Security Hub configuration policies that enable one or more 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 only AWS Foundational Security Best Practices (FSBP) in one OU, and you can choose to enable FSBP and Center for Internet Security (CIS) AWS Foundations Benchmark v1.4.0 in another OU. For instructions on creating a configuration policy that enables specified standards, see Creating and associating Security Hub configuration policies

If you use central configuration, Security Hub doesn't automatically enable any standards in new or existing accounts. Instead, when creating a configuration policy, the delegated administrator defines which standards to enable in different accounts. Security Hub offers a recommended configuration policy in which only FSBP is enabled. For more information, see Types of configuration policies.

> **Note**
>
> The delegated administrator can create configuration policies to enable any standard except Service-Managed Standard: AWS Control Tower. You can enable this standard only in the AWS Control Tower service. If you use central configuration, you can enable and disable controls in this standard for a centrally managed account only in AWS Control Tower.

If you want some accounts to configure their own standards rather than the delegated administrator, the delegated administrator can designate those accounts as self-managed. Self-managed accounts must configure standards separately in each Region.

Enabling a standard in a single account and Region

If you don't use central configuration or if you are a self-managed account, you can't use configuration policies to centrally enable standards in multiple accounts and Regions. However, you can use the following steps to enable a standard in a single account and Region.

Security Hub console

To enable a standard in one account and Region

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. Confirm that you are using Security Hub in the Region in which you want to enable the standard.
3. In the Security Hub navigation pane, choose **Security standards**.
4. For the standard you want to enable, choose **Enable**. This also enables all controls within that standard.
5. Repeat in each Region in which you want to enable the standard.

Security Hub API

To enable a standard in one account and Region

1. Invoke the BatchEnableStandards API.
2. Provide the Amazon Resource Name (ARN) of the standard that you want to enable. To obtain the standard ARN, invoke the DescribeStandards API.
3. Repeat in each Region in which you want to enable the standard.

AWS CLI

To enable a standard in one account and Region

1. Run the batch-enable-standards command.
2. Provide the Amazon Resource Name (ARN) of the standard that you want to enable. To obtain the standard ARN, run the describe-standards command.

```
aws securityhub batch-enable-standards --standards-subscription-requests
 '{"StandardsArn": "standard ARN"}'
```

Example

```
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"}'
```

3. Repeat in each Region in which you want to enable the standard.

Automatically enabling default security standards

If you don't use central configuration, Security Hub automatically enables default security standards in new accounts when they join your organization. All controls that are part of the default standards are also automatically enabled. Currently, the default security standards that are automatically enabled are AWS Foundational Security Best Practices (FSBP) and Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0. You can turn off automatically enabled standards if you prefer to manually enable standards in new accounts.

If you use central configuration, you can create a configuration policy that enables the default standards and associate this policy with the root. All of your organization accounts and OUs will inherit this configuration policy unless they are associated with a different policy or are self-managed.

Turn off automatically enabled standards

The following steps apply only if you integrate with AWS Organizations but don't use central configuration. If you don't use the Organizations integration, you can turn off a default standard when you first enable Security Hub, or you can follow the steps for disabling a standard.

Security Hub console

To turn off automatically enabled standards

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the administrator account.
2. In the Security Hub navigation pane, under **Settings**, choose **Configuration**.
3. In the **Accounts** section, turn off **Auto-enable default standards**.

Security Hub API

To turn off automatically enabled standards

1. Invoke the UpdateOrganizationConfiguration API from the Security Hub administrator account.

2. To turn off automatically enabled standards in new member accounts, set `AutoEnableStandards` equal to `NONE`.

AWS CLI

To turn off automatically enabled standards

1. Run the update-organization-configuration command.
2. Include the `auto-enable-standards` parameter to turn off automatically enabled standards in new member accounts.

```
aws securityhub update-organization-configuration --auto-enable-standards
```

Disabling a security standard

When you disable a security standard in Security Hub, the following occurs:

- All of the controls that apply to the standard are also disabled unless they are associated with another standard.
- Checks for the disabled controls are no longer performed, and no additional findings are generated for the disabled controls.
- Existing findings for disabled controls are archived automatically after approximately 3–5 days.
- The AWS Config rules that Security Hub created for the disabled controls are removed.

  This normally occurs within a few minutes after you disable the standard, but might take longer. If the first request to delete the AWS Config rules fails, then Security Hub retries every 12 hours. However, if you disabled Security Hub or you don't have any other standards enabled, then Security Hub can't retry the request, meaning that it can't delete the AWS Config rules. If this occurs, and you need to delete AWS Config rules, contact AWS Support.

Disabling a standard across multiple accounts and Regions

To disable a security standard across multiple accounts and Regions, you must use central configuration.

When you use central configuration, the delegated administrator can create configuration policies that disable one or more standards. You can associate a configuration policy with specific accounts

and 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 Payment Card Industry Data Security Standard (PCI DSS) in one OU, and you can choose to disable both PCI DSS and National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5 in another OU. For instructions on creating a configuration policy that disables specified standards, see Creating and associating Security Hub configuration policies.

> **Note**
>
> The delegated administrator can create configuration policies to disable any standard except the Service-Managed Standard: AWS Control Tower. You can disable this standard only in the AWS Control Tower service. If you use central configuration, you can enable and disable controls in this standard for a centrally managed account only in AWS Control Tower.

If you want some accounts to configure their own standards rather than the delegated administrator, the delegated administrator can designate those accounts as self-managed. Self-managed accounts must configure standards separately in each Region.

Disabling a standard 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 standards in multiple accounts and Regions. However, you can use the following steps to disable a standard in a single account and Region.

Security Hub console

To disable a standard in one account and Region

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. Confirm that you are using Security Hub in the Region in which you want to disable the standard.
3. In the Security Hub navigation pane, choose **Security standards**.
4. For the standard you want to disable, choose **Disable**.
5. Repeat in each Region in which you want to disable the standard.

Security Hub API

To disable a standard in one account and Region

1. Invoke the BatchDisableStandards API.
2. For each standard you want to disable, provide the standard subscription ARN. To get the subscription ARNs for your enabled standards, invoke the GetEnabledStandards API.
3. Repeat in each Region in which you want to disable the standard.

AWS CLI

To disable a standard in one account and Region

1. Run the batch-disable-standards command.
2. For each standard you want to disable, provide the standard subscription ARN. To get the subscription ARNs for your enabled standards, run the get-enabled-standards command.

```
aws securityhub batch-disable-standards --standards-subscription-arns "standard
 subscription ARN"
```

Example

```
aws securityhub batch-disable-standards --standards-subscription-arns
 "arn:aws:securityhub:us-west-1:123456789012:subscription/aws-foundational-
security-best-practices/v/1.0.0"
```

3. Repeat in each Region in which you want to disable the standard.

Viewing details for a standard

On the AWS Security Hub console, the details page for a standard includes the following information:

- The standard security score and a visual summary of security checks for the controls that are enabled in the standard. If you integrate with AWS Organizations, controls that are enabled in at least one organization account are considered enabled.
- The settings to enable or disable a control that applies to the standard.

- A list of controls that apply to the standard. The controls are divided into different tabs based on enablement status. The number of controls in the **All enabled** column is the sum of the controls in the **Failed**, **Unknown**, **No data**, and **Passed** columns.

You can also use the Security Hub API and AWS CLI to retrieve details for a standard. The following sections explain how to get details for a standard.

Displaying the details page for an enabled standard (console)

From the **Security standards** page, you can display the details page for an enabled standard.

If you are signed in to the administrator account, you can view details for any standard that is enabled in at least one member account.

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. In the Security Hub navigation pane, choose **Security standards**.
3. For the standard that you want to display the details for, choose **View results**.

Standard security score and security checks summary

At the top of the standard details page is the security score for the standard. The score is the percentage of passed controls relative to the number of enabled controls (that have data) for the standard.

Security Hub typically calculates the initial security score within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub console. Scores are only generated for standards that are enabled when you visit those pages. To view a list of standards that are currently enabled, use the `GetEnabledStandards` API operation. In addition, AWS Config resource recording must be configured for scores to appear. After first-time score generation, Security Hub updates the security score every 24 hours. Security Hub displays a timestamp to indicate when a security score was last updated. For more information, see the section called "Determining security scores".

> **Note**
>
> It can take up to 24 hours for first-time security scores to be generated in the China Regions and AWS GovCloud (US) Region.

Next to the score is a chart that summarizes security checks for controls that are enabled for the standard. The chart shows the percentage of failed and passed security checks. When you pause on the chart, the pop-up displays the following:

- The number of failed security checks for controls of each severity
- The number of security checks for controls with a status of **Unknown**
- The number of security checks that passed

For administrator accounts, the standard score and chart are aggregated across the administrator account and all member accounts.

All of the data on the **Security standards** details pages is specific to the current Region unless you have set an aggregation Region. If you have set an aggregation Region, the security scores apply across Regions and include findings in all linked Regions. The compliance status of controls on the standards details pages also reflect findings from linked Regions, and the number of security checks includes findings from linked Regions.

Viewing the controls in enabled standards

When you visit the details page for a standard, you can view a list of security controls that apply to the standard. This list is sorted based on the compliance status of the control and the severity assigned to each control. Security Hub updates the control statuses and security check count every 24 hours. A timestamp on each tab indicates when the control statuses and security check count were most recently updated. For more information, see the section called “Compliance status and control status”.

For administrator accounts, the control compliance statuses and number of security checks are aggregated across the administrator account and all member accounts.

The **All enabled** tab lists all of the controls that are currently enabled in the standard. For administrator accounts, the **All enabled** tab includes controls that are enabled in the standard in their account or at least one member account.

On the **Failed**, **Unknown**, **No data**, and **Passed** tabs, the controls from the **All enabled** tab are filtered to include only enabled controls with a specific status.

The **Disabled** tab contains the list of controls that are disabled in the standard. For administrator accounts, the **Disabled** tab includes controls that are disabled in the standard in their account and all member accounts.

For each control, the tabs display the following information:

- The status of the control (see the section called "Compliance status and control status")
- The severity assigned to the control
- The control ID and title
- The number of failed active findings out of the total number of active findings. If applicable, the **Failed checks** column also lists the number of findings with a status of **Unknown**.

In addition to the search filter on each tab, you can sort the lists based on the following fields:

- **Compliance Status**
- **Severity**
- **ID**
- **Title**
- **Failed checks**

You can sort each list using any of the columns. By default, the **All enabled** tab is sorted so that failed controls are at the top of the list. This helps you to immediately focus on issues that require remediation.

On the remaining tabs, the controls are sorted by default in descending order by severity. In other words, critical controls are first, followed by high, then medium, then low severity controls.

Choose your preferred access method, and follow the steps to display the available controls for an enabled standard. In lieu of these instructions, you can also use the DescribeStandardsControl API operation.

Security Hub console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. Choose **Security standards** in the navigation pane.
3. Choose **View results** for a standard. The bottom of the page lists the controls (divided by tabs) that apply to the standard.

Security Hub API

1. Run ListSecurityControlDefinitions and provide a standard Amazon Resource Name (ARN) 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 control IDs. 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 to find out whether a control is enabled in each standard that you've enabled in your account.
3. Identify the control by providing `SecurityControlId` or `SecurityControlArn`. Pagination parameters are optional.

   Example request:

   ```
   {
       SecurityControlId: Config.1
       NextToken: lkeyusdlk-sdlflsnd-ladfterb
       MaxResults: 5
   }
   ```

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 control IDs. 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 to find out whether a control is enabled in each standard that you've enabled in your account.
3. Identify the control by providing `security-control-id` or `security-control-arn`.

   Example command:

```
aws securityhub --region us-east-1 list-standards-control-associations --
security-control-id Config.1
```

Downloading the controls list

You can download the current page of the controls list to a `.csv` file.

If you filtered the controls list, then the downloaded file includes only the controls that match the filter settings.

If you chose a specific control from the list, then the downloaded file includes only that control.

To download the current page of the controls list or the currently selected control, choose **Download**.

Enabling and disabling controls in specific standards

When you enable a standard in AWS Security Hub, 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.

> **Note**
>
> If you use Security Hub central configuration, the delegated administrator can enable and and disable controls for organization accounts across *all* enabled standards. We recommend this approach so that a control's enablement status is aligned across standards. However, the delegated administrator can designate accounts as self-managed, which gives them the ability to enable and disable controls in specific standards. For more information, see Central configuration in Security Hub.

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 and disable controls in a specific standard. You must enable and disable controls separately in each AWS account and AWS Region. When you enable or disable a control, it only impacts the current account and Region.

You can enable and disable controls in each Region by using the Security Hub console, Security Hub API, or AWS CLI. If you have set an aggregation Region, you see controls from all linked Regions. If a control is available in a linked Region but not in the aggregation Region, you cannot enable or disable that control from the aggregation Region. For multi-account and multi-Region control disablement scripts, see Disabling Security Hub controls in a multi-account environment.

Enabling a control in a specific standard

To enable a control in a standard, you must first enable at least one standard to which the control applies. For more information about enabling a standard, see Enabling and disabling security standards. When you enable a control in a standard, AWS Security Hub starts to generate findings for that control. Security Hub 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 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 Enabling a control in all standards in a single account and Region.

Security Hub console

To enable a control in a specific standard

1. Open the AWS Security Hub 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 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"}]'
```

Disabling a control in a specific standard

When you disable a control in a standard, Security Hub stops generating findings for the control. The control status is no longer used in calculating the security score for the standard.

One way to disable a control is by disabling all standards that the control applies to. When you disable a standard, all of the controls that apply to the standard are disabled (however, those controls may still remain enabled in other standards). For information about disabling a standard, see the section called “Enabling and disabling standards”.

When you disable a control by disabling a standard that it applies to, the following occurs:

- Security checks for the control are no longer performed for that standard. This means the control status won't affect the standard security score (Security Hub will continue running security checks for the control if it is enabled in other standards).
- No additional findings are generated for that control.
- Existing findings are archived automatically after 3-5 days (note that this is best effort and not guaranteed).
- The related AWS Config rules that Security Hub created are removed.

When you disable a standard, Security Hub does not track which controls were disabled. If you subsequently enable the standard again, all of the controls that apply to it are automatically enabled. In addition, disabling a control is a one-time action. Suppose you disable a control, and then you enable a standard which was previously disabled. If the standard includes that control, it will be enabled in that standard. When you enable a standard in Security Hub, all of the controls that apply to that standard are automatically enabled.

Instead of disabling a control by disabling a standard that it applies to, you can just disable the control in one or more specific standards.

To reduce finding noise, it can be useful to disable controls that aren't relevant to your environment. For recommendations of which controls to disable, see Security Hub controls that you might want to disable.

Follow these steps to disable a control in *specific* standards. In lieu of the following steps, you can also use the `UpdateStandardsControl` API action to disable controls in a specific standard. For instructions on disabling a control in all standards, see Enabling and disabling controls in all standards.

Security Hub console

To disable a control in a specific standard

1. Open the AWS Security Hub 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 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"}]'
```

Security Hub controls reference

This controls reference provides a list of available AWS Security Hub controls with links to more information about each control. The overview table displays the controls in alphabetical order by control ID. Only controls in active use by Security Hub are included here. Retired controls are excluded from this list. 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 console displays security control IDs, regardless of whether consolidated control findings is turned on or off in your account. However, Security Hub 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 may occasionally update control titles or descriptions, control IDs stay the same.

  Control IDs may skip numbers. These are placeholders for future controls.
- **Applicable standards** – Indicates which standards a control applies to. Select a control to see specific requirements from third-party compliance frameworks.
- **Security control title** – This title applies across standards. The Security Hub console displays security control titles, regardless of whether consolidated control findings is turned on or off in your account. However, Security Hub 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.
- **Severity** – The severity of a control identifies its importance from a security standpoint. For information about how Security Hub determines control severity, see Assigning severity to control findings.
- **Schedule type** – Indicates when the control is evaluated. For more information, see Schedule for running security checks.
- **Supports custom parameters** – Indicates whether the control supports custom values for one or more parameters. Select a control to see the parameter details. For more information, see Custom control parameters.

Select a control to view further details. Controls are listed in alphabetical order of the service name.

