AWS Technical Guide

AWS Security Incident Response Guide

AWS Security Incident Response Guide: AWS Technical Guide

Table of Contents

AWS Security Incident Response Guide

Publication date: **January 1, 2023** (*Document revisions*)

Abstract

This guide presents an overview of the fundamentals of responding to security incidents within a customer's Amazon Web Services (AWS) Cloud environment. It provides an overview of cloud security and incident response concepts and identifies cloud capabilities, services, and mechanisms that are available to customers who respond to security issues.

This paper is intended for those in technical roles and assumes that you are familiar with the general principles of information security, have a basic understanding of security incident response in your current on-premises environments, and have some familiarity with cloud services.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console, you can review your workloads against these best practices by answering a set of questions for each pillar.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

Security is the top priority at AWS. AWS customers benefit from data centers and network architecture built to help support the needs of the most security-sensitive organizations. AWS has a shared responsibility model: AWS manages the security *of* the cloud, and customers are responsible for security *in* the cloud. This means that you have full control of your security implementation, including access to several tools and services to help meet your security objectives. These capabilities help you establish a security baseline for applications running in the AWS Cloud.

When a deviation from the baseline occurs, such as by a misconfiguration or changing external factors, you will need to respond and investigate. To successfully do so, you need to understand the basic concepts of security incident response within your AWS environment and the requirements to prepare, educate, and train cloud teams before security issues occur. It is important to know which controls and capabilities you can use, review topical examples for resolving potential concerns, and identify remediation methods that use automation to improve response speed and consistency. Additionally, you should understand your compliance and regulatory requirements as they relate to building a security incident response program to fulfill those requirements.

Security incident response can be complex, so we encourage you to implement an iterative approach: begin with the core security services, build foundational detection and response capabilities, then develop playbooks to create an initial library of incident response mechanisms upon which to iterate and improve.

Before you begin

Before you begin learning about security incident response in AWS, familiarize yourself with the relevant standards and frameworks for AWS security and incident response. These foundations will help you understand the concepts and best practices presented in this guide.

AWS security standards and frameworks

To start, we encourage you to review the Best Practices for Security, Identity, and Compliance, Security Pillar - AWS Well-Architected Framework and the Security Perspective of the Overview of the AWS Cloud Adoption Framework (AWS CAF) whitepaper.

The AWS CAF provides guidance supporting coordination between different parts of organizations moving to the cloud. The AWS CAF guidance is divided into several focus areas, referred to as

perspectives, that are relevant to building cloud-based IT systems. The security perspective describes how to implement a security program across workstreams, one of which is incident response. This document is a product of our experiences working with customers to help them build effective and efficient security incident response programs and capabilities.

Industry incident response standards and frameworks

This whitepaper follows the incident response standards and best practices from the Computer Security Incident Handling Guide SP 800-61 r2, which was created by the National Institute of Standards and Technology (NIST). Reading and understanding the concepts introduced by NIST is a helpful pre-requisite. Concepts and best practices from this NIST guide will be applied to AWS technologies in this paper. However, on-premises incident scenarios are out of scope for this guide.

AWS incident response overview

To start, it's important to understand how security operations and incident response are different in the cloud. To build response capabilities that are effective in AWS, you will need to understand the deviations from traditional on-premises response and their impact to your incident response program. Each of these differences, as well as core AWS incident response design principles, are detailed in this section.