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	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	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	HIGH	 No	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	MEDIUM	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
		Control Tower, NIST SP 800-53 Rev. 5			
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	MEDIUM	 No	Change triggered
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
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	MEDIUM	 No	Change triggered
AppSync.2	AWS AppSync should have field-level logging enabled	AWS Foundational Security Best Practices v1.0.0	MEDIUM	 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
AutoScaling.1	Auto scaling groups associated with a Classic Load Balancer should use load balancer health checks	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	LOW	 No	Change triggered
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	HIGH	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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	HIGH	 No	Change triggered
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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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	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	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
CloudFront.5	CloudFront distributions should have logging enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	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	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	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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	MEDIUM	 No	Change triggered
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	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	HIGH	 No	Periodic
CloudFront.13	CloudFront distributions should use origin access control	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
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 v1.2.0, 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	HIGH	 No	Periodic
CloudTrail.2	CloudTrail should have encryption at-rest enabled	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	MEDIUM	 No	Periodic
CloudTrail.3	CloudTrail should be enabled	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 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	LOW	 No	Periodic
CloudTrail.5	CloudTrail trails should be integrated with Amazon CloudWatch Logs	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	LOW	 No	Periodic
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	CRITICAL	 No	Change triggered and periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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	LOW	 No	Periodic
CloudWatch.1	A log metric filter and alarm should exist for usage of the "root" user	CIS AWS Foundations Benchmark v1.2.0, PCI DSS v3.2.1, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.2	Ensure a log metric filter and alarm exist for unauthorized API calls	CIS AWS Foundations Benchmark v1.2.0	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 v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.5	Ensure a log metric filter and alarm exist for CloudTrail configuration changes	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudWatch.6	Ensure a log metric filter and alarm exist for AWS Management Console authentication failures	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.7	Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.8	Ensure a log metric filter and alarm exist for S3 bucket policy changes	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.9	Ensure a log metric filter and alarm exist for AWS Config configuration changes	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.10	Ensure a log metric filter and alarm exist for security group changes	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	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 v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.12	Ensure a log metric filter and alarm exist for changes to network gateways	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.13	Ensure a log metric filter and alarm exist for route table changes	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.14	Ensure a log metric filter and alarm exist for VPC changes	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
CloudWatch.15	CloudWatch alarms should have specified actions configured	NIST SP 800-53 Rev. 5	HIGH	 Yes	Change triggered
CloudWatch.16	CloudWatch log groups should be retained for a specified time period	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
CloudWatch.17	CloudWatch alarm actions should be enabled	NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
CodeBuild.1	CodeBuild Bitbucket source repository URLs should not contain sensitive credentials	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
CodeBuild.2	CodeBuild project environment variables should not contain clear text credentials	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
CodeBuild.3	CodeBuild S3 logs should be encrypted	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
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
Config.1	AWS Config should be enabled	CIS AWS Foundations Benchmark v1.2.0, AWS Foundational Security Best Practices v1.0.0, PCI DSS v3.2.1, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
DMS.1	Database Migration Service replication instances should not be public	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

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 v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
DMS.7	DMS replication tasks for the target database should have logging enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
DMS.8	DMS replication tasks for the source database should have logging enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
DMS.9	DMS endpoints should use SSL	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
DocumentDB.2	Amazon DocumentDB clusters should have an adequate backup retention period	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 Yes	Change triggered
DocumentDB.3	Amazon DocumentDB manual cluster snapshots should not be public	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	CRITICAL	 No	Change triggered
DocumentDB.4	Amazon DocumentDB clusters should publish audit logs to CloudWatch Logs	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	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
DynamoDB.1	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 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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.2	VPC default security groups should not allow inbound or outbound traffic	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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.6	VPC flow logging should be enabled in all VPCs	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	MEDIUM	 No	Periodic
EC2.7	EBS default encryption should be enabled	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)	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
EC2.9	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
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	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, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
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	HIGH	 No	Change triggered

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 v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
EC2.16	Unused Network Access Control Lists should be removed	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	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
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	HIGH	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EC2.19	Security groups should not allow unrestricted access to ports with high risk	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	CRITICAL	 No	Change triggered
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	MEDIUM	 No	Change triggered
EC2.21	Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389	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	Change triggered
EC2.22	Unused EC2 security groups should be removed	Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
EC2.25	EC2 launch templates should not assign public IPs to network interfaces	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
EC2.28	EBS volumes should be in a backup plan	NIST SP 800-53 Rev. 5	LOW	 Yes	Periodic
EC2.51	EC2 Client VPN endpoints should have client connection logging enabled	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
ECR.1	ECR private repositories should have image scanning configured	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ECS.2	ECS services should not have public IP addresses assigned to them automatically	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.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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
ECS.8	Secrets should not be passed as container environment variables	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.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
ECS.10	ECS Fargate services should run on the latest Fargate platform version	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.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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
EFS.1	Elastic File System should be configured to encrypt file data at-rest using AWS KMS	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
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 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
EKS.1	EKS cluster endpoints should not be publicly accessible	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	HIGH	 No	Periodic
EKS.2	EKS clusters should run on a supported Kubernetes version	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
EKS.8	EKS clusters should have audit logging enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
ElastiCache.1	ElastiCache Redis clusters should have automatic backup enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	HIGH	 Yes	Periodic
ElastiCache.2	ElastiCache for Redis cache clusters should have auto minor version upgrades enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	HIGH	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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 have encryption-at-rest enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
ElastiCache.5	ElastiCache replication groups should have encryption-in-transit enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
ElastiCache.6	ElastiCache replication groups of earlier Redis versions should have Redis AUTH enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
ElasticBeanstalk.2	Elastic Beanstalk managed platform updates should be enabled	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	HIGH	 Yes	Change triggered
ElasticBeanstalk.3	Elastic Beanstalk should stream logs to CloudWatch	AWS Foundational Security Best Practices v1.0.0	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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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	MEDIUM	 No	Change triggered
ELB.3	Classic Load Balancer listeners should be configured with HTTPS or TLS termination	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.4	Application Load Balancer should be configured to drop http headers	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.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 v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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 v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	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
ELB.14	Classic Load Balancer should be configured with defensive or strictest desync mitigation mode	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.16	Application Load Balancers should be associated with an AWS WAF web ACL	NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
EMR.1	Amazon EMR cluster primary nodes 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	HIGH	 No	Periodic
EMR.2	Amazon EMR block public access setting should be enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	CRITICAL	 No	Periodic
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 v1.0.0, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	CRITICAL	 No	Periodic
ES.3	Elasticsearch 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
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
ES.5	Elasticsearch domains should have audit logging 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.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, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
EventBridge.3	EventBridge custom event buses should have a resource-based policy attached	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
EventBridge.4	EventBridge global endpoints should have event replication enabled	NIST SP 800-53 Rev. 5	MEDIUM	 No	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	Change triggered
GuardDuty.1	GuardDuty should be 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	HIGH	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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	HIGH	 No	Change triggered
IAM.2	IAM users should not have IAM policies attached	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	LOW	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.3	IAM users' access keys should be rotated every 90 days or less	CIS AWS Foundations Benchmark v1.2.0, 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
IAM.4	IAM root user access key should not exist	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	CRITICAL	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.5	MFA should be enabled for all IAM users that have a console password	CIS AWS Foundations Benchmark v1.2.0, 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
IAM.6	Hardware MFA should be enabled for the root user	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	CRITICAL	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.7	Password policies for IAM users should have strong configurations	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
IAM.8	Unused IAM user credentials should be removed	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	MEDIUM	 No	Periodic
IAM.9	MFA should be enabled for the root user	CIS AWS Foundations Benchmark v1.2.0, PCI DSS v3.2.1, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	CRITICAL	 No	Periodic
IAM.10	Password policies for IAM users should have strong configurations	PCI DSS v3.2.1	MEDIUM	 No	Periodic

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
IAM.11	Ensure IAM password policy requires at least one uppercase letter	CIS AWS Foundations Benchmark v1.2.0	MEDIUM	 No	Periodic
IAM.12	Ensure IAM password policy requires at least one lowercase letter	CIS AWS Foundations Benchmark v1.2.0	MEDIUM	 No	Periodic
IAM.13	Ensure IAM password policy requires at least one symbol	CIS AWS Foundations Benchmark v1.2.0	MEDIUM	 No	Periodic
IAM.14	Ensure IAM password policy requires at least one number	CIS AWS Foundations Benchmark v1.2.0	MEDIUM	 No	Periodic
IAM.15	Ensure IAM password policy requires minimum password length of 14 or greater	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	MEDIUM	 No	Periodic
IAM.16	Ensure IAM password policy prevents password reuse	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	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	LOW	 No	Periodic
IAM.18	Ensure a support role has been created to manage incidents with AWS Support	CIS AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0	LOW	 No	Periodic
IAM.19	MFA should be enabled for all IAM users	PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	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	LOW	 No	Change triggered
IAM.22	IAM user credentials unused for 45 days should be removed	CIS AWS Foundations Benchmark v1.4.0	MEDIUM	 No	Periodic

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
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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
KMS.4	AWS KMS key rotation should be enabled	CIS AWS Foundations Benchmark v1.2.0, PCI DSS v3.2.1, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Periodic
Lambda.1	Lambda function policies should prohibit public access	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
Lambda.2	Lambda functions should use supported runtimes	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
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 v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
MSK.2	MSK clusters should have enhanced monitoring configured	NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
MQ.5	ActiveMQ brokers should use active/standby deployment mode	NIST SP 800-53 Rev. 5	LOW	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
MQ.6	RabbitMQ brokers should use cluster deployment mode	NIST SP 800-53 Rev. 5	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 v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	MEDIUM	 No	Change triggered
Neptune.3	Neptune DB cluster snapshots should not be public	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5, Service-Managed Standard: AWS Control Tower	CRITICAL	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
NetworkFirewall.2	Network Firewall logging should be enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	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	MEDIUM	 No	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
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	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	MEDIUM	 No	Change triggered
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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 v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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.10	OpenSearch domains should have the latest software update installed	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.2	RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible configuration	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.3	RDS DB instances should have encryption at-rest enabled	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	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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.9	RDS DB instances should publish logs to CloudWatch Logs	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.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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.13	RDS automatic minor version upgrades should be 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
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	RDS DB clusters should be configured to copy tags to 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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
RDS.18	RDS instances should be deployed in a 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
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 v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	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 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
RDS.22	An RDS event notifications subscription should be configured for critical database security group 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.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 v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
RDS.25	RDS database instances should use a custom administrator 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
RDS.26	RDS DB instances should be protected by a backup plan	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
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.34	Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
RDS.35	RDS DB clusters should have automatic minor version upgrade enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
Redshift.1	Amazon Redshift clusters should prohibit public access	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
Redshift.2	Connections to Amazon Redshift clusters should be encrypted in transit	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.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
Redshift.4	Amazon Redshift clusters should have audit logging 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.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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
Route53.2	Route 53 public hosted zones should log DNS queries	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
S3.1	S3 general purpose buckets should have block public access settings enabled	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
S3.5	S3 general purpose buckets should require requests to use SSL	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	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	HIGH	 No	Change triggered
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
S3.8	S3 general purpose buckets should block public access	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	HIGH	 No	Change triggered
S3.9	S3 general purpose buckets should have server access logging enabled	AWS Foundational Security Best Practices, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	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
S3.11	S3 general purpose buckets should have event notifications enabled	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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	LOW	 No	Change triggered
S3.15	S3 general purpose buckets should have Object Lock enabled	NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Change triggered
S3.17	S3 general purpose buckets should be encrypted at rest with AWS KMS keys	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
S3.19	S3 access points should have block public access settings enabled	AWS Foundational Security Best Practices, NIST SP 800-53 Rev. 5	CRITICAL	 No	Change triggered
S3.20	S3 general purpose buckets should have MFA delete enabled	CIS AWS Foundations Benchmark v1.4.0, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
SageMaker.1	Amazon SageMaker notebook instances should not have direct internet access	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	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
SecretsManager.1	Secrets Manager secrets should have automatic rotation 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
SecretsManager.2	Secrets Manager secrets configured with automatic rotation should rotate successfully	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 No	Change triggered
SecretsManager.3	Remove unused Secrets Manager secrets	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
SecretsManager.4	Secrets Manager secrets should be rotated within a specified number of days	AWS Foundational Security Best Practices v1.0.0, Service-Managed Standard: AWS Control Tower, NIST SP 800-53 Rev. 5	MEDIUM	 Yes	Periodic
SNS.1	SNS topics should be encrypted at-rest using AWS KMS	NIST SP 800-53 Rev. 5	MEDIUM	 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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	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, Service-Managed Standard: AWS Control Tower, PCI DSS v3.2.1, NIST SP 800-53 Rev. 5	LOW	 No	Change triggered
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
StepFunctions.1	Step Functions state machines should have logging turned on	AWS Foundational Security Best Practices	MEDIUM	 Yes	Change triggered

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
WAF.1	AWS WAF Classic Global Web ACL logging should be enabled	AWS Foundational Security Best Practices v1.0.0, NIST SP 800-53 Rev. 5	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
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

Security control ID	Security control title	Applicable standards	Severity	Supports custom parameters	Schedule type
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
WAF.11	AWS WAF web ACL logging should be enabled	NIST SP 800-53 Rev. 5	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	MEDIUM	 No	Change triggered

Topics

- Amazon GuardDuty controls
- AWS Identity and Access Management controls
- Amazon Kinesis controls
- AWS Key Management Service controls
- AWS Lambda controls
- Amazon Macie controls
- Amazon MSK controls
- Amazon MQ controls
- Amazon Neptune controls
- AWS Network Firewall controls
- Amazon OpenSearch Service controls
- AWS Private Certificate Authority controls
- Amazon Relational Database Service controls
- Amazon Redshift controls
- Amazon Route 53 controls
- Amazon Simple Storage Service controls
- Amazon SageMaker controls
- AWS Secrets Manager controls
- Amazon Simple Notification Service controls
- Amazon Simple Queue Service controls
- AWS Step Functions controls
- AWS WAF controls

AWS account controls

These controls are related to 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 AWS 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 Adding, changing, or removing alternate contacts in the *AWS Billing and Cost Management User 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*.

AWS Certificate Manager controls

These controls are related to ACM resources.

These controls may 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)

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

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*.

Amazon API Gateway controls

These controls are related to API Gateway resources.

These controls may 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)

Category: Protect > Data protection

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)

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*.

AWS AppSync controls

These controls are related to AWS AppSync resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[AppSync.2] AWS AppSync should have field-level logging enabled

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`	`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.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*.

Amazon Athena controls

These controls are related to Athena resources.

These controls may 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

Category: Protect > Data protection > Encryption of data at rest

> ⚠ **Important**
>
> Security Hub retired this control in April 2024. For more information, see Change log for Security Hub controls.

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**.

AWS Backup controls

These controls are related to AWS Backup 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*.

AWS CloudFormation controls

These controls are related to CloudFormation 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 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*.

Amazon CloudFront controls

These controls are related to CloudFront resources.

These controls may 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)

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)

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)

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.

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 additional guidance on how to analyze access logs, see Querying Amazon CloudFront logs in the *Amazon Athena User Guide*.

Remediation

To configure access logging for a CloudFront distribution, see Configuring and using standard logs (access logs) in the *Amazon CloudFront Developer Guide*.

[CloudFront.6] CloudFront distributions should have WAF enabled

Related requirements: NIST.800-53.r5 AC-4(21)

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)

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.

[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)

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)

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)

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 policy configuration

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*.

AWS CloudTrail controls

These controls are related to CloudTrail resources.

These controls may 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, 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: PCI DSS v3.2.1/3.4, CIS AWS Foundations Benchmark v1.2.0/2.7, CIS AWS Foundations Benchmark v1.4.0/3.7, 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)

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] CloudTrail should be enabled

Related requirements: 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

Category: Identify > Logging

Severity: High

Resource type: `AWS::::Account`

AWS Config rule: `cloudtrail-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether CloudTrail is enabled in your AWS account. The control fails if your account doesn't have at least one CloudTrail trail.

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: PCI DSS v3.2.1/10.5.2, PCI DSS v3.2.1/10.5.5, CIS AWS Foundations Benchmark v1.2.0/2.2, CIS AWS Foundations Benchmark v1.4.0/3.2, 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)

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

Category: Identify > Logging

Severity: Critical

Resource type: `AWS::CloudTrail::Trail`

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

Category: Identify > Logging

Severity: Low

Resource type: `AWS::CloudTrail::Trail`

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**.

If the bucket is publicly accessible, the check generates a failed finding.

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*.

Amazon CloudWatch controls

These controls are related to CloudWatch resources.

These controls may 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: PCI DSS v3.2.1/7.2.1, 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

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 reduce 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`

Field	Value
	`EXISTS && $.eventType !="AwsServiceEvent"}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Creating 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

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 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 Creating 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") && ($.userIdentity.type = "IAMUser") && ($.responseElements.ConsoleLogin = "Success") }`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Creating 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

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) || ($.eventName=DeletePolicyVersion) \|\| ($.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 Creating 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 AWS Configuration changes

Related requirements: CIS AWS Foundations Benchmark v1.2.0/3.5, CIS AWS Foundations Benchmark v1.4.0/4.5

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`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Creating 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

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 Creating 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

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`**

Field	Value
Default value	**0**

4. Create an alarm based on the filter. For instructions, see Creating 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

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) || ($.eventName=DeleteBucketLifecycle) \|\| ($.eventName=DeleteBucketReplication))}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Creating 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

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`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Creating 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

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 Creating 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

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) ||`

Field	Value
	`($.eventName=DeleteNetworkAclEntry) \|\| ($.eventName=ReplaceNetworkAclEntry) || ($.eventName=ReplaceNetworkAclAssociation)}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Creating 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.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

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)}`

Field	Value
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Creating 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

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`**

Field	Value
Metric value	**1**
Default value	**0**

4. Create an alarm based on the filter. For instructions, see Creating 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

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) || ($.eventName=ModifyVpcAttribute) \|\| ($.eventName=AcceptVpcPeeringConnection) || ($.eventName=CreateVpcPeeringConnection) \|\| ($.eventName=DeleteVpcPeeringConnection) || ($.eventName=RejectVpcPeeringConnection) \|\| ($.eventName=AttachClassicLinkVpc) || ($.eventName=DetachClassicLinkVpc) \|\| ($.eventN`

Field	Value
	`ame=DisableVpcClassicLink) || ($.eventName=EnableVpcClass icLink)}`
Metric namespace	**`LogMetrics`**
Metric value	**`1`**
Default value	**`0`**

4. Create an alarm based on the filter. For instructions, see Creating 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

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 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)

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 the alarm state changes to `OK`.	Boolean	`true` or `false`	`false`

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*.

AWS CodeBuild controls

These controls are related to CodeBuild 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

Important

On January 10, 2024, the title of this control changed to the preceding title. For more information, see Change log for Security Hub controls.

Related requirements: PCI DSS v3.2.1/8.2.1, NIST.800-53.r5 SA-3

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: PCI DSS v3.2.1/8.2.1, NIST.800-53.r5 IA-5(7), NIST.800-53.r5 SA-3

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)

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 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.

AWS Config controls

These controls are related to AWS Config 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

Related requirements: PCI DSS v3.2.1/10.5.2,PCI DSS v3.2.1/11.5, CIS AWS Foundations Benchmark v1.2.0/2.5, CIS AWS Foundations Benchmark v1.4.0/3.5, 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)

Category: Identify > Inventory

Severity: Medium

Resource type: `AWS::::Account`

AWS Config rule: None (custom Security Hub rule)

Schedule type: Periodic

Parameters: None

This control checks whether AWS Config is enabled in your account in the current Region and is recording all resources. The control fails if AWS Config isn't enabled or isn't recording all 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 between resources.

Security Hub recommends that you enable AWS Config in all Regions. The AWS configuration item history that AWS Config captures enables security analysis, resource change tracking, and compliance auditing.

> **Note**
>
> Config.1 requires that AWS Config is enabled in all Regions in which you use Security Hub. Because Security Hub is a Regional service, the check performed for this control checks only the current Region for the account. It does not check all Regions.

To allow security checks against global resources in each Region, you also must record global resources. 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.
The 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. Since IAM is a global service, IAM resources will only be recorded in the Region in which global resource recording is turned on. For more information, see Security Hub controls that you might want to disable.

Remediation

To enable AWS Config and configure it to record all resources, see Manual setup in the *AWS Config Developer Guide*. To record global resources and ensure no resource types are excluded, select **All resources with customizable overrides**. Remove any **Override settings**, and set the recording frequency to **Continuous recording**.

You can also use an AWS CloudFormation template to automate this process. For more information, see the AWS CloudFormation StackSets sample templates in the *AWS CloudFormation User Guide*.

AWS Database Migration Service controls

These controls are related to AWS DMS 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: 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, 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::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.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)

Category: Detect > 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)

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 AWS 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)

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 AWS 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)

Category: Protect > 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*.

Amazon DocumentDB controls

These controls are related to Amazon DocumentDB resources.

These controls may 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

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)

Category: Protect > Secure network configuration

Severity: Critical

Resource type: `AWS::RDS::DBClusterSnapshot`, `AWS::RDS:DBSnapshot`

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)

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**.

Amazon DynamoDB controls

These controls are related to DynamoDB resources.

These controls may 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	Integer	1 to `40000`	No default value

Parameter	Description	Type	Valid custom values	Security Hub default value
	units for DynamoDB auto scaling			
`targetReadUtilization`	Target utilization percentage for read capacity	Integer	`20` to `90`	No 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 in on-demand capacity mode are only limited by the DynamoDB throughput default table quotas. To raise these quotas, you can file a support ticket with AWS Support.DynamoDB tables in 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::DynamoDB::Cluster`

AWS Config rule: `dax-encryption-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether a DAX cluster is 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`	Boolean	`true` or `false`	No default value

Parameter	Description	Type	Allowed custom values	Security Hub default value
	and the resource uses AWS Backup Vault Lock.			

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.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*.

Amazon Elastic Container Registry controls

These controls are related to Amazon ECR 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

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*.

Amazon ECS controls

These controls are related to Amazon ECS resources.

These controls may 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)

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::ECS::Service`

AWS Configrule: `ecs-service-assign-public-ip-disabled` (custom Security Hub rule)

Schedule type: Change triggered

Parameters:

- `exemptEcsServiceArns` (not customizable). Security Hub does not populate this parameter. Comma-separated list of ARNs of Amazon ECS services that are exempt from this rule.

  This rule is `COMPLIANT` if an Amazon ECS service has `AssignPublicIP` set to `ENABLED` and is specified in this parameter list.

  This rule is `NON_COMPLIANT` if an Amazon ECS service has `AssignPublicIP` set to `ENABLED` and is not specified in this parameter list.

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

To disable automatic public IP assignment, see To configure VPC and security group settings for your service in the *Amazon Elastic Container Service Developer Guide*.

[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 Configrule: 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 Configrule: 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 Configrule: ecs-containers-readonly-access

Schedule type: Change triggered

Parameters: None

This control checks if Amazon ECS containers are limited to read-only access to mounted root filesystems. The control fails if the `readonlyRootFilesystem` parameter is set to `false` or if the parameter doesn't exist in the container definition within the task definition. This control only evaluates the latest active revision of an Amazon ECS task definition.

Enabling this option reduces security attack vectors since the container instance's filesystem cannot be tampered with or written to unless it has explicit read-write permissions on its filesystem folder and directories. This control also adheres to the principle of least privilege.

Remediation

Limiting container definitions to read-only access to root filesystems

1. Open the Amazon ECS classic console at https://console.aws.amazon.com/ecs/.
2. In the left navigation pane, choose **Task definitions**.
3. Select a task definition that has container definitions that need to be updated. For each, complete the following steps:

   - From the drop down, choose **Create new revision with JSON**.

- Add the `readonlyRootFilesystem` parameter, and set it to `true` in the container definition within the task definition.
- Choose **Create**.

[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

Category: Protect > Secure development > Credentials not hard-coded

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Configrule: 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 Configrule: 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)

Category: Identify > Vulnerability, patch, and version management

Severity: Medium

Resource type: `AWS::ECS::Service`

AWS Configrule: `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 Configrule: `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*.

Amazon Elastic Compute Cloud controls

These controls are related to Amazon EC2 resources.

These controls may 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.

To make a public EBS snapshot private, see Share a snapshot in the *Amazon EC2 User Guide for Linux Instances*. 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, 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 for Linux Instances*.

After you assign the new security groups to your resources, remove all inbound and outbound rules from the default security groups. For instructions, see Delete 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 for Linux Instances*.

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 for Linux Instances*.

[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 for Linux Instances*.

[EC2.6] VPC flow logging should be enabled in all VPCs

Related requirements: CIS AWS Foundations Benchmark v1.2.0/2.9, 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, CIS AWS Foundations Benchmark v1.4.0/3.9, 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)

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`.

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, 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). The control fails if the account level encryption is not enabled.

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 for Linux Instances*.

Note that following instance types do not support encryption: R1, C1, and M1.

Remediation

To configure default encryption for Amazon EBS volumes, see Encryption by default in the *Amazon EC2 User Guide for Linux Instances*.

[EC2.8] EC2 instances should use 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

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 for Linux Instances*.

[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 > Public IP addresses

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.

Remediation

Use a non-default VPC so that your instance is not 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 cannot manually associate or disassociate an automatically-assigned public IP address from your EC2 instance. To control whether your EC2 instance receives a public IP address, do one of the following:

- Modify the public IP addressing attribute of your subnet. For more information, see Modifying the public IPv4 addressing attribute for your subnet in the *Amazon VPC User Guide*.
- Enable or disable the public IP addressing feature during launch. This overrides the subnet's public IP addressing attribute. For more information, see Assign a public IPv4 address during instance launch in the *Amazon EC2 User Guide for Linux Instances*.

For more information, see Public IPv4 addresses and external DNS hostnames in the *Amazon EC2 User Guide for Linux Instances*.

If your EC2 instance is associated with an Elastic IP address, then your EC2 instance is reachable from the internet. You can disassociate an Elastic IP address from an instance or network interface at any time. To disassociate an Elastic IP address, see Disassociate an Elastic IP address in the *Amazon EC2 User Guide for Linux Instances*.

[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)

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 for Linux Instances*.

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 for Linux Instances*.

[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).

To release an unused EIP, see Release an Elastic IP address in the *Amazon EC2 User Guide for Linux Instances*.

[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, PCI DSS v3.2.1/1.2.1,PCI DSS v3.2.1/1.3.1,PCI DSS v3.2.1/2.2.2, 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)

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::EC2::SecurityGroup`

AWS Config rule: `restricted-ssh`

Schedule type: Change triggered

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 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 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

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

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)

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 public IPv4 addressing attribute for your subnet in the *Amazon VPC User Guide*. Clear the check box for **Enable auto-assign public IPv4 address**.

[EC2.16] Unused Network Access Control Lists should be removed

Related requirements: NIST.800-53.r5 CM-8(1)

Category: Prevent > 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 (ACLs).

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, then 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: Network security

Severity: Low

Resource type: `AWS::EC2::Instance`

AWS Config rule: `ec2-instance-multiple-eni-check`

Schedule type: Change triggered

Parameters:

- `Adapterids` – A list of network interface IDs that are attached to EC2 instances (not customizable)

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 for Linux Instances*.

[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)

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 (maximum of 32 items)	1 to 65535	`[80,443]`

Parameter	Description	Type	Allowed custom values	Security Hub default value
`authorizedUdpPorts`	List of authorized UDP ports	IntegerList (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.
- If no parameter is used, the control fails for any security group that has an unrestricted inbound traffic rule.

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)

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

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 for Linux Instances*.

[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)

Category: Resilience > Recover > 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, 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)

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 (NACL) allows unrestricted access to the default TCP ports for SSH/RDP ingress traffic. The rule fails if a NACL inbound entry allows a source CIDR block of '0.0.0.0/0' or '::/0' for TCP ports 22 or 3389.

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

For more information about NACLs, see Network ACLs in the VPC User Guide.

[EC2.22] Unused Amazon EC2 security groups should be removed

> **Important**
>
> **RETIRED FROM SPECIFIC STANDARDS** – Security Hub removed this control on September 20, 2023 from the AWS Foundational Security Best Practices standard and the NIST SP 800-53 Rev. 5. This control 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.

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 AWS control checks that security groups are attached to Amazon Elastic Compute Cloud (Amazon EC2) instances or to an elastic network interface. The control will fail if the security group is not associated with an Amazon EC2 instance or an elastic network interface.

Remediation

To create, assign and delete security groups, see Security groups in 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 for Linux Instances.

Remediation

To update an EC2 instance to a new instance type, see Change the instance type in the *Amazon EC2 User Guide for Linux Instances*.

[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)

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*.

[EC2.51] 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)

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*.

Amazon EC2 Auto Scaling controls

These controls are related to Amazon EC2 Auto Scaling 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 Classic Load Balancer should use load balancer 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 your Auto Scaling groups that are associated with a Classic Load Balancer are using Elastic Load Balancing health checks.

This ensures that the group can determine an instance's health based on additional tests provided by the load balancer. Using Elastic Load Balancing health checks can help 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

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 is not included in the launch configuration or if both IMDSv1 and IMDSv2 are enabled.

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 for Linux Instances*.

[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 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 for Linux Instances*.

[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)

Category: Protect > Secure network configuration

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 for Windows Instances*.

Amazon EC2 Systems Manager controls

These controls are related to Amazon EC2 instances that are managed by AWS Systems Manager.

These controls may 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 to maintain security and compliance, Systems Manager scans your stopped and running managed instances. A managed instance is a machine that is 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 to configure and maintain your managed instances.

To learn more, see *AWS Systems Manager User Guide*.

Remediation

To manage EC2 instances with Systems Manager, see Amazon EC2 host management in the *AWS Systems Manager User Guide*. In the **Configuration options** section, you can keep the default choices 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: PCI DSS v3.2.1/6.2, 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)

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: 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(3), NIST.800-53.r5 SI-2(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 SSM 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 the account are public. This control fails if SSM documents with the owner `Self` are public.

SSM documents that are public might allow unintended access to your documents. A public SSM 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 setting for Systems Manager documents that are owned by `Self`.

Remediation

To block public sharing for SSM documents, see Block public sharing for SSM documents in the *AWS Systems Manager User Guide*.

Amazon Elastic File System controls

These controls are related to Amazon EFS resources.

These controls may 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: 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)

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*.

Amazon Elastic Kubernetes Service controls

These controls are related to Amazon EKS resources.

These controls may 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)

Category: Protect > Secure access management > Resource 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 your 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)

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`: `1.25` (not customizable)

This control checks whether an Amazon Elastic Kubernetes Service (Amazon EKS cluster is running on a supported Kubernetes version. The control fails if the EKS cluster is running 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 Amazon EKS version support and FAQ in the Amazon EKS User Guide.

Remediation

To update an EKS cluster, Updating an Amazon EKS cluster Kubernetes version 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)

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::EKS::Cluster`

AWS Config rule: `eks-cluster-logging-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon EKS cluster has audit logging enabled. The control fails if audit logging isn't enabled for the cluster.

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.

Amazon ElastiCache controls

These controls are related to ElastiCache resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[ElastiCache.1] ElastiCache Redis clusters should have automatic backup 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 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 if an Amazon ElastiCache for Redis 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 for Redis 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 for Redis cluster, see Scheduling automatic backups in the *Amazon ElastiCache User Guide*.

[ElastiCache.2] ElastiCache for Redis cache clusters should have auto 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)

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 ElastiCache for Redis automatically applies minor version upgrades to cache clusters. This control fails if ElastiCache for Redis cache clusters do not have minor version upgrades automatically applied.

`AutoMinorVersionUpgrade` is a feature that you can turn on in ElastiCache for Redis to have your cache clusters automatically upgraded 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 apply automatic minor version upgrades to an existing ElastiCache for Redis cache cluster, see Upgrading engine versions in the *Amazon ElastiCache User Guide*.

[ElastiCache.3] ElastiCache for Redis 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 if ElastiCache for Redis replication groups have automatic failover enabled. This control fails if automatic failover isn't enabled for a Redis 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 for Redis 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 for Redis 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 if ElastiCache for Redis replication groups are encrypted at rest. This control fails if an ElastiCache for Redis 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 for Redis replication groups should be encrypted at rest for an added layer of security.

Remediation

To configure at-rest encryption on an ElastiCache for Redis replication group, see Enabling at-rest encryption in the *Amazon ElastiCache User Guide*.

[ElastiCache.5] ElastiCache for Redis 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)

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 if ElastiCache for Redis replication groups are encrypted in transit. This control fails if an ElastiCache for Redis 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 for Redis 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 for Redis replication group, see Enabling in-transit encryption in the *Amazon ElastiCache User Guide*.

[ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH

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

Severity: Medium

Resource type: `AWS::ElastiCache::ReplicationGroup`

AWS Config rule: `elasticache-repl-grp-redis-auth-enabled`

Schedule type: Periodic

Parameters: None

This control checks if ElastiCache for Redis replication groups have Redis AUTH enabled. The control fails for an ElastiCache for Redis replication group if the Redis version of its 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 for Redis replication group, see Modifying the AUTH token on an existing ElastiCache for Redis 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 if ElastiCache clusters are configured with a custom subnet group. The control fails for an ElastiCache cluster if `CacheSubnetGroupName` 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*.

AWS Elastic Beanstalk controls

These controls are related to Elastic Beanstalk 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)

Category: Detect > 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

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*.

Elastic Load Balancing controls

These controls are related to Elastic Load Balancing resources.

These controls may 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)

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)

Category: Protect > 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)

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 http headers

Related requirements: NIST.800-53.r5 SC-7(4), NIST.800-53.r5 SC-8(2)

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 AWS Application Load Balancers to ensure they are 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 that you can disable this control if ELB.12 is enabled.

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: 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 Configrule: `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)

Category: Protect > 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

Category: Data protect > 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

Category: Data Protect > 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*.

Amazon EMR controls

These controls are related to Amazon EMR resources.

These controls may 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, 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: 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*.

Elasticsearch controls

These controls are related to Elasticsearch 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)

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)

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)

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)

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`.

Amazon EventBridge controls

These controls are related to EventBridge resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[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)

Category: Protect > Secure access management > Resource policy configuration

Severity: Low

Resource type: `AWS::Events::EventBus`

AWS Config rule: `custom-schema-registry-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 Managing event bus permissions 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**.

Amazon FSx controls

These controls are related to Amazon FSx resources.

These controls may 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: Change triggered

Parameters: None

This control checks if 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

To configure an FSx for OpenZFS file system to copy tags to backups and volumes, see Updating a file system in the *Amazon FSx OpenZFS User Guide*.

Amazon GuardDuty controls

These controls are related to GuardDuty resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[GuardDuty.1] GuardDuty should be enabled

Related requirements: PCI DSS v3.2.1/11.4, 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)

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 remediate this issue, you enable GuardDuty.

For details on how to enable GuardDuty, including how to use AWS Organizations to manage multiple accounts, see Getting started with GuardDuty in the *Amazon GuardDuty User Guide*.

AWS Identity and Access Management controls

These controls are related to IAM resources.

These controls may 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: PCI DSS v3.2.1/7.2.1, 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)

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: PCI DSS v3.2.1/7.2.1, 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)

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**
>
> IAM users created by Amazon Simple Email Service are automatically created using inline policies. Security Hub automatically exempts these users from this control.
> 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 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 v1.2.0/1.4, CIS AWS Foundations Benchmark v1.4.0/1.14, NIST.800-53.r5 AC-2(1), NIST.800-53.r5 AC-2(3), NIST.800-53.r5 AC-3(15)

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: PCI DSS v3.2.1/2.1, PCI DSS v3.2.1/2.2, PCI DSS v3.2.1/7.2.1, CIS AWS Foundations Benchmark v1.2.0/1.12, CIS AWS Foundations Benchmark v1.4.0/1.4, 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 v1.2.0/1.2, CIS AWS Foundations Benchmark v1.4.0/1.10, 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: 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: PCI DSS v3.2.1/8.3.1, CIS AWS Foundations Benchmark v1.2.0/1.14, CIS AWS Foundations Benchmark v1.4.0/1.6, 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-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 MFA isn't enabled or if any 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 only a virtual MFA device while you wait for hardware purchase approval or for your hardware to arrive. To learn more, see Enabling a virtual multi-factor authentication (MFA) device (console) in the *IAM User Guide*.

Both time-based one-time password (TOTP) and Universal 2nd Factor (U2F) tokens are viable as hardware MFA options.

Remediation

To add a hardware MFA device for the root user, see Enable a hardware MFA device for the AWS account root user (console) 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.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)

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`

Parameter	Description	Type	Allowed custom values	Security Hub default value
`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`
`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: PCI DSS v3.2.1/8.1.4, 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

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, CIS AWS Foundations Benchmark v1.2.0/1.13, CIS AWS Foundations Benchmark v1.4.0/1.5, 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

The root user has complete access to all the services and resources in an AWS account. MFA adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to the AWS Management Console, they're prompted for their user name and password and for an authentication code from their AWS MFA device.

When you use virtual MFA for the root user, CIS recommends that the device used is *not* a personal device. Instead, use a dedicated mobile device (tablet or phone) that you manage to keep charged and secured independent of any individual personal devices. This lessens the risks of losing access to the MFA due to device loss, device trade-in, or if the individual owning the device is no longer employed at the company.

Remediation

To enable MFA for the root user, see Activate MFA on the AWS account root user in the *AWS Account Management Reference Guide*.

[IAM.10] Password policies for IAM users should have strong AWS Configurations

Related requirements: 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)

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

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

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

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

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 v1.2.0/1.9, CIS AWS Foundations Benchmark v1.4.0/1.8

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 v1.2.0/1.10, CIS AWS Foundations Benchmark v1.4.0/1.9

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

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 v1.2.0/1.20, CIS AWS Foundations Benchmark v1.4.0/1.17

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 AWS 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 AWS Support incidents.

To create the role to use for AWS 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: 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)

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 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)

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 v1.4.0/1.12

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*.

Amazon Kinesis controls

These controls are related to Kinesis 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*.

AWS Key Management Service controls

These controls are related to AWS KMS resources.

These controls may 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: PCI DSS v3.2.1/3.6.4, CIS AWS Foundations Benchmark v1.2.0/2.8, CIS AWS Foundations Benchmark v1.4.0/3.8, NIST.800-53.r5 SC-12, NIST.800-53.r5 SC-12(2), NIST.800-53.r5 SC-28(3)

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*.

AWS Lambda controls

These controls are related to Lambda resources.

These controls may 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: 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, 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::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.

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)

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, dotnet6, java21, java17, java11, java8.al2, nodejs20.x, nodejs18.x, nodejs16.x, python3.12, python3.11, python3.10, python3.9, python3.8, ruby3.3, ruby3.2` (not customizable)

This control checks whether the Lambda function settings for runtimes match the expected values set for the supported runtimes for each language. The control fails if a Lambda function doesn't use a supported runtime.

The control checks function settings for the runtimes noted previously under parameters. 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)

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.

Amazon Macie controls

These controls are related to Macie resources.

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*.

Amazon MSK controls

These controls are related to Amazon Managed Streaming for Apache Kafka (Amazon MSK) resources.

These controls may 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)

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

To update encryption settings for MSK clusters, 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 Updating security settings of a cluster in the *Amazon Managed Streaming for Apache Kafka Developer Guide*.

Amazon MQ controls

These controls are related to Amazon MQ resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[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.

Amazon Neptune controls

These controls are related to Neptune 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)

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)

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*.

AWS Network Firewall controls

These controls are related to Network Firewall resources.

These controls may 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)

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

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

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)

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.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 > High availability

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`.

Amazon OpenSearch Service controls

These controls are related to OpenSearch Service resources.

These controls may 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)

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 rule is `NON_COMPLIANT` if the CloudWatch Logs log group of the OpenSearch domain is not specified in this parameter list.

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`.