Aspects of AWS incident response

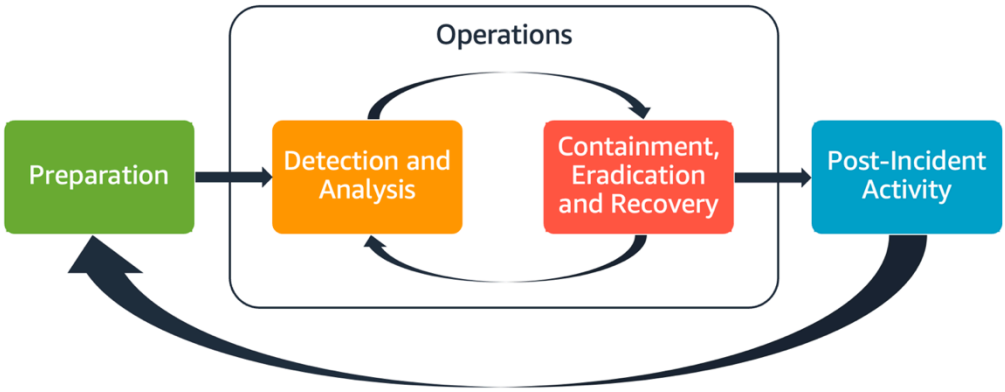
All AWS users within an organization should have a basic understanding of security incident response processes, and security staff should understand how to respond to security issues. Education, training, and experience are vital to a successful cloud incident response program and are ideally implemented well in advance of having to handle a possible security incident. The foundation of a successful incident response program in the cloud is *Preparation*, *Operations*, and *Post-Incident Activity*.

To understand each of these aspects, consider the following descriptions:

- **Preparation** – Prepare your incident response team to detect and respond to incidents within AWS by enabling detective controls and verifying appropriate access to the necessary tools and cloud services. Additionally, prepare the necessary playbooks, both manual and automated, to verify reliable and consistent responses.
- **Operations** – Operate on security events and potential incidents following NIST's phases of incident response: detect, analyze, contain, eradicate, and recover.

- **Post-incident activity** – Iterate on the outcome of your security events and simulations to improve the efficacy of your response, increase value derived from response and investigation, and further reduce risk. You have to learn from incidents and have strong ownership of improvement activities.

Each of these aspects are explored and detailed in this guide. The following diagram shows the flow of these aspects, aligning with the previously mentioned NIST incident response lifecycle, but with operations encompassing detection and analysis with containment, eradication, and recovery.



Aspects of AWS incident response

AWS incident response principles and design goals

While the general processes and mechanisms of incident response as defined by the NIST SP 800-61 Computer Security Incident Handling Guide are sound, we encourage you to also consider these specific design goals that are relevant to responding to security incidents in a cloud environment:

- **Establish response objectives** – Work with stakeholders, legal counsel, and organizational leadership to determine the goal of responding to an incident. Some common goals include containing and mitigating the issue, recovering the affected resources, preserving data for forensics, returning to known safe operations, and ultimately learning from incidents.
- **Respond using the cloud** – Implement response patterns within the cloud, where the event and data occurs.
- **Know what you have and what you need** – Preserve logs, resources, snapshots, and other evidence by copying and storing them in a centralized cloud account dedicated to response. Use tags, metadata, and mechanisms that enforce retention policies. You'll need to understand what

services you use and then identify requirements for investigating those services. To help you understand your environment, you can also use tagging, which is covered later in this document in the the section called "Develop and implement a tagging strategy" section.

- **Use redeployment mechanisms** – If a security anomaly can be attributed to a misconfiguration, the remediation might be as simple as removing the variance by redeploying resources with the proper configuration. If a possible compromise is identified, verify that your redeployment includes successful and verified mitigation of the root causes.
- **Automate where possible** – As issues arise or incidents repeat, build mechanisms to programmatically triage and respond to common events. Use human responses for unique, complex, or sensitive incidents where automations are insufficient.
- **Choose scalable solutions** – Strive to match the scalability of your organization's approach to cloud computing. Implement detection and response mechanisms that scale across your environments to effectively reduce the time between detection and response.
- **Learn and improve your process** – Be proactive in identifying gaps in your processes, tools, or people, and implement a plan to fix them. Simulations are safe methods to find gaps and improve processes. Refer to the *Post-incident activity* section of this document for details on how to iterate on your processes.

These design goals are a reminder to review your architecture implementation for the ability to conduct both incident response and threat detection. As you plan your cloud implementations, think about responding to an incident, ideally with forensically sound response methodology. In some cases, this means you might have multiple organizations, accounts, and tools specifically set up for these response tasks. These tools and functions should be made available to the incident responder by deployment pipeline. They should not be static because it can cause a larger risk.