An OpenSearch domain requires at least three data nodes for high availability and fault-tolerance. 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-1-0-2019-07` (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 set the `TLSSecurityPolicy`. For more information, see Node-to-node encryption 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)

Category: Detect > 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*.

AWS Private Certificate Authority controls

These controls are related to AWS Private CA 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*.

Amazon Relational Database Service controls

These controls are related to Amazon RDS resources.

These controls may 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 AWS Configuration

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-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: 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 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.

RDS DB instances should be configured for multiple Availability Zones (AZs). This ensures the availability of the data stored. 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 monitorin	Enum	1, 5, 10, 15, 30, 60	No default value	

Parameter	Description	Type	Allowed custom values	Security Hub default value	
	g metric collection intervals				

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)

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, Upgrade).

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

To publish 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: 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: Detect > Vulnerability and patch 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.

Enabling automatic minor version upgrades ensures that the latest minor version updates to the relational database management system (RDBMS) are installed. These upgrades might include security patches and bug fixes. Keeping up to date with patch installation is an important step in securing systems.

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] RDS 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

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 RDS DB clusters 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 clusters 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 clusters. Enabling this setting ensures that snapshots inherit the tags of their parent database clusters.

Remediation

To automatically copy tags to snapshots for an RDS DB cluster, see Modifying the DB cluster by using the console, CLI, and API in the *Amazon Aurora User Guide*. Select **Copy tags to snapshots**.

[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

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::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

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

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.

```
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

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.

```
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.

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

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

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 does not apply to engines of the type neptune (Neptune DB) or docdb (DocumentDB). The control fails if the administrative username is set to the default value.

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.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)

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.

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)

Category: Detect > 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 RDS database 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*.

Amazon Redshift controls

These controls are related to Amazon Redshift resources.

These controls may 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: 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 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)

Category: Protect > Data protection > Encryption of data in transit

Severity: Medium

Resource type: `AWS::Redshift::Cluster`

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)

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: Detect > Vulnerability and patch 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 to set the `--allow-version-upgrade` attribute.

```
aws redshift modify-cluster --cluster-identifier clustername --allow-version-upgrade
```

Where *`clustername`* is the name of your Amazon Redshift cluster.

[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 it is created. To create a new cluster, follow the instructions here.

[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*.

Amazon Route 53 controls

These controls are related to Route 53 resources.

These controls may not be available in all AWS Regions. For more information, see Availability of controls by Region.

[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)

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*.

Amazon Simple Storage Service controls

These controls are related to Amazon S3 resources.

These controls may 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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

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, 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)

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

> ⚠ **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

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.

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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

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.

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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

Related requirements: PCI DSS v3.2.1/4.1, 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)

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 `awsexamplebucket` with the name of the bucket you're modifying.

```
{
    "Id": "ExamplePolicy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSSLRequestsOnly",
            "Action": "s3:*",
            "Effect": "Deny",
            "Resource": [
                "arn:aws:s3:::awsexamplebucket",
                "arn:aws:s3:::awsexamplebucket/*"
            ],
            "Condition": {
                "Bool": {
                     "aws:SecureTransport": "false"
                }
```

```
            },
            "Principal": "*"
        }
    ]
}
```

For more information, see the Knowledge Center article What S3 bucket policy should I use to comply with the AWS Config rule s3-bucket-ssl-requests-only?.

[S3.6] S3 general purpose bucket policies should restrict access to other AWS accounts

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

Related requirements: NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

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-replication-enabled`

Schedule type: Change triggered

Parameters:

- `ReplicationType: CROSS-REGION` (not customizable)

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.

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 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)

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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

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::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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. Security Hub retired this control in April 2024 from the AWS Foundational Security Best Practices standard, but it is still included in the NIST SP 800-53 Rev. 5 standard. For more information, see Change log for Security Hub controls.

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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. Security Hub retired this control in April 2024 from the AWS Foundational Security Best Practices standard, but it is still included in the NIST SP 800-53 Rev. 5 standard:. For more information, see Change log for Security Hub controls.

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)

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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub 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-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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

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
`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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

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)

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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

Category: Protect > Data protection > Data deletion protection

Related requirements: NIST.800-53.r5 CP-6(2)

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

> **Important**
>
> On March 12, 2024, the title of this control changed to the title shown. For more information, see Change log for Security Hub controls.

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

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)

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 v1.4.0, 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 on an Amazon S3 general purpose bucket. The control fails if MFA delete is not enabled on the bucket.

When working with S3 Versioning in Amazon S3 buckets, you can optionally add another layer of security by configuring a bucket to enable MFA delete. When you do this, the bucket owner must include two forms of authentication in any request to delete a version or change the versioning state of the bucket. MFA delete provides added security if your 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 and adding an extra layer of friction and security to the delete action.

> **Note**
>
> The MFA delete feature requires bucket versioning as a dependency. Bucket versioning is a method of keeping 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 S3 buckets.

Remediation

To enable S3 Versioning and configure MFA delete on a bucket, see Configuring MFA delete in the *Amazon Simple Storage Service User Guide*.

Amazon SageMaker controls

These controls are related to SageMaker resources.

These controls may 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: 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

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 notebook instance. The control fails if the `DirectInternetAccess` field is enabled for the notebook instance.

If you configure your SageMaker 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 Developer Guide*. You should also ensure that access to your SageMaker configuration is limited to only authorized users. Restrict IAM permissions that permit users to change SageMaker 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 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 notebook instance is launched within a custom virtual private cloud (VPC). This control fails if a SageMaker notebook instance is not launched within a custom VPC or if it is launched in the SageMaker 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 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 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 notebook instance. The control fails if root access is turned on for a SageMaker 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 notebook instances, see Control root access to a SageMaker notebook instance in the *Amazon SageMaker Developer Guide*.

AWS Secrets Manager controls

These controls are related to Secrets Manager 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)

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)

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)

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.

Amazon Simple Notification Service controls

These controls are related to Amazon SNS resources.

These controls may 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

> **Important**
>
> Security Hub retired this control in April 2024 from the AWS Foundational Security Best Practices standard, but it is still included in the NIST SP 800-53 Rev. 5 standard. For more information, see Change log for Security Hub controls.

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::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 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*.

Amazon Simple Queue Service controls

These controls are related to Amazon SQS resources.

These controls may 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*.

AWS Step Functions controls

These controls are related to Step Functions 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

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*.

AWS WAF controls

These controls are related to AWS WAF resources.

These controls may 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)

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)

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.

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)

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.

Viewing and managing security controls

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, a control is related to a specific AWS resource.

Consolidated controls view

The **Controls** page of the Security Hub 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 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 console:

- An overall security score based on the proportion of passed controls compared to the total number of enabled controls with data
- The percentage of failed security checks across all enabled controls
- The number of passed and failed security checks for controls of varying severity
- A list of controls divided into different tabs based on enablement status. Available controls that don't apply to any of your enabled standards appear in the **Disabled** column. Unprocessed controls, such as those that are unavailable in your current Region, appear in the **No data** column. The number of controls in the **All** column is equal to the sum of the controls in the **Failed**, **Unknown**, **Passed**, **Disabled**, and **No data** columns.

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 and disabling controls in all standards.

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 appears in the **Failed** tab of the **Controls** page. 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 appears in the **Failed** tab of the **Controls** page.

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.

Overall security score for controls

The **Controls** page displays an overall security score from 0–100 percent. The overall security score is calculated based on the proportion of passed controls compared to the total number of enabled controls with data.

> **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. This permission isn't required to view security scores for specific standards.

When you enable Security Hub, Security Hub calculates the initial security score within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub console. It can take up to 24 hours for first-time security scores to be generated in the China Regions and AWS GovCloud (US) Region. Scores are only generated for standards that are enabled when you visit those pages. To view a list of standards that are currently enabled, use the `GetEnabledStandards` API operation. In addition, AWS Config resource recording must be configured for scores to appear. The overall security score is the average of the standard security scores.

After first-time score generation, Security Hub updates security scores every 24 hours. Security Hub 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.

Topics

Control categories

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, 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?

Enabling and disabling controls in all standards

AWS Security Hub generates findings for enabled controls, and considers all enabled controls when calculating security scores. You can choose to enable and disable controls across *all* security standards or 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. This section explains how to enable and disable controls across standards. To enable or

disable a control in one or more *specific* standards, see Enabling and disabling controls in specific standards.

If you have set an aggregation Region, the Security Hub console displays controls from all linked Regions. 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.

> **Note**
>
> The instructions for enabling and disabling controls vary based on whether or not you use central configuration. This section describes the differences. Central configuration is available to users who integrate Security Hub and AWS Organizations. We recommend using central configuration to simplify the process of enabling and disabling controls in multi-account, multi-Region environments.

Enabling controls

When you enable a control in a standard, Security Hub starts to run security checks for the control and generate control findings.

Security Hub includes the control status in the calculation of the overall security score and standard security scores. If you turn on consolidated control findings, you receive a single finding for a security check even if you've enabled a control in multiple standards. For more information, see Consolidated control findings.

Enabling a control in all standards across multiple accounts and Regions

To enable a security control across multiple accounts and AWS Regions, you must use central configuration.

When you use central configuration, the delegated administrator can create Security Hub 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 Security Hub 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.

Enabling a control in all standards 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 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 console

To enable a control across standards in one account and Region

1. Open the AWS Security Hub 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 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.

Automatically enabling new controls in enabled standards

Security Hub regularly releases new security controls and adds them to one or more standards. You can choose whether to automatically enable new controls in your enabled standards.

> **Note**
>
> We recommend using central configuration to automatically enable new controls. If your configuration policy includes a list of controls to disable (programmatically, this reflects the `DisabledSecurityControlIdentifiers` parameter), Security Hub automatically enables all other controls across standards, including newly released controls. If your policy includes a list of controls to enable (this reflects the `EnabledSecurityControlIdentifiers` parameter), Security Hub automatically disables all other controls across standards, including newly released ones. For more information, see How Security Hub configuration policies work.

Choose your preferred access method, and follow the steps to automatically enable new controls in enabled standards. The following instructions apply only if you don't use central configuration.

Security Hub console

To automatically enable new controls

1. Open the AWS Security Hub 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 API

To automatically enable new controls

1. Invoke the UpdateSecurityHubConfiguration API.
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
```

Disabling controls

When you disable a control in all standards, the following occurs:

- Security checks for the control are no longer performed.
- No additional findings are generated for that control.
- Existing findings are archived automatically after 3-5 days (note that this is best effort).
- Any related AWS Config rules that Security Hub created are removed.

Instead of disabling a control in *all* standards, you can just disable it in one or more *specific* standards. If you do this, Security Hub doesn't run security checks for the control for the standards

you disabled it in, so it doesn't affect the security score for those standards. However, Security Hub retains the AWS Config rule and continues running security checks for the control if it is enabled in other standards. This can affect your summary security score. For instructions on configuring controls in specific standards, see Enabling and disabling controls in specific standards.

To reduce finding noise, it can be useful to disable controls that aren't relevant to your environment. For recommendations of which controls to disable, see Security Hub controls that you might want to disable.

When you disable a standard, all of the controls that apply to the standard are disabled (however, those controls might remain enabled in other standards). For information about disabling a standard, see the section called “Enabling and disabling standards”.

When you disable a standard, Security Hub doesn't track which controls were disabled. If you subsequently enable the standard again, all of the controls that apply to it are automatically enabled. In addition, disabling a control is a one-time action. Suppose you disable a control, and then you enable a standard that was previously disabled. If the standard includes that control, it will be enabled in that standard. When you enable a standard in Security Hub, all of the controls that apply to that standard are automatically enabled.

Disabling a control in all standards across multiple accounts and Regions

To disable a security control across multiple accounts and AWS Regions, you must use central configuration.

When you use central configuration, the delegated administrator can create Security Hub 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 Security Hub 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.

Disabling a control in all standards 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 use the following steps to disable a control in a single account and Region.

Security Hub console

To disable a control across standards in one account and Region

1. Open the AWS Security Hub 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 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.

Enabling new controls in enabled standards automatically

AWS Security Hub 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.

> **Note**
>
> If you use central configuration and include a list of specific controls to disable in your configuration policy (programmatically, this reflects the `DisabledSecurityControlIdentifiers` parameter, Security Hub automatically enables all other controls across standards, including newly released controls. For more information, see How Security Hub configuration policies work.

We recommend using Security Hub 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 Central configuration in Security Hub.

Security Hub 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.

Security Hub console

To automatically enable new controls

1. Open the AWS Security Hub 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 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 the section called “Enabling and disabling controls in all standards”.

Custom control parameters

Some Security Hub controls use parameters that affect how the control is evaluated. Typically, such controls are evaluated against the default parameter values that Security Hub defines. However, for a subset of these controls, you can customize the parameter values. When you customize a parameter value for a control, Security Hub starts evaluating the control against the value that you specify. If the resource underlying the control satisfies the custom value, Security Hub generates a `PASSED` finding. If the resource doesn't satisfy the custom value, Security Hub generates a `FAILED` finding.

By customizing control parameters, you can refine the security best practices recommended and monitored by Security Hub 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 custom control parameters:

- **[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 explains how to customize and manage control parameters.

How custom control parameters work

A control can have one or more customizable parameters. Possible data types for individual control parameters include the following:

- Boolean
- Double
- Enum
- EnumList
- Integer
- IntegerList
- String
- StringList

For some controls, acceptable parameter values must also fall into a specified range to be valid. In these cases, Security Hub provides the acceptable range.

Security Hub 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 value, even if the custom value of the parameter matches the current, default value defined by Security Hub. 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 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 value for a parameter, set the `ValueType` field to `DEFAULT` instead of `CUSTOM`. For more information, see Reverting to default parameter values in a single account and Region.

When you change a parameter value, you also trigger a new security check that evaluates the control based on the new value. Security Hub then generates new control findings based on the new value. During periodic updates to control findings, Security Hub 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 doesn't conduct any security checks using the new values. You have to enable at least one relevant standard for Security Hub to evaluate the control based on the new parameter value.

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 controls.

Customizing control parameters

The instructions for customizing control parameters vary based on whether you use central configuration. Central configuration is a feature that the delegated Security Hub administrator can use to manage Security Hub capabilities across AWS Regions, accounts, and organizational units (OUs) in their organization.

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.

Customizing control parameters across multiple accounts and Regions

When you use central configuration, you can customize control parameters for centrally managed accounts and OUs across multiple accounts and Regions. 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.

If you're the delegated Security Hub 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 console

To customize control parameters in multiple accounts and Regions

1. Open the AWS Security Hub 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 parameters

   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 add or edit custom parameters 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.

Security Hub API

To customize control parameters in multiple accounts and Regions

To create a new configuration policy with custom parameters

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 add or edit custom parameters 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.

Example API request to create a new 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
                                }
                            }
                        }
```

```
                    }
                ]
            }
        }
    }
}
```

AWS CLI

To customize control parameters in multiple accounts and Regions

To create a new configuration policy with custom parameters

1. Run the create-configuration-policy command 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 running the get-security-control-definition command.

To add or edit parameters in an existing configuration policy

1. To add or update custom input parameters in an existing configuration policy, run the update-configuration-policy command 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 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 running the get-security-control-definition command.

Example command to create a new configuration policy:

```
$ 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

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 console

To customize control parameters in one account and Region

1. Open the AWS Security Hub 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 API

To customize control parameters in one account and Region

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.

Example API request:

```
{
    "SecurityControlId": "ACM.1",
    "Parameters": {
        "daysToExpiration": {
            "ValueType": "CUSTOM",
            "Value": {
                "Integer": 15
            }
        }
    },
    "LastUpdateReason": "Internal compliance requirement"
}
```

AWS CLI

To customize control parameters in one account and Region

1. Run the update-security-control command.
2. For `security-control-id`, 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 running the get-security-control-definition command.
4. Optionally, for `last-update-reason`, provide a reason for customizing the control parameters.

Example command:

```
$ 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"
```

Checking the status of control parameters

It's important to validate and check the status of changes to control parameters. This helps ensure that a control works as you expect and provides the intended security value. To verify that a parameter update was successful, you can review the details of the control on the Security Hub console. On the console, choose the 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 your findings might not yet include the updated parameter values. When the

value of `UpdateState` changes to `READY`, your findings begin to 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 specified a value that doesn't use the correct data type. Submit your request again with valid input. If a parameter update is unsuccessful, Security Hub retains the current value for the parameter.

If an internal failure occurs when you try to update a parameter value, Security Hub automatically retries if you have AWS Config enabled. For more information, see Configuring AWS Config.

Reviewing control parameters

You can review the current values for individual control parameters in your account. If you use central configuration, the delegated Security Hub 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 console

To review current parameter values

1. Open the AWS Security Hub 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 API

To review current parameter values

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.

Example API request:

```
{
```

```
    "SecurityControlIds": ["APIGateway.1", "CloudWatch.15", "IAM.7"]
}
```

AWS CLI

To review current parameter values

Run the batch-get-security-controls command, 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.

Example command:

```
$ 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 console

To review current parameter values in a configuration policy

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the delegated Security Hub 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 API

To review current parameter values in a configuration policy

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.

```
{
    "Identifier": "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

AWS CLI

To review current parameter values in a configuration policy

1. Run the get-configuration-policy command 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.

```
$ aws securityhub get-configuration-policy \
--region us-east-1 \
--identifier "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Your control findings also show current parameter values. In the AWS Security Finding Format (ASFF) syntax, these values appear in the `Parameters` field of the `Compliance` object. To review findings on the Security Hub console, choose **Findings** in the navigation pane. To review findings programmatically, use the GetFindings operation.

> **Note**
>
> After release of the custom control parameters feature, Security Hub will update existing control findings to include the `Parameters` ASFF field. This may take up to 24 hours.

Reverting to default control parameter values

A control parameter can have a default value that Security Hub defines. We might update 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. How you do this depends on whether you use central configuration.

> **Note**
>
> Not all control parameters have a default Security Hub value. In such cases, when `ValueType` is set to `DEFAULT`, there isn't a specific default value that Security Hub uses. Rather, Security Hub ignores the parameter in the absence of a custom value.

Reverting to default parameter values across multiple accounts and Regions

If you use central configuration, you can revert control parameters for centrally managed accounts and OUs across multiple accounts and 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 console

To revert to default parameter values in multiple accounts and Regions

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.

   Sign in using the credentials of the Security Hub delegated 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 API

To revert to default parameter values in multiple accounts and Regions

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 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 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.

Example API request:

```
{
    "Identifier": "arn:aws:securityhub:us-east-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "TestConfigurationPolicy",
    "Description": "Updated configuration policy",
    "UpdatedReason": "Revert ACM.1 parameter to default value",
    "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": {
                "DisbledSecurityControlIdentifiers": [
                    "CloudTrail.2"
                ],
                "SecurityControlCustomParameters": [
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "DEFAULT"
                             }
                         }
                    }
                ]
             }
        }
    }
}
```

AWS CLI

To revert to default parameter values in multiple accounts and Regions

1. Run the update-configuration-policy command 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 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 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.