Cloud security incident domains

To effectively prepare for and respond to security events in your AWS environment, you need to understand the commons types of cloud security incidents. There are three domains within the customer's responsibility where security incidents might occur: service, infrastructure, and application. Different domains require different knowledge, tools, and response processes. Consider these domains:

- **Service domain** – Incidents in the service domain might affect your AWS account, AWS Identity and Access Management (IAM) permissions, resource metadata, billing, or other areas. A service domain event is one that you respond to exclusively with AWS API mechanisms, or where you

have root causes associated with your configuration or resource permissions, and might have related service-oriented logging.

- **Infrastructure domain** – Incidents in the infrastructure domain include data or network-related activity, such as processes and data on your Amazon Elastic Compute Cloud (Amazon EC2) instances, traffic to your Amazon EC2 instances within the virtual private cloud (VPC), and other areas, such as containers or other future services. Your response to infrastructure domain events often involves acquiring incident-related data for forensic analysis. It likely includes interaction with the operating system of an instance, and, in various cases, might also involve AWS API mechanisms. In the infrastructure domain, you can use a combination of AWS APIs and digital forensics/incident response (DFIR) tooling within a guest operating system, such as an Amazon EC2 instance dedicated to performing forensic analysis and investigations. Infrastructure domain incidents might involve analyzing network packet captures, disk blocks on an Amazon Elastic Block Store (Amazon EBS) volume, or volatile memory acquired from an instance.
- **Application domain** – Incidents in the application domain occur in the application code or in software deployed to the services or infrastructure. This domain should be included in your cloud threat detection and response playbooks and might incorporate similar responses to those in the infrastructure domain. With appropriate and thoughtful application architecture, you can manage this domain with cloud tools by using automated acquisition, recovery, and deployment.

In these domains, consider the actors who might act against AWS accounts, resources, or data. Whether internal or external, use a risk framework to determine specific risks to the organization and prepare accordingly. Additionally, you should develop threat models, which can help with your incident response planning and thoughtful architecture building.

Key differences of incident response in AWS

Incident response is an integral part of a cyber security strategy either on-premises or in the cloud. Security principles such as least-privilege and defense-in-depth intend to protect the confidentiality, integrity, and availability of data both on-premises and in the cloud. Several incident response patterns that support these security principles follow suit, including log retention, alert selection derived from threat modeling, playbook development, and security information and event management (SIEM) integration. The differences begin when customers start architecting and engineering these patterns in the cloud. The following are the key differences of incident response in AWS.

Difference #1: Security as a shared responsibility

The responsibility for security and compliance is shared between AWS and its customers. This shared responsibility model relieves some of the customer's operational burden because AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates. For more details on the shared responsibility model, refer to the Shared Responsibility Model documentation.

As your shared responsibility in the cloud changes, your options for incident response also change. Planning for and understanding these tradeoffs and matching them with your governance needs is a crucial step in incident response.

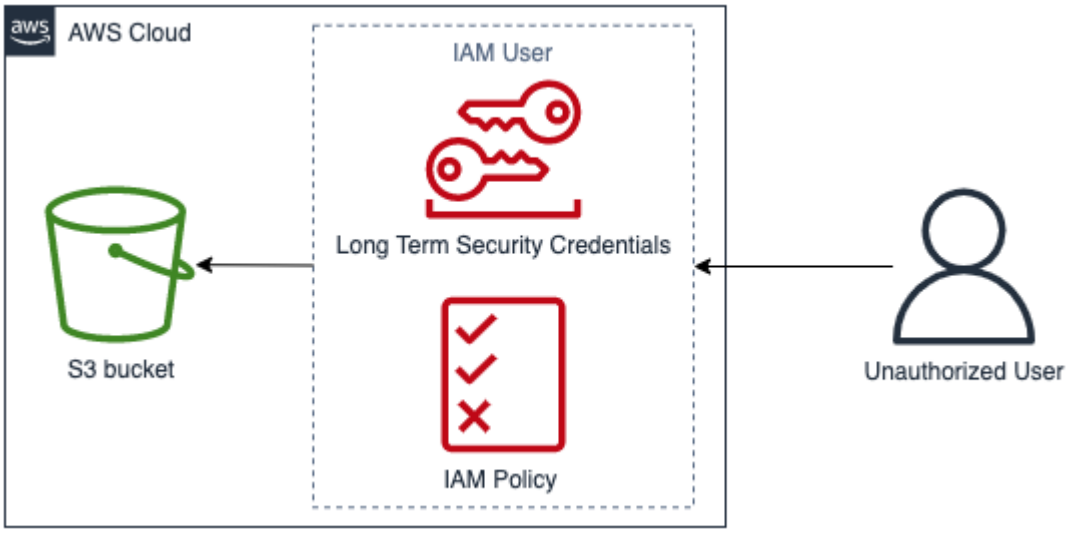
In addition to the direct relationship you have with AWS, there might be other entities that have responsibilities in your particular responsibility model. For example, you might have internal organizational units that take responsibility for some aspects of your operations. You might also have partners or other parties that develop, manage, or operate some of your cloud technology.

Creating and testing an appropriate incident response plan and appropriate playbooks that match your operating model is extremely important.

Difference #2: Cloud service domain

Because of the differences in security responsibility that exist in cloud services, a new domain for security incidents was introduced: the service domain, which was explained earlier in the Incident domain section. The service domain encompasses a customer's AWS account, IAM permissions, resource metadata, billing, and other areas. This domain is different for incident response because of how you respond. Response within the service domain is typically done by reviewing and issuing API calls, rather than traditional host-based and network-based response. In the service domain, you won't interact with an affected resource's operating system.

The following diagram shows an example of a security event in the service domain based on an architectural anti-pattern. In this event, an unauthorized user obtains the long-term security credentials of an IAM user. The IAM user has an IAM policy that allows them to retrieve objects from an Amazon Simple Storage Service (Amazon S3) bucket. To respond to this security event, you would use AWS APIs to analyze AWS logs such as AWS CloudTrail and Amazon S3 access logs. You would also use AWS APIs to contain and recover from the incident.



Service domain example

Difference #3: APIs for provisioning infrastructure

Another difference comes from the Cloud characteristic of on-demand self-service. The main facility customers interact with the AWS Cloud by using a RESTful API through public and private endpoints available in many geographical locations around the globe. Customers can access these APIs with AWS credentials. In contrast to on-premises access control, these credentials are not necessarily bound by a network or a Microsoft Active Directory domain. Credentials are instead associated with an IAM principal inside of an AWS account. These API endpoints can be accessed outside of your corporate network, which will be important to understand when you respond to an incident where credentials are used outside of your expected network or geography.

Because of the API-based nature of AWS, an important log source for responding to security events is AWS CloudTrail, which tracks the management API calls made in your AWS accounts and where you can find information about the source location of the API calls.

Difference #4: Dynamic nature of the cloud

The cloud is dynamic; it allows you to quickly create and delete resources. With autoscaling, resources can be spun up and spun down based on increases in traffic. With short-lived infrastructure and fast-paced changes, a resource that you're investigating might no longer exist or might have been modified. Understanding the ephemeral nature of AWS resources and how you can track the creation and deletion of AWS resources will be important for incident analysis. You can use AWS Config to track the configuration history of your AWS resources.

Difference #5: Data access

Data access is also different in the cloud. You can't plug into a server in order to collect the data you need for a security investigation. Data is collected over the wire and through API calls. You'll need to practice and understand how to perform data collection over APIs in order to be prepared for this shift, and verify appropriate storage for effective collection and access.

Difference #6: Importance of automation

For customers to fully realize the benefits of cloud adoption, their operational strategy must embrace automation. Infrastructure-as-code (IaC) is a pattern of highly efficient automated environments where AWS services are deployed, configured, re-configured, and destroyed using code facilitated by native IaC services such as AWS CloudFormation or third-party solutions. This pushes the implementation of incident response to be highly automated, which is desirable to avoid human mistakes, especially when handling evidence. While automation is used on-premises, it is essential and simpler in the AWS Cloud.