Example command:

```
$ 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 parameter values 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, your custom control parameters are reset. If you enable Security Hub again in the future, all controls will use default parameter values to start.

Security Hub console

To revert to default parameter values in one account and Region

1. Open the AWS Security Hub 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 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 API

To revert to default parameter values in one account and Region

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 ignores it.
4. Optionally, for `LastUpdateReason`, provide a reason for reverting to default parameter values.

Example API request:

```
{
    "SecurityControlId": "ACM.1",
    "Parameters": {
        "daysToExpiration": {
            "ValueType": "DEFAULT"
        },
    "LastUpdateReason": "New internal requirement"
}
```

AWS CLI

To revert to default parameter values in one account and Region

1. Run the update-security-control command.
2. For `security-control-id`, 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 ignores it.
4. Optionally, for `last-update-reason`, provide a reason for reverting to default parameter values.

Example command:

```
$ aws securityhub update-security-control \
--region us-east-1 \
--security-control-id ACM.1 \
--parameters '{"daysToExpiration": {"ValueType": "DEFAULT"}}' \
--last-update-reason "New internal requirement"
```

Controls that support custom parameters

For a list of security controls that support custom parameters, you can refer to the **Controls** page on the Security Hub console or the Security Hub controls reference. To retrieve this list programmatically, you can use the ListSecurityControlDefinitions operation. In the response, the `CustomizableProperties` object indicates which controls support customizable parameters.

Security Hub controls that you might want to disable

We recommend disabling some AWS Security Hub controls to reduce finding noise and limit costs.

Controls that deal with 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 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, you should also disable controls that involve global resources in all Regions except the Region that records global resources.

> **Note**
>
> If you use central configuration, Security Hub automatically disables controls that involve global resources in all Regions except the home Region. Other controls 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. For more information about central configuration, see Central configuration in Security Hub.

For controls with a *periodic* schedule type, disabling them in Security Hub is required to prevent billing. Setting the AWS Config parameter `includeGlobalResourceTypes` to `false` doesn't affect periodic Security Hub controls.

If you disable recording of global resources in one or more Regions, the control **[Config.1] AWS Config should be enabled** generates a failed finding in those Regions. This is because Config.1 requires recording of global resources in order to pass. You can suppress findings for this control manually or through an automation rule.

The following is a list of Security Hub controls that involve 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
- [EventBridge.4] EventBridge global endpoints should have event replication 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 AWS 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
- [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
- [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

Controls that deal with CloudTrail logging

This control deals with using AWS Key Management Service (AWS KMS) to encrypt AWS CloudTrail trail logs. If you log these trails in a centralized logging account, you only need to enable this control in the account and Region where centralized logging takes place.

> **Note**
>
> 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, suppress findings from the following controls to reduce finding noise.

- [CloudTrail.2] CloudTrail should have encryption at-rest enabled

Controls that deal with CloudWatch alarms

If you prefer to use Amazon GuardDuty for anomaly detection instead of Amazon CloudWatch alarms, you can disable these 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 AWS 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

Viewing details for a control

For each AWS Security Hub control, you can display a page of useful details.

The top of the control details page provides an overview of the control, including:

- **Enablement status** – The top of the page tells you whether the control is enabled for at least one standard in at least one member account. If you have set an aggregation Region, the control is enabled if it is enabled for at least one standard in at least one Region. If the control is disabled, you can enable it from this page. If the control is enabled, you can disable it from this page. For more information, see the section called “Enabling and disabling controls in all standards”.
- **Control status** – This status summarizes the performance of a control based on the compliance status of the control findings. Security Hub 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 console. Statuses are only available for controls that are enabled when you visit those pages. Use the `UpdateStandardsControl` API operation to enable or disable a control. In addition, AWS Config resource recording must be configured for the control status to appear. After control statuses are generated for the first time, Security Hub updates the control status every 24 hours based on the findings from the previous 24 hours. On the standard details page and the control details page, Security Hub displays a timestamp to indicate when the status was last updated.

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".

> **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) Region.

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 bottom of the details page contains information about the active findings for the control. Control findings are generated by security checks against the control. The control finding list does not include archived findings.

The finding list uses tabs that display different subsets of the list. On most of the tabs, the finding list shows findings that have a workflow status of `NEW`, `NOTIFIED`, or `RESOLVED`. A separate tab displays SUPPRESSED 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 the section called "Viewing and taking action on control findings".

Viewing details for a control

Choose your preferred access method, and follow these steps to view details for a control. Details apply to the current account and Region and include the following:

- Title and description of the control
- Link to remediation instructions for failed control findings
- Severity of the control
- Enablement status of the control
- (On the console) A list of recent findings for the control. When using the Security Hub API or AWS CLI, use `GetFindings` to retrieve control findings.

Security Hub console

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. Choose **Controls** in the navigation pane.
3. Select a control.

Security Hub 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 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 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 the list of controls

On the **Controls** page, you can see a list of your controls. You can filter and sort the list to focus on a specific subset of controls.

- **All enabled** (controls that are enabled in at least one enabled standard)
- **Failed** (controls with a `Failed` status)
- **Unknown** (controls with an `Unknown` status)
- **Passed** (controls with a `Passed` status)
- **Disabled** (controls that are disabled in all standards)
- **No data** (controls with no findings)
- **All** (all controls, both enabled and disabled, and without regard to control status or findings count)

For more information about control status, see Compliance status and control status.

If you're using the integration with AWS Organizations and are logged in to the AWS Security Hub administrator account, the **All enabled** tab includes controls that are enabled in at least one

member account. If you have set an aggregation Region, the **All enabled** tab includes controls that are enabled in at least one linked Region.

The **Failed** tab is displayed by default. On each tab, the controls are by default sorted by severity, from **Critical** to **Low**. You can also sort controls by control ID, compliance status, severity, or the number of failed checks. The search bar allows you to search for specific controls.

> **Tip**
>
> If you have automated workflows based on 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.

Choosing the option next to the control brings up a side panel which 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 by disabling it in all standards. For more information about enabling and disabling controls across standards, see Enabling and disabling controls in all standards. For administrator accounts, the information presented in the side panel reflects all member accounts.

On the Security Hub API, run `ListSecurityControlDefinitions` to get back a list of control IDs. Once you have the control IDs you are interested in, run `BatchGetSecurityControls` to get data about that subset of controls for the current AWS account and AWS Region.

Viewing and taking action on control findings

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 finding 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 and the configuration timeline.

You can also use the AWS Security Hub API to retrieve a list of findings. For more information, see the section called “Reviewing finding details”.

Topics

Viewing details about a control finding and finding resource

AWS Security Hub provides the following details for each control finding to help you investigate it:

- A history of changes that users have made to the finding
- A `.json` file for the finding
- Information about the resource related to the finding
- The configuration rule related to the finding
- Notes that users have added to the finding

The following section explains how to access these details.

Finding history

Finding history is a Security Hub feature that lets you track changes made to a finding during the last 90 days.

Finding history is available for control findings and other Security Hub findings. For more information, see Reviewing finding history.

Viewing the complete .json for a finding

You can display and download the full `.json` of a finding.

To display the `.json`, in the **Finding .json** column, choose the icon.

On the **Finding JSON** panel, to download the `.json`, choose **Download**.

Viewing information about a finding resource

The **Resource** column contains the resource type and resource identifier.

To display information about the resource, choose the resource identifier. For AWS accounts, if the account is an organization member account, then the information includes both the account ID and the account name. For accounts that were invited manually, the information only includes the account ID.

If you have permission to view the resource in its original service, then the resource identifier displays a link to the service. For example, for an AWS user, the resource details provide a link to the view the user details in IAM.

If the resource is in a different account, Security Hub displays a message to notify you.

Viewing the configuration timeline for a finding resource

One avenue of investigation is the configuration timeline for the resource in AWS Config.

If you have permission to view the configuration timeline for the finding resource, then the finding list provides a link to the timeline.

Security Hub displays a message to notify you if the resource is in a different account.

To navigate to the configuration timeline in AWS Config

1. In the **Investigate** column, choose the icon.
2. On the menu, choose **Configuration timeline**. If you do not have access to the configuration timeline, then the link does not appear.

Viewing the AWS Config rule for a finding resource

If the control is based on an AWS Config rule, then you might also want to view the details for the AWS Config rule. The AWS Config rule information can help you to get a better understanding why a check passed or failed.

If you have permission to view the AWS Config rule for the control, then the finding list provides a link to the AWS Config rule in AWS Config.

Security Hub displays a message to notify you if the resource is in a different account.

To navigate to the AWS Config rule

1. In the **Investigate** column, choose the icon.
2. On the menu, choose **Config rule**. If you do not have access to the AWS Config rule, then **Config rule** is not linked.

Viewing notes for findings

If a finding has an associated note, then the **Updated** column displays a note icon.

To display the note that is associated with a finding

In the **Updated** column, choose the note icon.

Sample control findings

The format of control findings varies depending on whether you've turned on consolidated control findings. When you turn on this feature, Security Hub generates a single finding for a control check even when the control applies to multiple enabled standards. For more information, see Consolidated control findings.

The following section shows sample control findings. These include findings from each Security Hub standard when consolidated control findings is turned off in your account, and a sample control finding across standards when it's turned on.

> **Note**
>
> Findings will reference different fields and values in the China Regions and AWS GovCloud (US) Region. For more information, see Impact of consolidation on ASFF fields and values.

Consolidated control findings is turned off

- Sample finding for AWS Foundational Security Best Practices (FSBP) standard
- Sample finding for Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0
- Sample finding for Center for Internet Security (CIS) AWS Foundations Benchmark v1.4.0
- Sample finding for National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5
- Sample finding for Payment Card Industry Data Security Standard (PCI DSS)
- Sample finding for Service-Managed Standard: AWS Control Tower

Consolidated control findings is turned on

- Sample finding across standards

Sample finding for FSBP


```
{
  "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",
  "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 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",
    "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 v1.2.0

```
{
  "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",
  "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 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",
    "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 CIS AWS Foundations Benchmark v1.4.0

```
{
  "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",
  "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 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",
    "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 NIST SP 800-53 Rev. 5

```
{
  "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",
  "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
 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",
    "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 PCI DSS

```
{
  "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",
  "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 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",
    "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 Service-Managed Standard: AWS Control Tower

> **Note**
>
> This standard is available to you only if you're an AWS Control Tower user who has created the standard in AWS Control Tower. For more information, see Service-Managed Standard: AWS Control Tower.

```
{
  "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",
  "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 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",
    "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 finding across standards (when consolidated control findings is turned on)

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:securityhub:us-east-2:123456789012:security-control/CloudTrail.2/
finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "ProductArn": "arn:aws:securityhub:us-east-2::product/aws/securityhub",
  "ProductName": "Security Hub",
  "CompanyName": "AWS",
  "Region": "us-east-2",
  "GeneratorId": "security-control/CloudTrail.2",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory Standards"
  ],
  "FirstObservedAt": "2022-10-06T02:18:23.076Z",
  "LastObservedAt": "2022-10-28T16:10:06.956Z",
  "CreatedAt": "2022-10-06T02:18:23.076Z",
  "UpdatedAt": "2022-10-28T16:10:00.093Z",
  "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 directions on how to correct this issue, consult the AWS Security
 Hub controls documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/CloudTrail.2/remediation"
    }
  },
  "ProductFields": {
    "RelatedAWSResources:0/name": "securityhub-cloud-trail-encryption-enabled-
fe95bf3f",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
    "aws/securityhub/ProductName": "Security Hub",
    "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:security-control/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",
    "RelatedRequirements": [
        "PCI DSS v3.2.1/3.4",
        "CIS AWS Foundations Benchmark v1.2.0/2.7",
        "CIS AWS Foundations Benchmark v1.4.0/3.7"
    ],
    "SecurityControlId": "CloudTrail.2",
    "AssociatedStandards": [
       { "StandardsId": "standards/aws-foundational-security-best-practices/v/1.0.0"},
       { "StandardsId": "standards/pci-dss/v/3.2.1"},
       { "StandardsId": "ruleset/cis-aws-foundations-benchmark/v/1.2.0"},
       { "StandardsId": "standards/cis-aws-foundations-benchmark/v/1.4.0"},
```

```
        { "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"
      ]
    }
}
```

Filtering, sorting, and downloading control findings

You can filter the list of control findings based on compliance status by using the filtering tabs. You can also filter the list based on other finding field values, and download findings from the list.

Filtering and sorting the control finding list

The **All checks** tab lists all active findings that have a workflow status of `NEW`, `NOTIFIED`, or `RESOLVED`. By default, the list is sorted so that failed findings are at the top of the list. This sort order helps you prioritize findings that need to be addressed.

The lists on the **Failed**, **Unknown**, and **Passed** tabs are filtered based on the value of `Compliance.Status`. The lists also only include active findings that have a workflow status of `NEW`, `NOTIFIED`, or `RESOLVED`.

The **Suppressed** tab contains a list of active findings that have a workflow status of `SUPPRESSED`.

In addition to the built-in filters on each tab, you can filter the lists using values from the following fields:

- Account ID
- Workflow status
- Compliance status

- Resource ID
- Resource type

You can sort each list using any of the columns.

Downloading the control finding list

If you navigate to **Security standards** and choose a standard, you see a list of controls for the standard. Choosing a control from the list takes you to the control details page with a list of findings for the control. From here, you can download 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.

To download the findings, choose **Download**. The current page of findings is downloaded.

Taking action on control findings

To reflect the current status of your investigation, you set the workflow status. For more information, see the section called “Setting the workflow status of findings”.

In AWS Security Hub, you can also send selected findings to a custom action in Amazon EventBridge. For more information, see the section called “Sending findings to a custom action”.

Working with the Summary dashboard

On the AWS Security Hub console, the dashboard on the **Summary** page can help you identify areas of security concern in your AWS environment, without the need for additional analytics tools or complex queries. You can customize the dashboard layout, add or remove widgets, and filter the data to focus on areas of particular interest. You can also save your filter criteria as a filter set to quickly retrieve specific types of data in the future.

If you customize the dashboard or filter the data, Security Hub automatically saves your settings for subsequent use. In addition, the settings are saved independently for each user of your Security Hub account. This means that different users can have different layouts, widgets, and filter sets for the dashboard.

Each time you open the **Summary** dashboard, Security Hub automatically refreshes most dashboard data. However, some of the data is updated less frequently. For example, security scores and control statuses are updated every 24 hours.

If you configured a cross-Region aggregation Region for Security Hub, your dashboard data includes findings from the aggregation Region and all linked Regions. If you're the delegated Security Hub administrator for an organization, the data includes findings for your administrator account and your member accounts. You can optionally filter the data by account. If you have a member account or a standalone account, the data includes findings only for your account.

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 them, choose **Add widget** at the top right of the **Summary** page. In the search bar, enter the title of the widget. Drag and drop the widget on to the dashboard.

Widgets shown by default

By default, the **Summary** dashboard includes the following widgets:

Security standards

Displays your most recent summary security score and the security score for each Security Hub 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 How security scores are calculated. 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 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 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 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 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 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 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 then opens the console for that service.

Filtering the Summary dashboard

To curate data on the **Summary** dashboard and include only the security data that's most relevant to you, you can filter the dashboard. 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 filter data on the **Summary** dashboard, you enter filter criteria in the filter box above the dashboard. If you apply filter criteria, the criteria applies to all the data on the dashboard except the data in the **Insights** and **Security standards** widgets.

You can filter the data by using the following fields:

- Account name
- Account ID
- Application Amazon Resource Name (ARN)
- Application name
- Product name (for an AWS service or third-party product that sends findings to Security Hub)
- 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 `Record state` 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.

> **Note**
>
> The following fields can't be saved as part of a filter set: 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 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 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 view.

To update or delete a filter set

1. Open the AWS Security Hub 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

You can customize the **Summary** dashboard in several ways. You can add and remove widgets from the dashboard. You can also rearrange and resize widgets on the dashboard.

If you customize the dashboard, Security Hub 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 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.

Creating Security Hub resources with AWS CloudFormation

AWS Security Hub 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 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 and AWS CloudFormation templates

To provision and configure resources for Security Hub 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 resources:

- Enabling Security Hub
- Designating the delegated Security Hub administrator for an organization
- Enabling a security standard
- Creating a custom insight
- Creating an automation rule
- Subscribing to a third-party product integration

For more information, including examples of JSON and YAML templates for resources, see the AWS Security Hub 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 announcements with Amazon Simple Notification Service

This section provides information about subscribing to AWS Security Hub announcements with Amazon Simple Notification Service (Amazon SNS) to receive notifications about Security Hub.

After subscribing, you will receive notifications about the following events (note the corresponding `AnnouncementType` for each event):

- `GENERAL` – General notifications about the Security Hub service.
- `UPCOMING_STANDARDS_CONTROLS` – Specified Security Hub 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 is available in a new AWS Region.
- `NEW_STANDARDS_CONTROLS` – New Security Hub controls or standards have been added.
- `UPDATED_STANDARDS_CONTROLS` – Existing Security Hub controls or standards have been updated.
- `RETIRED_STANDARDS_CONTROLS` – Existing Security Hub 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 features are available.
- `UPDATED_FEATURE` – Existing Security Hub features have been updated.

Notifications are available in all formats that Amazon SNS supports. You can subscribe to Security Hub announcements in all AWS Regions that Security Hub 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 sends Amazon SNS announcements about updates to the Security Hub service to any subscribed AWS account. To receive notifications about Security Hub findings, see Managing and reviewing finding details and history.

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 Tutorial: Subscribing an Amazon SQS 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`

AWS Region	Amazon SNS topic ARN
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
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

AWS Region	Amazon SNS topic ARN
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
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

AWS Region	Amazon SNS topic ARN
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
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 announcements.

For information about the cost of subscribing to Security Hub announcements, see Amazon SNS pricing.

Subscribing to Security Hub 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 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 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 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 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 announcement for new controls (email protocol)

```
{
"AnnouncementType":"NEW_STANDARDS_CONTROLS",
"Title":"[New Controls] 36 new Security Hub 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 to automatically enable new controls, these controls are enabled by default.
 Availability of controls can vary by Region. "
}
```

Example: Security Hub 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 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 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"
}
```

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.
- 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 140-2 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) 140-2.

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 AWS 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 AWS 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 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 Signing AWS 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 Using multi-factor authentication (MFA) in AWS 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 When to create an IAM user (instead of a role) 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. You can temporarily assume an IAM role in the AWS Management Console by switching roles. 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 Using IAM roles 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 Creating a role for a third-party Identity Provider 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 How IAM roles differ from resource-based policies 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 Creating 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 Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) 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 Creating IAM 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 Choosing 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 How SCPs work 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 AWS Security Hub works with IAM

Before you use AWS Identity and Access Management to manage access to Security Hub, learn which IAM features are available to use with Security Hub.

IAM features you can use with Amazon Macie

IAM feature	Macie support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	No

IAM feature	Macie support
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 Creating IAM 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 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 How IAM roles differ from resource-based policies 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 Security Hub.

Resources

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 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 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 What is ABAC? 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 AWS 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 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 Switching to a 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 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 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 Security Hub.

Identity-based policy examples for 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

- Example: Allow users to view their own permissions
- Example: Allow users to create and manage a configuration policy
- Example: Allow users to view findings
- Example: Allow users to create and manage automation rules

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 IAM Access Analyzer policy validation 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 Configuring MFA-protected API access 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*:

```
{
    "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.

```
{
    "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.