Addressing these differences

To address these differences, follow the steps outlined in the next section to verify that your incident response program across people, processes, and technology is well prepared.

Preparation

Preparing for an incident is critical for timely and effective incident response. Preparation is done across three domains:

- **People** – Preparing your people for a security incident involves identifying the relevant stakeholders for incident response and training them on incident response and cloud technologies.
- **Process** – Preparing your processes for a security incident involves documenting architectures, developing thorough incident response plans, and creating playbooks for consistent response to security events.
- **Technology** – Preparing your technology for a security incident involves setting up access, aggregating and monitoring necessary logs, implementing effective alerting mechanisms, and developing response and investigative capabilities.

Each of these domains are equally important for effective incident response. No incident response program is complete or effective without all three. You will need to prepare people, processes, and technologies with tight integration in order to be prepared for an incident.

People

To respond to a security event, you need to identify the stakeholders who would support the response to a security event. Additionally, it is critical for an effective response to have them trained on AWS technologies and your AWS environment.

Define roles and responsibilities

Handling security events requires cross-organizational discipline and an inclination for action. Within your organizational structure, there should be many people who are responsible, accountable, consulted, or kept informed during an incident, such as representatives from human resources (HR), the executive team, and legal. Consider these roles and responsibilities, and whether any third parties must be involved. Note that in many geographies, there are local laws that govern what should and should not be done. Although it might seem bureaucratic to build a responsible, accountable, consulted, and informed (RACI) chart for your security response plans, doing so enables quick and direct communication and clearly outlines the leadership across different stages of the event.

During an incident, including the owners/developers of impacted applications and resources is key because they are subject matter experts (SMEs) that can provide information and context to aid in measuring impact. Make sure to practice and build relationships with the developers and application owners before you rely on their expertise for incident response. Application owners or SMEs, such as your cloud administrators or engineers, might need to act in situations where the environment is unfamiliar or has complexity, or where the responders don't have access.

Lastly, trusted partners might be involved in the investigation or response because they can provide additional expertise and valuable scrutiny. When you don't have these skills on your own team, you might want to hire an external party for assistance.

Train incident response staff

Training your incident response staff on the technologies their organization uses will be crucial for them to adequately respond to a security event. Responses might be prolonged if your staff members don't understand the underlying technologies. In addition to traditional incident response concepts, it's also important that they understand AWS services and their AWS environment. There are a number of traditional mechanisms to train your incident staff, such as online training and classroom training. You should also consider running gamedays or simulations as a mechanism for training. For details on how to run simulations, refer the the section called "Run regular simulations" section of this document.

Understand AWS Cloud technologies

To reduce dependencies and decrease response time, ensure that your security teams and responders are educated about cloud services and have opportunities for hands-on practice with the specific cloud environment that your organization uses. For incident responders to be effective, it's important to understand AWS foundations, IAM, AWS Organizations, AWS logging and monitoring services, and AWS security services.

AWS provides online security workshops (refer to AWS Security Workshops) where you can get hands-on experiences with AWS security and monitoring services. AWS also provides a number of training options and learning paths through digital training, classroom training, APN partners, and certifications. To learn more, refer to AWS Training and Certification.

Understand your AWS environment

In addition to understanding AWS services, their use cases, and how they integrate with each other, it's equally important to understand how your organization's AWS environment is actually

architected and what operational processes are in place. Often, internal knowledge such as this is not documented and is understood by only a few domain experts, which can create dependencies, hinder innovation, and slow response time.

To avoid these dependencies and quicken response times, internal knowledge of your AWS environment should be documented, accessible, and understood by your security analysts. Understanding your complete cloud footprint will require collaboration between relevant security stakeholders and cloud administrators. Part of preparing your processes for incident response includes documenting and centralizing architecture diagrams, which is the section called "Document and centralize architecture diagrams" later in this whitepaper. However, from a people perspective, it's important that your analysts can access and understand the diagrams and operational processes related to your AWS environment.