```
{
    "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.

```
{
    "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 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 the permissions are 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 view the details of the service-linked role, on the **Settings** page of the Security Hub console, choose **General** and then **View service permissions**.

You can delete the Security Hub service-linked role only after first disabling Security Hub in all 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 Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Topics

- Service-linked role permissions for Security Hub
- Creating a service-linked role for Security Hub
- Editing a service-linked role for Security Hub
- Deleting a service-linked role for Security Hub

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. The service-linked role lets Security Hub 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 following services to assume the role:

- `securityhub.amazonaws.com`

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 role, group, or user.

```
{
    "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 automatically created when you enable Security Hub for the first time or enable Security Hub in a supported Region where you previously didn't have it enabled. You can also create the `AWSServiceRoleForSecurityHub` service-linked role manually using the IAM console, the IAM CLI, or the IAM API.

> **Important**
>
> The service-linked role that is created for the Security Hub administrator account doesn't apply to the Security Hub member accounts.

For more information about creating the role manually, see Creating a service-linked role in the *IAM User Guide*.

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 that role. That way, you don't have an unused entity that isn't actively monitored or maintained.

> **Important**
>
> To delete the `AWSServiceRoleForSecurityHub` service-linked role, you must first disable Security Hub in all Regions where it's enabled.
> If Security Hub isn't disabled when you try to delete the service-linked role, the deletion fails. For more information, see Disabling Security Hub.

When you disable Security Hub, the `AWSServiceRoleForSecurityHub` service-linked role is *not* automatically deleted. If you enable Security Hub again, it starts using the existing `AWSServiceRoleForSecurityHub` service-linked role.

To manually delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the `AWSServiceRoleForSecurityHub` service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

AWS managed policies for AWS 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 actions. This policy must be attached to a principal before they enable Security Hub manually for their account. For example, principals with these permissions can both view and update the status of findings. They can configure custom insights, and enable integrations. They can 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 actions.
- `guardduty` – Allows principals to get information about account status in Amazon GuardDuty.
- `iam` – Allows principals to create a service-linked role.
- `inspector` – Allows principals to get information about account status in Amazon Inspector.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SecurityHubAllowAll",
            "Effect": "Allow",
            "Action": "securityhub:*",
            "Resource": "*"
        },
        {
            "Sid": "SecurityHubServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
```

```
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "securityhub.amazonaws.com"
                }
            }
        },
        {
            "Sid": "OtherServicePermission",
            "Effect": "Allow",
            "Action": [
                "guardduty:GetDetector",
                "guardduty:ListDetectors",
                "inspector2:BatchGetAccountStatus"
            ],
            "Resource": "*",
        }
    ]
}
```

Security Hub 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. Principals with this policy attached cannot make any updates in Security Hub. 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 either a list of items or details about an item. This includes API operations that start with `Get`, `List`, or `Describe`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "AWSSecurityHubReadOnlyAccess",
            "Effect": "Allow",
            "Action": [
                "securityhub:Get*",
                "securityhub:List*",
                "securityhub:BatchGet*",
                "securityhub:Describe*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS managed policy: AWSSecurityHubOrganizationsAccess

You can attach the `AWSSecurityHubOrganizationsAccess` policy to your IAM identities.

This policy grants administrative permissions in AWS Organizations that are required to support the Security Hub integration with Organizations.

These permissions allow the organization management account to designate the delegated administrator account for Security Hub. They also allow the delegated Security Hub administrator account to enable organization accounts as member accounts.

This policy only provides the permissions for Organizations. The organization management account and delegated Security Hub administrator account also require permissions for the associated actions in Security Hub. 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: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:DescribeOrganization` – Allows principals to retrieve information about the organization configuration.
- `organizations:EnableAWSServiceAccess` – Allows principals to enable the Security Hub integration with Organizations.
- `organizations:RegisterDelegatedAdministrator` – Allows principals to designate the delegated administrator account for Security Hub.
- `organizations:DeregisterDelegatedAdministrator` – Allows principals to remove the delegated administrator account for Security Hub.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "OrganizationPermissions",
            "Effect": "Allow",
            "Action": [
                "organizations:ListAccounts",
                "organizations:DescribeOrganization",
                "organizations:ListRoots",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListAccountsForParent",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganizationalUnit"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "OrganizationPermissionsEnable",
            "Effect": "Allow",
            "Action": "organizations:EnableAWSServiceAccess",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "organizations:ServicePrincipal": "securityhub.amazonaws.com"
                }
            }
        },
        {
            "Sid": "OrganizationPermissionsDelegatedAdmin",
            "Effect": "Allow",
            "Action": [
                "organizations:RegisterDelegatedAdministrator",
                "organizations:DeregisterDelegatedAdministrator"
            ],
            "Resource": "arn:aws:organizations::*:account/o-*/*",
            "Condition": {
            "StringEquals": {
                "organizations:ServicePrincipal": "securityhub.amazonaws.com"
                }
            }
        }
    ]
}
```

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 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 the security checks for Security Hub controls.

Permissions details

This policy includes permissions to do the following:

- `cloudtrail` – Retrieve information about CloudTrail trails.

- `cloudwatch` – Retrieve the current CloudWatch alarms.
- `logs` – Retrieve the 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` – Get 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 service, standards, and controls are configured.
- `tag` – Retrieve information about resource tags.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SecurityHubServiceRolePermissions",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:DescribeTrails",
                "cloudtrail:GetTrailStatus",
                "cloudtrail:GetEventSelectors",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeAlarmsForMetric",
                "logs:DescribeMetricFilters",
                "sns:ListSubscriptionsByTopic",
                "config:DescribeConfigurationRecorders",
                "config:DescribeConfigurationRecorderStatus",
                "config:DescribeConfigRules",
                "config:DescribeConfigRuleEvaluationStatus",
                "config:BatchGetResourceConfig",
                "config:SelectResourceConfig",
                "iam:GenerateCredentialReport",
                "organizations:ListAccounts",
                "config:PutEvaluations",
                "tag:GetResources",
                "iam:GetCredentialReport",
                "organizations:DescribeAccount",
```

```
                "organizations:DescribeOrganization",
                "organizations:ListChildren",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:DescribeOrganizationalUnit",
                "securityhub:BatchDisableStandards",
                "securityhub:BatchEnableStandards",
                "securityhub:BatchUpdateStandardsControlAssociations",
                "securityhub:BatchGetSecurityControls",
                "securityhub:BatchGetStandardsControlAssociations",
                "securityhub:CreateMembers",
                "securityhub:DeleteMembers",
                "securityhub:DescribeHub",
                "securityhub:DescribeOrganizationConfiguration",
                "securityhub:DescribeStandards",
                "securityhub:DescribeStandardsControls",
                "securityhub:DisassociateFromAdministratorAccount",
                "securityhub:DisassociateMembers",
                "securityhub:DisableSecurityHub",
                "securityhub:EnableSecurityHub",
                "securityhub:GetEnabledStandards",
                "securityhub:ListStandardsControlAssociations",
                "securityhub:ListSecurityControlDefinitions",
                "securityhub:UpdateOrganizationConfiguration",
                "securityhub:UpdateSecurityControl",
                "securityhub:UpdateSecurityHubConfiguration",
                "securityhub:UpdateStandardsControl",
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SecurityHubServiceRoleConfigPermissions",
            "Effect": "Allow",
            "Action": [
                "config:PutConfigRule",
                "config:DeleteConfigRule",
                "config:GetComplianceDetailsByConfigRule"
            ],
            "Resource": "arn:aws:config:*:*:config-rule/aws-service-rule/*securityhub*"
        },
        {
            "Sid": "SecurityHubServiceRoleOrganizationsPermissions",
            "Effect": "Allow",
            "Action": [
```

```
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "organizations:ServicePrincipal": [
                        "securityhub.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Security Hub updates to AWS managed policies

View details about updates to AWS managed policies for Security Hub since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Security Hub Document history page.

Change	Description	Date
AWSSecurityHubReadOnlyAccess – Update to an existing policy	Security Hub updated this managed policy by adding a `Sid` field.	February 22, 2024
AWSSecurityHubFullAccess – Update to an existing policy	Security Hub 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

Change	Description	Date
AWSSecurityHubOrganizationsAccess – Update to an existing policy	Security Hub 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
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub 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 added the `tag:GetResources` permission to read resource tags related to findings.	November 7, 2023
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub added the `BatchGetStandardsControlAssociations` permission to get information about the enablement status of a control in a standard.	September 27, 2023

Change	Description	Date
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub added new permissions to get AWS Organizations data and read and update Security Hub configurations, including standards and controls.	September 20, 2023
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub 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 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 added a new permission to allow the service-linked role to deliver evaluation results to AWS Config.	June 29, 2021

Change	Description	Date
AWSSecurityHubServiceRolePolicy – Added to the list of managed policies	Added information about the managed policy AWSSecurityHubServiceRolePolicy, which is used by the Security Hub service-linked role.	June 11, 2021
AWSSecurityHubOrganizationsAccess – New policy	Security Hub added a new policy that grants permissions that are needed for the Security Hub integration with Organizations.	March 15, 2021
Security Hub started tracking changes	Security Hub 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 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. • 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*.
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

Which user needs programmatic access?	To	By
		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 Creating 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 Creating 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 AWS 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 How IAM roles differ from resource-based policies in the *IAM User Guide*.

Compliance validation for AWS Security Hub

Third-party auditors assess the security and compliance of AWS Security Hub as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. 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 Security Hub 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 and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Config – This AWS 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 that helps you check your compliance with security industry standards and best practices.

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 Interface VPC endpoints (AWS PrivateLink) in the *AWS PrivateLink Guide*.

Considerations for Security Hub VPC endpoints

Before you set up an interface VPC endpoint for Security Hub, ensure that you review Interface endpoint properties and limitations in the *AWS PrivateLink Guide*.

Security Hub supports making calls to all of its API actions from your VPC.

> **Note**
>
> Security Hub does not support VPC endpoints in the Asia Pacific (Osaka) Region.

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 an interface endpoint in the *AWS PrivateLink Guide*.

Create a VPC endpoint for Security Hub using the following service name:

- com.amazonaws.*region*.securityhub

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 more information, see Access a service through an interface endpoint in the *AWS PrivateLink Guide*.

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 services with VPC endpoints in the *AWS PrivateLink 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 with other accounts in the *Amazon VPC User Guide*.

Logging AWS Security Hub API calls with AWS CloudTrail

AWS Security Hub 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. CloudTrail captures API calls for Security Hub as events. The captured calls include calls from the Security Hub console and code calls to the Security Hub 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. 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, 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 information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in Security Hub, 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, 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 supports logging all of the Security Hub API actions as events in CloudTrail logs. To view a list of Security Hub operations, see the Security Hub 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 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 Insights in AWS Security Hub.

```
{
```

```
    "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 AWS 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 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 assign tags to the following types of Security Hub resources: automation rules, configuration policies, and the Hub resource.

Topics

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 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, the tags are applied only to the resource that's stored directly in Security Hub in the applicable AWS Region. They aren't applied to any associated, supporting resources that Security Hub 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 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 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 resources, you can use the Security Hub console, the Security Hub API, or the AWS Resource Groups Tagging API. With Security Hub, 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.

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 resources, if the `Owner` tag for the resource specifies their username:

```
{
    "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 AWS Security Hub resources

To add tags to an individual AWS Security Hub resource, you can use the Security Hub console or the Security Hub API. The console doesn't support adding tags to the Hub resource.

To add tags to multiple Security Hub 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 a tag to a resource

When you create an automation rule or a configuration policy, the Security Hub console provides options for adding tags to it. You can provide the tag key and tag value in the **Tags** section.

Security Hub API & AWS CLI

To add a tag to a resource

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 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 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 key=Environment,value=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 key=Environment,value=Prod key=CostCenter,value=12345 key=Owner,value=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 key=Owner,value=
```

If a tagging operation succeeds, Security Hub returns an empty HTTP 200 response. Otherwise, Security Hub returns an HTTP 4*xx* or 500 response that indicates why the operation failed.

Reviewing tags for AWS Security Hub resources

You can review the tags (both tag keys and tag values) for a Security Hub automation rule or configuration policy by using the Security Hub console or the Security Hub API. The console doesn't support reviewing tags for the Hub resource.

To review tags for multiple Security Hub resources at the same time, use the tagging operations of the AWS Resource Groups Tagging API.

Console

To review the tags for a resource

1. Using the credentials of the Security Hub administrator, open the AWS Security Hub 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 API & AWS CLI

To review the tags for a resource

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 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 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 resource that has tags. Each object specifies the ARN of a Security Hub resource, and the tag keys and values that are assigned to the resource.

Editing tags for AWS Security Hub resources

To edit tags (tag keys or tag values) for an AWS Security Hub resource, you can use the Security Hub API. The Security Hub console currently doesn't support tag editing.

To edit tags for multiple Security Hub resources at the same time, use the tagging operations of the AWS Resource Groups Tagging API.

> **Important**
>
> Editing the tags for a resource can affect access to the resource. Before you edit a tag key or value for a resource, review any AWS Identity and Access Management (IAM) policies that might use the tag to control access to resources.

Security Hub API & AWS CLI

To edit the tags for a resource

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 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 key=Environment,value=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 key=Owner,value=
```

If the operation succeeds, Security Hub returns an empty HTTP 200 response. Otherwise, Security Hub returns an HTTP 4*xx* or 500 response that indicates why the operation failed.

Removing tags from AWS Security Hub resources

To remove tags from an AWS Security Hub resource, you can use the Security Hub API. The Security Hub console currently doesn't support tag removal.

To remove tags from multiple Security Hub 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 API & AWS CLI

To remove tags from a resource

To remove one or more tags from a resource programmatically, use the UntagResource operation of the Security Hub 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 returns an empty HTTP 200 response. Otherwise, Security Hub returns an HTTP 4*xx* or 500 response that indicates why the operation failed.

Security Hub quotas

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 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 Security Hub API operations, see the *AWS Security Hub API Reference*.

If you have set up Cross-Region aggregation, 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.

Security Hub Regional limits

Some AWS Security Hub features are available in only certain AWS Regions. The following sections specify these Regional limits.

For a list of Regions in which Security Hub is available, see AWS Security Hub endpoints and quotas in the *AWS General Reference*.

Cross-Region aggregation restrictions

In AWS GovCloud (US), cross-Region aggregation is available for findings, finding updates, and insights across AWS GovCloud (US) only. 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 available for findings, finding updates, and insights across the China Regions only. Specifically, you can only aggregate findings, finding updates, and insights between China (Beijing) and China (Ningxia).

You can't use a Region that is disabled by default as your aggregation Region. For a list of Regions that are disabled by default, see Enabling a Region in the *AWS General Reference*.

Availability of integrations by Region

Some integrations are not available in all Regions. If an integration is not available in a specific Region, it is not listed on the **Integrations** page of the Security Hub console when you choose that Region.

Integrations that are supported in China (Beijing) and China (Ningxia)

The China (Beijing) and China (Ningxia) Regions only support 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

The China (Beijing) and China (Ningxia) Regions only support 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 that are supported in AWS GovCloud (US-East) and AWS GovCloud (US-West)

The AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions only support 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

The AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions only support 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

Service-Managed Standard: AWS Control Tower is only available in Regions that AWS Control Tower supports, including AWS GovCloud (US). For a list of Regions that AWS Control Tower supports, see How AWS Regions Work With AWS Control Tower in the *AWS Control Tower User Guide*.

Other security standards are available in all Regions that Security Hub is available in.

Availability of controls by Region

Security Hub controls may not be available in all Regions. To see a list of unavailable controls in each Region, see Regional limits on controls. A control doesn't appear on the list of controls in the Security Hub console if it's not available in the Region that you're signed in to. The exception is if you're signed in to an aggregation Region. In that case, you can see controls that are available in the aggregation Region or in one or more linked Regions.

Regional limits on controls

AWS Security Hub controls may not be available in all AWS Regions. This page shows which controls are unavailable in specific Regions. A control doesn't appear on the list of controls in the Security Hub console if it's not available in the Region that you are signed in to. The exception is if you're signed in to an aggregation Region. In that case, you can see controls that are available in the aggregation Region or in one or more linked Regions.

Contents

- Asia Pacific (Hyderabad)
- Asia Pacific (Jakarta)
- Asia Pacific (Melbourne)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- 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)
- Middle East (Bahrain)
- Middle East (UAE)
- South America (São Paulo)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

US East (Ohio)

The following controls are not supported in US East (Ohio).

- [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
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [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

US East (N. Virginia)

The following controls are not supported in US East (N. Virginia).

- [ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [ElastiCache.7] ElastiCache clusters should not use the default subnet group

US West (N. California)

The following controls are not supported in US West (N. California).

- [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
- [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
- [EKS.1] EKS cluster endpoints should not be publicly accessible
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [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

US West (Oregon)

The following controls are not supported in US West (Oregon).

- [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
- [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 Africa (Cape Town).

- [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
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [DMS.1] Database Migration Service replication instances should not be public
- [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
- [EC2.3] Attached Amazon EBS volumes should be encrypted at-rest
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)
- [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
- [EC2.14] 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
- [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
- [EKS.1] EKS cluster endpoints should not be publicly accessible
- [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.4] Application Load Balancer should be configured to drop http headers
- [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
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [ES.3] Elasticsearch domains should encrypt data sent between nodes
- [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
- [GuardDuty.1] GuardDuty should be enabled
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [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.9] RDS DB instances should publish logs to CloudWatch Logs
- [RDS.10] IAM authentication should be configured for RDS instances
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet 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
- [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
- [WAF.11] AWS WAF web ACL logging should be enabled

Asia Pacific (Hong Kong)

The following controls are not supported in Asia Pacific (Hong Kong).

- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [EventBridge.4] EventBridge global endpoints should have event replication enabled
- [RDS.10] IAM authentication should be configured for RDS instances
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [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

Asia Pacific (Hyderabad)

The following controls are not supported in Asia Pacific (Hyderabad).