Understand AWS response teams and support

AWS Support

AWS Support offers a range of plans that provide access to tools and expertise that support the success and operational health of your AWS solutions. If you need technical support and more resources to help plan, deploy, and optimize your AWS environment, you can select a support plan that best aligns with your AWS use case.

Consider the Support Center in the AWS Management Console (sign-in required) as the central point of contact to get support for issues that affect your AWS resources. Access to AWS Support is controlled by IAM. For more information about getting access to AWS Support features, refer to Getting started with AWS Support.

Additionally, if you need to report abuse, contact the AWS abuse team.

AWS Customer Incident Response Team (CIRT)

The AWS Customer Incident Response Team (CIRT) is a specialized 24/7 global AWS team that provides support to customers during active security events on the customer side of the AWS Shared Responsibility Model.

When the AWS CIRT supports you, you will receive assistance with triage and recovery for an active security event on AWS. They will assist in root cause analysis through the use of AWS service logs and provide you with recommendations for recovery. They will also provide security recommendations and best practices to help you avoid security events in the future.

AWS customers can engage the AWS CIRT through an AWS support case.

DDoS response support

AWS offers AWS Shield, which provides a managed distributed denial of service (DDoS) protection service that safeguards web applications running on AWS. AWS Shield provides always-on detection and automatic inline mitigations that can minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection. There are two tiers of AWS Shield: Shield Standard and Shield Advanced. To learn about the differences between these two tiers, refer to the Shield features documentation.

AWS Managed Services (AMS)

AWS Managed Services (AMS) provides ongoing management of your AWS infrastructure so you can focus on your applications. By implementing best practices to maintain your infrastructure, AMS helps reduce your operational overhead and risk. AMS automates common activities such as change requests, monitoring, patch management, security, and backup services, and provides full-lifecycle services to provision, run, and support your infrastructure.

AMS takes responsibility for deploying a suite of security detective controls and provides a 24/7 first line of response to alerts. When an alert is initiated, AMS follows a standard set of automated and manual playbooks to verify a consistent response. These playbooks are shared with AMS customers during onboarding so that they can develop and coordinate a response with AMS.

Process

Developing thorough and clearly defined incident response processes is key to a successful and scalable incident response program. When a security event occurs, clear steps and workflows will help you to respond in a timely manner. You might already have existing incident response processes. Regardless of your current state, it's important to update, iterate, and test your incident response processes regularly.

Develop and test an incident response plan

The first document to develop for incident response is the *incident response plan*. The incident response plan is designed to be the foundation for your incident response program and strategy. An incident response plan is a high-level document that typically includes these sections:

- **An incident response team overview** – Outlines the goals and functions of the incident response team

- **Roles and responsibilities** – Lists the incident response stakeholders and details their roles when an incident occurs
- **A communication plan** – Details contact information and how you will communicate during an incident

  It's a best practice to have out-of-band communication as a backup for incident communication. An example of an application that provides a secure out-of-band communications channel is AWS Wickr.
- **Phases of incident response and actions to take** – Enumerates the phases of incident response – for example, detect, analyze, eradicate, contain and recover – including high-level actions to take within those phases
- **Incident severity and prioritization definitions** – Details how to classify the severity of an incident, how to prioritize the incident, and then how the severity definitions affect escalation procedures

While these sections are common throughout companies of different sizes and industries, each organization's incident response plan is unique. You will need to build an incident response plan that works best for your organization.

Document and centralize architecture diagrams

To quickly and accurately respond to a security event, you need to understand how your systems and networks are architected. Understanding these internal patterns is not only important for incident response, but also for verifying consistency across applications that the patterns are architected with, according to best practices. You should also verify that this documentation is up to date and regularly updated in accordance with new architecture patterns. You should develop documentation and internal repositories that detail items such as:

- **AWS account structure** - You need to know:
  - How many AWS accounts do you have?
  - How are those AWS accounts organized?
  - Who are the business owners of the AWS accounts?
  - Do you use Service Control Policies (SCPs)? If so, what organizational guardrails are implemented by using SCPs?
  - Do you limit the regions and services that can be used?
  - What differences are there between business units and environments (dev/test/prod)?

- **AWS service patterns**
  - What AWS services do you use?
  - What are the most widely used AWS services?
- **Architecture patterns**
  - What cloud architectures do you use?
- **AWS authentication patterns**
  - How do your developers typically authenticate to AWS?
  - Do you use IAM roles or users (or both)? Is your authentication to AWS connected to an identity provider (IdP)?
  - How do you map an IAM role or user to an employee or system?
  - How does access get revoked when someone is no longer authorized?
- **AWS authorization patterns**
  - What IAM policies do your developers use?
  - Do you use resource-based policies?
- **Logging and monitoring**
  - What logging sources do you use and where are they stored?
  - Do you aggregate AWS CloudTrail logs? If so, where are they stored?
  - How do you query CloudTrail logs?
  - Do you have Amazon GuardDuty enabled?
  - How do you access GuardDuty findings (for example, console, ticketing system, SIEM)?
  - Are findings or events aggregated in a SIEM?
  - Are tickets automatically created?
  - What tooling is in place to analyze logs for an investigation?
- **Network topology**
  - How are devices, endpoints, and connections on your network physically or logically arranged?
  - How does your network connect with AWS?
  - How is network traffic filtered between environments?
- **External infrastructure**
  - How are externally facing application deployed?
  - What AWS resources are publicly accessible?
  - What AWS accounts contain infrastructure that is externally facing?

- What DDoS or external filtering is there?

Documenting internal technical diagrams and processes eases the incident response analyst's job, helping them quickly obtain the institutional knowledge to respond to a security event. Thorough documentation of internal technical processes not only simplifies security investigations, but also adjusts for rationalization and evaluation of the processes.

Develop incident response playbooks

A key part of preparing your incident response processes is developing playbooks. Incident response playbooks provide a series of prescriptive guidance and steps to follow when a security event occurs. Having clear structure and steps simplifies the response and reduces the likelihood for human error.

What to create playbooks for

Playbooks should be created for incident scenarios such as:

- **Expected incidents** – Playbooks should be created for incidents you anticipate. This includes threats like denial of service (DoS), ransomware, and credential compromise.
- **Known security findings or alerts** – Playbooks should be created for your known security findings and alerts, such as GuardDuty findings. You might receive a GuardDuty finding and think, "Now what?" To prevent mishandling of a GuardDuty finding or ignoring the finding, create a playbook for each potential GuardDuty finding. Some remediation details and guidance can be found in the GuardDuty documentation. It's worth noting that GuardDuty is not enabled by default and does incur a cost. More details on GuardDuty can be found in Appendix A: Cloud capability definitions - the section called "Visibility and alerting".

What to include in playbooks

Playbooks should contain technical steps for a security analyst to complete in order to adequately investigate and respond to a potential security incident.

Items to include in a playbook include:

- **Playbook overview** – What risk or incident scenario does this playbook address? What is the goal of the playbook?

- **Prerequisites** – What logs and detection mechanisms are required for this incident scenario? What is the expected notification?
- **Stakeholder information** – Who is involved and what is their contact information? What are each of the stakeholders' responsibilities?
- **Response steps** – Across phases of incident response, what tactical steps should be taken? What queries should an analyst run? What code should be run to achieve the desired outcome?
  - **Detect** – How will the incident be detected?
  - **Analyze** – How will the scope of impact be determined?
  - **Contain** – How will the incident be isolated to limit scope?
  - **Eradicate** – How will the threat be removed from the environment?
  - **Recover** – How will the affected system or resource be brought back into production?
- **Expected outcomes** – After queries and code are run, what is the expected result of the playbook?