- [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
- [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.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [Athena.1] Athena workgroups should be encrypted at rest
- [AutoScaling.1] Auto Scaling groups associated with a Classic Load Balancer should use load balancer health checks

- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeBuild.5] CodeBuild project environments should not have privileged mode 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
- [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
- [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
- [EC2.18] 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
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.51] EC2 Client VPN endpoints should have client connection logging 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.9] ECS task definitions should have a logging configuration
- [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.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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.5] Application and Classic Load Balancers logging should be enabled
- [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
- [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
- [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
- [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

- [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.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
- [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
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible AWS Configuration
- [RDS.7] RDS clusters should have deletion protection enabled
- [RDS.9] RDS DB instances should publish logs to CloudWatch Logs
- [RDS.12] IAM authentication should be configured for RDS clusters
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.16] RDS DB clusters should be configured to copy tags to snapshots
- [RDS.24] RDS Database clusters 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
- [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.10] Redshift clusters should be encrypted at rest
- [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
- [S3.17] 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

- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [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
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [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

Asia Pacific (Jakarta)

The following controls are not supported in Asia Pacific (Jakarta).

- [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
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys

- [AutoScaling.3] Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)
- [AutoScaling.4] Auto Scaling group launch configuration should not have a metadata response hop limit greater than 1
- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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
- [CodeBuild.5] CodeBuild project environments should not have privileged mode 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
- [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
- [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
- [EC2.18] 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
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.51] EC2 Client VPN endpoints should have client connection logging 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.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
- [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.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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.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
- [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
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [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
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [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.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.9] RDS DB instances should publish logs to CloudWatch Logs
- [RDS.14] Amazon Aurora clusters should have backtracking enabled

- [RDS.16] RDS DB clusters should be configured to copy tags to snapshots
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [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.7] Redshift clusters should use enhanced VPC routing
- [Redshift.9] Redshift clusters should not use the default database name
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.11] S3 general purpose buckets should have event notifications enabled
- [S3.13] S3 general purpose 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
- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [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
- [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

Asia Pacific (Melbourne)

The following controls are not supported in Asia Pacific (Melbourne).

- [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.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
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [Athena.1] Athena workgroups should be encrypted at rest
- [AutoScaling.1] Auto Scaling groups associated with a Classic Load Balancer should use load balancer health checks
- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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.5] CodeBuild project environments should not have privileged mode 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
- [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
- [EC2.1] Amazon EBS snapshots should not be publicly restorable
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)
- [EC2.9] Amazon EC2 instances should not have a public IPv4 address
- [EC2.13] 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
- [EC2.18] 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
- [EC2.28] EBS volumes should be covered by a backup plan
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions.
- [ECS.9] ECS task definitions should have a logging configuration
- [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
- [EKS.8] EKS clusters should have audit logging enabled
- [ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.2] ElastiCache for Redis cache clusters should have auto minor version upgrade enabled
- [ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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.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
- [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
- [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.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 AWS 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
- [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

- [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.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
- [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
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [RDS.1] RDS snapshot should be private
- [RDS.3] RDS DB instances should have encryption at-rest enabled
- [RDS.7] RDS clusters should have deletion protection enabled
- [RDS.12] IAM authentication should be configured for RDS clusters
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.16] RDS DB clusters should be configured to copy tags to snapshots
- [RDS.24] RDS Database clusters 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
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.14] S3 general purpose buckets should have versioning enabled
- [S3.15] S3 general purpose buckets should have Object Lock 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
- [SNS.1] SNS topics should be encrypted at-rest using AWS KMS
- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [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
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [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
- [WAF.11] AWS WAF web ACL logging should be enabled

Asia Pacific (Mumbai)

The following controls are not supported in Asia Pacific (Mumbai).

- [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
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [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

Asia Pacific (Osaka)

The following controls are not supported in Asia Pacific (Osaka).

- [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
- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [CloudWatch.15] CloudWatch alarms should have specified actions configured
- [CloudWatch.16] CloudWatch log groups should be retained for a specified time period

- [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.5] CodeBuild project environments should not have privileged mode enabled
- [DMS.1] Database Migration Service replication instances should not be public
- [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
- [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.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
- [EC2.1] Amazon EBS snapshots should not be publicly restorable
- [EC2.3] Attached Amazon EBS volumes should be encrypted at-rest
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.7] 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
- [EC2.13] 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

- [EC2.17] Amazon EC2 instances should not use multiple ENIs
- [EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports
- [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
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECR.2] ECR private repositories should have tag immutability 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.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.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
- [EKS.1] EKS cluster endpoints should not be publicly accessible
- [EKS.2] EKS clusters should run on a supported Kubernetes version
- [ElastiCache.1] ElastiCache Redis clusters should have automatic backup 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 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.9] Classic Load Balancers should have cross-zone load balancing enabled
- [ELB.16] Application Load Balancers should be associated with an AWS WAF web ACL
- [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
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [GuardDuty.1] GuardDuty should be enabled
- [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.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
- [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
- [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
- [RDS.1] RDS snapshot should be private
- [RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest
- [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.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.26] RDS DB instances should be protected by a backup plan
- [RDS.35] RDS DB clusters should have automatic minor version upgrade 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.7] Redshift clusters should use enhanced VPC routing
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.8] S3 general purpose buckets should block public access
- [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
- [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.3] Remove unused Secrets Manager secrets
- [SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days
- [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
- [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
- [WAF.11] AWS WAF web ACL logging should be enabled

Asia Pacific (Seoul)

The following controls are not supported in Asia Pacific (Seoul).

- [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
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [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

Asia Pacific (Singapore)

The following controls are not supported in Asia Pacific (Singapore).

- [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
- [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

Asia Pacific (Sydney)

The following controls are not supported in Asia Pacific (Sydney).

- [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
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [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

Asia Pacific (Tokyo)

The following controls are not supported in Asia Pacific (Tokyo).

- [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
- [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

Canada (Central)

The following controls are not supported in Canada (Central).

- [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
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [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

Canada West (Calgary)

The following controls are not supported in Canada West (Calgary).

- [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
- [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.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys

- [Athena.1] Athena workgroups should be encrypted at rest
- [AutoScaling.1] Auto Scaling groups associated with a Classic Load Balancer should use load balancer 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.4] Auto Scaling group launch configuration should not have a metadata response hop limit greater than 1
- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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.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
- [CodeBuild.5] CodeBuild project environments should not have privileged mode 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
- [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.6] DynamoDB tables should have deletion protection enabled
- [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

- [EC2.51] EC2 Client VPN endpoints should have client connection logging 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
- [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
- [EKS.8] EKS clusters should have audit logging enabled
- [ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.2] ElastiCache for Redis cache clusters should have auto minor version upgrade enabled
- [ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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.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
- [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.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
- [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.10] Password policies for IAM users should have strong AWS 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
- [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
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Macie.2] Macie automated sensitive data discovery should be 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
- [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

- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.16] RDS DB clusters should be configured to copy tags to 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.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade 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
- [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
- [S3.17] 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
- [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
- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [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
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [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

China (Beijing)

The following controls are not supported in China (Beijing).

- [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
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [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
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [CloudWatch.15] CloudWatch alarms should have specified actions configured
- [CloudWatch.16] CloudWatch log groups should be retained for a specified time period
- [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
- [EC2.15] Amazon EC2 subnets should not automatically assign public IP addresses
- [EC2.16] Unused Network Access Control Lists should be removed
- [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
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions.
- [ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled
- [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.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
- [EMR.2] Amazon EMR block public access setting should be enabled
- [ES.3] Elasticsearch domains should encrypt data sent between nodes
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [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
- [GuardDuty.1] GuardDuty 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
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be 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
- [RDS.7] RDS clusters should have deletion protection enabled
- [RDS.10] IAM authentication should be configured for RDS instances
- [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] RDS DB clusters should be configured to copy tags to 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.34] Aurora MySQL DB clusters should publish audit logs to CloudWatch Logs
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [Redshift.7] Redshift clusters should use enhanced VPC routing
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.1] S3 general purpose buckets should have block public access settings enabled
- [S3.8] S3 general purpose buckets should block public access
- [S3.14] S3 general purpose buckets should have versioning enabled
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SecretsManager.3] Remove unused Secrets Manager secrets
- [SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days
- [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.11] AWS WAF web ACL logging should be enabled

China (Ningxia)

The following controls are not supported in China (Ningxia).

- [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
- [ACM.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [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
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [CloudWatch.15] CloudWatch alarms should have specified actions configured
- [CloudWatch.16] CloudWatch log groups should be retained for a specified time period
- [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
- [EC2.15] Amazon EC2 subnets should not automatically assign public IP addresses
- [EC2.16] Unused Network Access Control Lists should be removed
- [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
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [ECR.1] ECR private repositories should have image scanning configured
- [ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions.
- [EFS.3] EFS access points should enforce a root directory
- [EFS.4] EFS access points should enforce a user identity
- [ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled
- [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.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
- [EMR.2] Amazon EMR block public access setting should be enabled
- [ES.1] Elasticsearch domains should have encryption at-rest enabled
- [ES.3] Elasticsearch domains should encrypt data sent between nodes
- [ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled
- [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
- [GuardDuty.1] GuardDuty 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
- [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
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be 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.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
- [RDS.7] RDS clusters 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.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.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.35] RDS DB clusters should have automatic minor version upgrade enabled
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Redshift.7] Redshift clusters should use enhanced VPC routing
- [Redshift.10] Redshift clusters should be encrypted at rest
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.1] S3 general purpose buckets should have block public access settings enabled
- [S3.8] S3 general purpose buckets should block public access
- [S3.14] S3 general purpose buckets should have versioning enabled
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access
- [SecretsManager.3] Remove unused Secrets Manager secrets
- [SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days
- [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.11] AWS WAF web ACL logging should be enabled

Europe (Frankfurt)

The following controls are not supported in Europe (Frankfurt).

- [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
- [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

Europe (Ireland)

The following controls are not supported in Europe (Ireland).

- [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

- [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

Europe (London)

The following controls are not supported in Europe (London).

- [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
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [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

Europe (Milan)

The following controls are not supported in Europe (Milan).

- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [CodeBuild.1] CodeBuild Bitbucket source repository URLs should not contain sensitive credentials
- [CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials
- [DMS.1] Database Migration Service replication instances should not be public
- [DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [EC2.3] Attached Amazon EBS volumes should be encrypted at-rest
- [EC2.4] Stopped EC2 instances should be removed after a specified time period
- [EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)
- [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
- [EC2.14] 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
- [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
- [EKS.1] EKS cluster endpoints should not be publicly accessible
- [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.4] Application Load Balancer should be configured to drop http headers
- [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
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [ES.3] Elasticsearch domains should encrypt data sent between nodes
- [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
- [GuardDuty.1] GuardDuty should be enabled
- [IAM.3] IAM users' access keys should be rotated every 90 days or less
- [IAM.18] Ensure a support role has been created to manage incidents with AWS Support
- [KMS.3] AWS KMS keys should not be deleted unintentionally
- [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
- [RDS.1] RDS snapshot should be private
- [RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest
- [RDS.9] RDS DB instances should publish logs to CloudWatch Logs
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [SageMaker.1] Amazon SageMaker notebook instances should not have direct internet 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
- [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
- [WAF.11] AWS WAF web ACL logging should be enabled

Europe (Paris)

The following controls are not supported in Europe (Paris).

- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [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

Europe (Spain)

The following controls are not supported in Europe (Spain).

- [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
- [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.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [AppSync.2] AWS AppSync should have field-level logging enabled

- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [Athena.1] Athena workgroups should be encrypted at rest
- [AutoScaling.1] Auto Scaling groups associated with a Classic Load Balancer should use load balancer health checks
- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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.16] CloudWatch log groups should be retained for a specified time period
- [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.5] CodeBuild project environments should not have privileged mode 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
- [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
- [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
- [EC2.13] 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
- [EC2.17] Amazon EC2 instances should not use multiple ENIs
- [EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports

- [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
- [EC2.51] EC2 Client VPN endpoints should have client connection logging 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.9] ECS task definitions should have a logging configuration
- [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.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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 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.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.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
- [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
- [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
- [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.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
- [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.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
- [Macie.1] Amazon Macie should be enabled
- [Macie.2] Macie automated sensitive data discovery should be 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
- [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
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [RDS.1] RDS snapshot should be private
- [RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible AWS 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] RDS DB clusters should be configured to copy tags to snapshots
- [RDS.24] RDS Database clusters 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
- [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.10] Redshift clusters should be encrypted at rest
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.1] S3 general purpose buckets should have block public access settings enabled
- [S3.5] 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
- [S3.15] S3 general purpose buckets should have Object Lock enabled
- [S3.17] S3 general purpose buckets should be encrypted at rest with AWS KMS keys
- [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.2] Secrets Manager secrets configured with automatic rotation should rotate successfully
- [SNS.1] SNS topics should be encrypted at-rest using AWS KMS
- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [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
- [StepFunctions.1] Step Functions state machines should have logging turned on

- [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

Europe (Stockholm)

The following controls are not supported in Europe (Stockholm).

- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [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

Europe (Zurich)

The following controls are not supported in Europe (Zurich).

- [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.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [Athena.1] Athena workgroups should be encrypted at rest
- [AutoScaling.1] Auto Scaling groups associated with a Classic Load Balancer should use load balancer health checks
- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)

- [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
- [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
- [CodeBuild.4] CodeBuild project environments should have a logging AWS Configuration
- [CodeBuild.5] CodeBuild project environments should not have privileged mode 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
- [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
- [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
- [EC2.8] 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
- [EC2.13] 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
- [EC2.17] Amazon EC2 instances should not use multiple ENIs
- [EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports
- [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
- [EC2.51] EC2 Client VPN endpoints should have client connection logging 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.9] ECS task definitions should have a logging configuration

- [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.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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 http headers
- [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.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
- [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
- [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
- [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.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
- [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
- [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.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
- [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
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [RDS.1] RDS snapshot should be private
- [RDS.3] RDS DB instances should have encryption at-rest enabled
- [RDS.5] RDS DB instances should be configured with multiple Availability Zones
- [RDS.8] RDS DB instances should have deletion protection enabled
- [RDS.16] RDS DB clusters should be configured to copy tags to snapshots
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [Redshift.3] Amazon Redshift clusters should have automatic snapshots 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
- [S3.8] S3 general purpose buckets should block public access
- [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.2] Secrets Manager secrets configured with automatic rotation should rotate successfully
- [SNS.1] SNS topics should be encrypted at-rest using AWS KMS
- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [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
- [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

Israel (Tel Aviv)

The following controls are not supported in Israel (Tel Aviv).

- [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.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [Athena.1] Athena workgroups should be encrypted at rest
- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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.5] CodeBuild project environments should not have privileged mode 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
- [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
- [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
- [EC2.10] 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
- [EC2.14] 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
- [EC2.20] 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
- [EC2.51] EC2 Client VPN endpoints should have client connection logging enabled
- [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.9] ECS task definitions should have a logging configuration
- [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
- [EKS.8] EKS clusters should have audit logging enabled
- [ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.2] ElastiCache for Redis cache clusters should have auto minor version upgrade enabled
- [ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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.4] Application Load Balancer should be configured to drop 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.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
- [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
- [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
- [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.10] Password policies for IAM users should have strong AWS 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

- [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
- [Lambda.5] VPC Lambda functions should operate in multiple Availability Zones
- [Macie.1] Amazon Macie should be 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
- [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
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [PCA.1] AWS Private CA root certificate authority should be disabled
- [RDS.1] RDS snapshot should be private
- [RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest
- [RDS.7] RDS clusters should have deletion protection enabled
- [RDS.8] RDS DB instances should have deletion protection enabled
- [RDS.12] IAM authentication should be configured for RDS clusters
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.16] RDS DB clusters should be configured to copy tags to snapshots
- [RDS.24] RDS Database clusters 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
- [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
- [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
- [S3.8] 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
- [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
- [SNS.1] SNS topics should be encrypted at-rest using AWS KMS
- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [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
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [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.11] AWS WAF web ACL logging should be enabled
- [WAF.12] AWS WAF rules should have CloudWatch metrics enabled

Middle East (Bahrain)

The following controls are not supported in Middle East (Bahrain).

- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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
- [EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up
- [EC2.23] Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests
- [EC2.24] Amazon EC2 paravirtual instance types should not be used
- [ElasticBeanstalk.3] Elastic Beanstalk should stream logs to CloudWatch
- [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
- [GuardDuty.1] GuardDuty should be enabled
- [RDS.7] RDS clusters should have deletion protection enabled
- [RDS.12] IAM authentication should be configured for RDS clusters

- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.16] RDS DB clusters should be configured to copy tags to snapshots
- [RDS.24] RDS Database clusters should use a custom administrator username
- [Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [SSM.2] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation
- [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

Middle East (UAE)

The following controls are not supported in Middle East (UAE).

- [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.8] API Gateway routes should specify an authorization type
- [APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [Athena.1] Athena workgroups should be encrypted at rest
- [AutoScaling.1] Auto Scaling groups associated with a Classic Load Balancer should use load balancer health checks
- [Backup.1] AWS Backup recovery points should be encrypted at rest
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events
- [CloudTrail.6] Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible
- [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
- [CodeBuild.5] CodeBuild project environments should not have privileged mode 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
- [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
- [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
- [EC2.8] EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)
- [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
- [EC2.14] 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
- [EC2.23] 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
- [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.9] ECS task definitions should have a logging configuration
- [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.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.2] ElastiCache for Redis cache clusters should have auto minor version upgrade enabled

- [ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination
- [ELB.9] Classic Load Balancers should have cross-zone load balancing enabled
- [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
- [EMR.1] Amazon EMR cluster primary nodes should not have public IP addresses
- [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
- [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.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
- [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.4] AWS KMS key rotation should be enabled
- [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
- [MSK.1] MSK clusters should be encrypted in transit among broker nodes
- [MSK.2] MSK clusters should have enhanced monitoring configured
- [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
- [Opensearch.10] OpenSearch domains should have the latest software update installed
- [RDS.1] RDS snapshot should be private
- [RDS.2] RDS DB Instances should prohibit public access, as determined by the PubliclyAccessible AWS Configuration
- [RDS.3] RDS DB instances should have encryption at-rest enabled
- [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.11] RDS instances should have automatic backups enabled
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.16] RDS DB clusters should be configured to copy tags to snapshots
- [RDS.24] RDS Database clusters should use a custom administrator username
- [RDS.26] RDS DB instances should be protected by a backup plan
- [RDS.35] RDS DB clusters should have automatic minor version upgrade enabled
- [Redshift.9] Redshift clusters should not use the default database name
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [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

- [S3.14] S3 general purpose buckets should have versioning 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
- [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
- [SNS.1] SNS topics should be encrypted at-rest using AWS KMS
- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [SQS.1] Amazon SQS queues should be encrypted at rest
- [SSM.1] Amazon EC2 instances should be managed by AWS Systems Manager
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [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

South America (São Paulo)

The following controls are not supported in South America (São Paulo).

- [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
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [RDS.7] RDS clusters should have deletion protection enabled
- [RDS.12] IAM authentication should be configured for RDS clusters
- [RDS.14] Amazon Aurora clusters should have backtracking enabled
- [RDS.15] RDS DB clusters should be configured for multiple Availability Zones
- [RDS.16] RDS DB clusters should be configured to copy tags to snapshots
- [RDS.24] RDS Database clusters should use a custom administrator username
- [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

AWS GovCloud (US-East)

The following controls are not supported in AWS GovCloud (US-East).

- [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.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [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
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [Athena.1] Athena workgroups should be encrypted at rest
- [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.4] Auto Scaling group launch configuration should not have a metadata response hop limit greater than 1
- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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
- [CodeBuild.5] CodeBuild project environments should not have privileged mode 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.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.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [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
- [EC2.21] 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

- [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.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.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
- [EKS.8] EKS clusters should have audit logging enabled
- [ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.2] ElastiCache for Redis cache clusters should have auto minor version upgrade enabled
- [ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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.2] Classic Load Balancers with SSL/HTTPS listeners should use a certificate provided by AWS Certificate Manager

- [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
- [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
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [GuardDuty.1] GuardDuty 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
- [Kinesis.1] Kinesis streams should be encrypted at rest
- [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.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
- [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
- [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.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
- [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
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.1] S3 general purpose buckets should have block public access settings enabled
- [S3.8] S3 general purpose buckets should block public access
- [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
- [S3.20] 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
- [SecretsManager.3] Remove unused Secrets Manager secrets
- [SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days
- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [SSM.4] SSM documents should not be public
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [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

AWS GovCloud (US-West)

The following controls are not supported in AWS GovCloud (US-West).

- [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.2] RSA certificates managed by ACM should use a key length of at least 2,048 bits
- [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
- [AppSync.2] AWS AppSync should have field-level logging enabled
- [AppSync.5] AWS AppSync GraphQL APIs should not be authenticated with API keys
- [Athena.1] Athena workgroups should be encrypted at rest
- [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.4] Auto Scaling group launch configuration should not have a metadata response hop limit greater than 1
- [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
- [CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)
- [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
- [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
- [CodeBuild.5] CodeBuild project environments should not have privileged mode 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.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.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest
- [DynamoDB.4] DynamoDB tables should be present in a backup plan
- [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
- [EC2.21] 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
- [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.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.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
- [EKS.8] EKS clusters should have audit logging enabled
- [ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled
- [ElastiCache.2] ElastiCache for Redis cache clusters should have auto minor version upgrade enabled
- [ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled
- [ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest
- [ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit
- [ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH
- [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.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
- [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
- [FSx.1] FSx for OpenZFS file systems should be configured to copy tags to backups and volumes
- [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

- [Kinesis.1] Kinesis streams should be encrypted at rest
- [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.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
- [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
- [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.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
- [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
- [Route53.2] Route 53 public hosted zones should log DNS queries
- [S3.1] S3 general purpose buckets should have block public access settings enabled
- [S3.8] S3 general purpose buckets should block public access
- [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
- [S3.20] S3 general purpose buckets should have MFA delete enabled
- [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.3] Remove unused Secrets Manager secrets
- [SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days
- [SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic
- [SSM.4] SSM documents should not be public
- [StepFunctions.1] Step Functions state machines should have logging turned on
- [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

Disabling Security Hub

> **Note**
>
> If you use central configuration, the AWS Security Hub delegated administrator can create configuration policies that disable Security Hub in specific accounts and organizational units (OUs) and keep it enabled in others. Configuration policies take effect in your home Region and all linked Regions. For more information, see Central configuration in Security Hub.

You can use the Security Hub console, Security Hub API, or AWS CLI to disable Security Hub.

The following occurs when you disable Security Hub for an account:

- No new findings are process for the account.
- After 90 days, your existing findings and insights and any Security Hub configuration settings are deleted and cannot be recovered.

  If you want to save your existing findings, you must export them before you disable Security Hub. For more information, see the section called “Effect of account actions on Security Hub data”.
- Any enabled standards and controls are disabled.

You can't disable Security Hub in the following cases:

- Your account is the designated Security Hub administrator account for an organization. If you use central configuration, you can't associate a configuration policy that disables Security Hub with the delegated administrator account. The association can succeed for other accounts, but Security Hub doesn't apply such a policy to the delegated administrator account.
- Your account is a Security Hub administrator account by invitation, and you have member accounts that are enabled. Before you can disable Security Hub, you must disassociate all of your member accounts. See the section called “Disassociating member accounts”.

Before you can disable Security Hub for a member account, the account must be disassociated from its administrator account. For an organization account, only the administrator account

can disassociate member accounts. For more information, see the section called "Disassociating organization member accounts". For manually invited accounts, either the administrator account or the member account can disassociate the member account. For more information, see the section called "Disassociating member accounts" or the section called "Disassociating from your administrator account". Disassociation isn't required if you use central configuration because you can create a policy that disables Security Hub in specific member accounts.

When you disable Security Hub in an account, it is disabled only in the current Region. However, if you use central configuration to disable Security Hub in specific accounts, it is disabled in the home Region and all linked Regions.

Choose your preferred method, and follow the steps to disable Security Hub.

Security Hub console

To disable Security Hub

1. Open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. On the navigation pane, choose **Settings**.
3. On the **Settings** page, choose **General**.
4. Under **Disable AWS Security Hub**, choose **Disable AWS Security Hub**. Then choose **Disable AWS Security Hub** again.

Security Hub API

To disable Security Hub

Invoke the `DisableSecurityHub` API.

AWS CLI

To disable Security Hub

Run the `disable-security-hub` command.

Example command:

```
aws securityhub disable-security-hub
```

Change log for Security Hub controls

The following change log tracks material changes to AWS Security Hub security controls, which may result in changes to the overall status of a control and the compliance status of its findings. For information about how Security Hub evaluates control status, see Compliance status and control status. Changes may 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.

Select a control to view more details about it. Title changes are noted on each control's detailed description for 90 days.

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

Date of change	Control ID and title	Description of change
		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.
April 10, 2024	**[Athena.1] Athena workgroups should be encrypted at rest**	Security Hub 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.

Date of change	Control ID and title	Description of change
April 10, 2024	**[AutoScaling.4] Auto Scaling group launch configuration should not have a metadata response hop limit greater than 1**	Security Hub retired this control and removed it from all standards. Metadata response hop limits for Amazon Elastic Compute Cloud (Amazon EC2) instances are workload dependent.
April 10, 2024	**[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)**	Security Hub 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.

Date of change	Control ID and title	Description of change
April 10, 2024	**[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled**	Security Hub retired this control and removed it from all standards. Enabling privileged mode in a CodeBuild project does not impose an additional risk to the customer environment.
April 10, 2024	**[IAM.20] Avoid the use of the root user**	Security Hub 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.

Date of change	Control ID and title	Description of change
April 10, 2024	**[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic**	Security Hub 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 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 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 removed this control from AWS Foundational Security Best Practices and Service-Managed Standard: AWS Control Tower. Since SNS already encrypts topics by default, 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 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 changed the title to account for a new S3 bucket type.
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 changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.5] 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 changed the title to account for a new S3 bucket type.
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 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 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 changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.8] S3 general purpose buckets should block public access	Changed title from **S3 Block Public Access setting should be enabled at the bucket-level** to **S3 general purpose buckets should block public access**. Security Hub changed the title to account for a new S3 bucket type.
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 changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.10] S3 general purpose buckets with versioning enabled should have Lifecycle configurations	Changed title from **S3 buckets with versioning enabled should have lifecycle policies configured** to **S3 general purpose buckets with versioning enabled should have Lifecycle configurations**. Security Hub changed the title to account for a new S3 bucket type.
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 changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.12] 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 changed the title to account for a new S3 bucket type.
March 12, 2024	[S3.13] S3 general purpose buckets should have Lifecycle configurations	Changed title from **S3 buckets should have lifecycle policies configure d** to **S3 general purpose buckets should have Lifecycle configura tions**. Security Hub changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.14] 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 changed the title to account for a new S3 bucket type.
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 changed the title to account for a new S3 bucket type.

Date of change	Control ID and title	Description of change
March 12, 2024	[S3.17] 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 changed the title to account for a new S3 bucket type.
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 now supports `nodejs20.x` and `ruby3.3` as a parameter.

Date of change	Control ID and title	Description of change
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 now supports `dotnet8` as a parameter.
February 5, 2024	[EKS.2] EKS clusters should run on a supported Kubernetes version	Security Hub 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 removed mention of OAuth because other connection methods can also be secure. Security Hub 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 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 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 now supports `java21` and `python3.12` as parameters. Security Hub 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 distribution has a default root object configured. Security Hub 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 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
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 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'.
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**.

Date of change	Control ID and title	Description of change
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**.
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**.

Date of change	Control ID and title	Description of change
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**.
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 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 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 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 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 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 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 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 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 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 retired this control and removed it from all standards. Amazon EC2 has migrated EC2-Classic 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 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 encryption 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 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 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-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.The oldest supported version is now `1.23`.
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 now supports `ruby3.2` as a parameter.

Date of change	Control ID and title	Description of change
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 updated the control to evaluate the encryption of a particular method only when caching is enabled for that method.
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 now supports `java17` as a parameter.

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 no longer supports `nodejs12.x` as a parameter.
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 updated this control to produce **Passed** findings when you use CodeDeploy to deploy ECS Fargate services.

Date of change	Control ID and title	Description of change
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 updated the control to account for conditionals in a bucket policy.
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 now supports `python3.10` as a parameter.

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 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 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.

Document history for the AWS Security Hub User Guide

The following table describes the updates to the documentation for AWS Security Hub.

> **Note**
>
> For security control releases, the date specified is the date when the controls are available in all accounts and Regions. It can take 1-2 weeks for controls to reach all accounts and Regions.

Change	Description	Date
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 console.	March 29, 2024
Update to existing managed policy	Security Hub 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 available in Canada West (Calgary)	Security Hub is now available in Canada West (Calgary). All Security Hub features are	December 20, 2023

now available in this Region, with the exception of certain security controls. For more information, see Availability of controls by Region.

New security controls	The following new Security Hub 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 enhanced monitoring configured"	December 14, 2023

- 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 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 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 delegated administrator can configure Security Hub, standards, and controls across multiple organization accounts, organizational units (OUs), and Regions.	November 27, 2023
Updates to managed policy	Security Hub added new permissions to the `AWSSecurityHubServiceRolePolicy` managed policy that allow Security Hub to read and update customizable security control properties.	November 26, 2023

Custom control parameters	You can now customize parameter values for select Security Hub 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 updated the `AWSSecurityHubFullAccess` and `AWSSecurityHubOrganizationsAccess` managed policies that permit you to use, respectively, Security Hub 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 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 added a new tagging permission to the `AWSSecurityHubServiceRolePolicy` managed policy that allows Security Hub to read resource tags related to findings.	November 7, 2023

New security controls	The following new Security Hub 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 added new Organizations actions to the `AWSSecurityHubServiceRolePolicy` managed policy that allow Security Hub to retrieve account and organizational unit (OU) information. We also added new Security Hub actions that allow Security Hub 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 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 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 console shows all your controls across standards. Each control has the same control ID across standards. When you turn on consolida ted control findings, you receive a single finding per security check even when a control applies to multiple enabled standards.	August 28, 2023

Security Hub available in Israel (Tel Aviv) Region	Security Hub is now available in Israel (Tel Aviv). All Security Hub features are now available in this Region, with the exception of certain security controls. For more information, see Availability of controls by Region.	August 8, 2023

New security controls	The following new Security Hub 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 deletion protection enabled"	July 28, 2023

	• 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 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.	June 12, 2023

Existing security controls added to Service-Managed Standard: AWS Control Tower	The following existing Security Hub 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 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 available in Asia Pacific (Melbourne)	Security Hub is now available in Asia Pacific (Melbourne). All Security Hub 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 can now track the history of a finding during the last 90 days.	May 4, 2023
New security controls	The following new Security Hub 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 available in new AWS Regions	Security Hub 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 has updated an existing permission in the `AWSSecurityHubServiceRolePolicy` managed policy.	March 17, 2023

New security controls for NIST 800-53 standard	Security Hub 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 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 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 controls are available. Some controls have Regional limitations. • the section called "[ElastiCache.1] ElastiCache Redis clusters should have automatic backup enabled" • the section called "[ElastiCache.2] ElastiCache for Redis cache clusters should have auto minor version upgrade enabled" • the section called "[ElastiCache.3] ElastiCache for Redis replication groups should have automatic failover enabled" • the section called "[ElastiCache.4] ElastiCache for Redis replication groups should be encrypted at rest" • the section called "[ElastiCache.5] ElastiCache for Redis replication groups should be encrypted in transit" • the section called "[ElastiCache.6] ElastiCache for Redis replication groups before version 6.0 should use Redis AUTH"	February 16, 2023

	• the section called "[ElastiCache.7] ElastiCache clusters should not use the default subnet group"	
New ASFF fields	Security Hub 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 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 console for findings that Amazon Inspector sends to Security Hub.	January 14, 2023
Security Hub is available in Middle East (UAE)	Security Hub is now available in Middle East (UAE). Some controls have Regional limits.	January 12, 2023
Added third-party integration with MetricStream	Security Hub 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 now supports up to 11,000 member accounts for each Security Hub administrator account per Region.	December 27, 2022
ElasticBeanstalk.3 rolled back	Security Hub 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 adds new security controls	New Security Hub 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 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 by receiving Security Hub findings.	November 29, 2022

Support for Service-Managed Standard: AWS Control Tower	Security Hub 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 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 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 in all available Regions.	November 17, 2022
Support for CIS AWS Foundations Benchmark v1.4.0	Security Hub 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 announcements in AWS GovCloud (US)	You can now subscribe to Security Hub announcements with Amazon Simple Notification Service (Amazon SNS) in AWS GovCloud (US-East) and AWS GovCloud (US-West) to receive notifications about Security Hub.	October 3, 2022
AWS Security Hub adds a new security control	The new Security Hub 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 announcements	You can now subscribe to Security Hub announcements with Amazon Simple Notification Service (Amazon SNS) to receive notifications about Security Hub.	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 findings, and JFrog is a third-party integration that sends findings to Security Hub.	July 26, 2022

EC2.27 is retired	Security Hub 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 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 adds new security controls	New Security Hub controls are available to customers who have enabled the FSBP standard. Some controls have Regional limitations.	June 22, 2022
AWS Security Hub supports a new Region	Security Hub is now available in Asia Pacific (Jakarta). Some controls are not available in this Region.	June 7, 2022
Improved integration between AWS Security Hub and AWS Config	Security Hub users can see the results of AWS Config rule evaluations as findings in Security Hub.	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 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.	January 19, 2022
Added integration with AWS Trusted Advisor	Trusted Advisor sends the results of its checks to Security Hub as Security Hub findings. Security Hub 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 is integrated with the new version of Amazon Inspector as well as with Amazon Inspector Classic. Amazon Inspector sends findings to Security Hub.	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 now offers an integration with Sonrai Dig. Sonrai Dig monitors cloud environments to identify security risks. Sonrai Dig sends findings to Security Hub.	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 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.	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 integrati on 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-block s` to `s3-account-level-public-access-block s-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 now offers an integration with HackerOne Vulnerability Intelligence. The integration sends findings to Security Hub.	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 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 is retiring the control **[Lambda.4] Lambda functions should have a dead-letter queue configure d**. When a control is retired, it no longer displays on the console, and Security Hub does not perform checks against it.	August 31, 2021
Retiring the PCI.EC2.3 control	Security Hub 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 does not perform checks against it.	August 27, 2021
Change to how Security Hub sends findings to custom actions	When you send findings to a custom action, Security Hub 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_CU STOM_RUNTIME_DETAI LS_NOT_AVAILABLE` compliance status reason code. This reason code indicates that Security Hub 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 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 now offers integrations with Caveonix Cloud and Forcepoint Cloud Security Gateway. Both integrations send findings to Security Hub.	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 AWS Chatbot	Added the integration with AWS Chatbot. Security Hub sends findings to AWS Chatbot.	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 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 handling of suspended accounts	Added information on how Security Hub 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. NETSCOUT Cyber Investigator sends findings to Security Hub.	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 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 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.	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 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 now supports an integration with Systems Manager Explorer and OpsCenter. The integrati on receives findings from Security Hub and updates those findings in Security Hub.	April 26, 2021
New type for product integrations	A new integration type, `UPDATE_FINDINGS_IN _SECURITY_HUB` , indicates that a product integrati on updates findings that it receives from Security Hub.	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 console and API.	April 22, 2021
Updated APIGateway.1 to replace HTTP with Websocket	Updated the title, descripti on, 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 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 launched in Asia Pacific (Osaka)	Security Hub 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.	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 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.	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 as findings.	January 27, 2021

Updated the security score calculation process for administrator accounts	For an administrator account, Security Hub 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 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 finding is received. Updates to these incidents or problems result in updates to the findings in Security Hub.	December 11, 2020
New integration with AWS Audit Manager	Security Hub now offers an integration with AWS Audit Manager. The integration allows Audit Manager to receive control-based findings from Security Hub.	December 8, 2020
New integration with Aqua Security Kube-bench	Security Hub added an integration with Aqua Security Kube-bench. The integration sends findings to Security Hub.	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 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 administrator account, who determines which organization accounts to enable in Security Hub. 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 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 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 launched in China (Beijing) and China (Ningxia)	Security Hub 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 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. VM-Series receives findings from Security Hub.	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. Patch Manager sends findings to Security Hub 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 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 is now integrated with RSA Archer. RSA Archer receives findings from Security Hub.	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 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 calculates the security score for a standard	When calculating the security score for a standard, Security Hub 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 automatically sets `Workflow.Status` to `RESOLVED`.	June 24, 2020

AWS Command Line Interface examples	Added AWS CLI syntax and examples for several Security Hub tasks. Includes enabling Security Hub, managing insights, managing standards and controls, managing product integrations, and disabling Security Hub.	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. 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, Forcepoin t 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 security standard for the Payment Card Industry Data Security Standard (PCI DSS). When this standard is enabled, Security Hub 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 Terminology and Concepts	Updated some descriptions and added new terms to Terminology and Concepts.	September 21, 2019

AWS Security Hub general availability release	Content updates to reflect improvements made to Security Hub during the preview period.	June 25, 2019
Added remediation steps for CIS AWS Foundations checks	Added remediation steps to Security Standards Supported in AWS Security Hub.	April 15, 2019
Preview release of AWS Security Hub	Published the preview release version of the *AWS Security Hub User Guide*.	November 18, 2018