To verify consistent information in each playbook, it can be helpful to create a playbook template to use across your other security playbooks. Some of the previously listed items, such as stakeholder information, can be shared across multiple playbooks. If that is the case, you can create centralized documentation for that information and reference it in the playbook, then enumerate the explicit differences in the playbook. This will prevent you from having to update the same information in all of your individual playbooks. Through creating a template and identifying common or shared information in playbooks, you can simplify and speed up playbook development. Lastly, your playbooks will likely evolve over time; once you have confirmed that the steps are consistent, this forms the requirements for automation.

Sample playbooks

A number of sample playbooks can be found in Appendix B in the section called "Playbook resources". The examples here can be used to guide you on what playbooks to create and what to include in your playbooks. However, it's important you craft playbooks that incorporate the risks most relevant to your business. You need to verify that the steps and workflows within your playbooks include your technologies and processes.

Run regular simulations

Organizations grow and evolve over time, as does the threat landscape. Because of this, it's important to continually review your incident response capabilities. Simulations are one method

that can be used to perform this assessment. Simulations use real-world security event scenarios designed to mimic a threat actor's tactics, techniques, and procedures (TTPs) and allow an organization to exercise and evaluate their incident response capabilities by responding to these mock cyber events as they might occur in reality.

Simulations have a variety of benefits, including:

- Validating cyber readiness and developing the confidence of your incident responders.
- Testing the accuracy and efficiency of tools and workflows.
- Refining communication and escalation methods aligned with your incident response plan.
- Providing an opportunity to respond to less common vectors.

Types of simulations

There are three main types of simulations:

- **Tabletop exercises** – The tabletop approach to simulations is strictly a discussion-based session involving the various incident response stakeholders to practice roles and responsibilities and use established communication tools and playbooks. Exercise facilitation can typically be accomplished in a full day in a virtual venue, physical venue, or a combination. Because of its discussion-based nature, the tabletop exercise focuses on processes, people, and collaboration. Technology is an integral part of the discussion; however, the actual use of incident response tools or scripts is generally not a part of the tabletop exercise.
- **Purple Team exercises** – Purple Team exercises increase the level of collaboration between the incident responders (*Blue Team*) and simulated threat actors (*Red Team*). The Blue Team is generally comprised of members of the Security Operations Center (SOC), but can also include other stakeholders that would be involved during an actual cyber event. The Red Team is generally comprised of a penetration testing team or key stakeholders that are trained in offensive security. The Red Team works collaboratively with the exercise facilitators when designing a scenario so that the scenario is accurate and feasible. During Purple Team exercises, the primary focus is on the detection mechanisms, the tools, and the standard operating procedures (SOPs) supporting the incident response efforts.
- **Red Team exercises** – During a Red Team exercise, the offense (*Red Team*) conducts a simulation to achieve a certain objective or set of objectives from a pre-determined scope. The defenders (*Blue Team*) will not necessarily have knowledge of the scope and duration of the exercise, which provides a more realistic assessment of how they would respond to an actual incident. Because

Red Team exercises can be invasive tests, you should be cautious and implement controls to verify that the exercise does not cause actual harm to your environment.

> **Note**
>
> AWS requires customers to review the policy for penetration testing available on the Penetration Testing website before they conduct Purple Team or Red Team exercises.

Table 1 summarizes a few key differences in these types of simulations. It's important to note that the definitions are generally considered loose definitions and can be customized to fit the needs of your organization.

Table 1 – Types of simulations

	Tabletop exercise	Purple Team exercise	Red Team exercise
Summary	Paper-driven exercises that focus on one specific security incident scenario. These can be either high-level or technical, and are driven by a series of paper injects.	A more realistic offering compared to tabletop exercises. During Purple Team exercises, facilitators work collaboratively with the participants to increase exercise engagement and offer training where necessary.	Generally a more advanced simulation offering. There is usually a high level of covertness, where the participants might not know all of the details of the exercise.
Resources required	Limited technical resources required	Various stakeholders required and high level of technical resources needed	Various stakeholders required and high level of technical resources needed
Complexity	Low	Medium	High

Consider facilitating cyber simulations at a regular interval. Each exercise type can provide unique benefits to the participants and the organization as a whole, so you might choose to start with less complex simulation types (such as tabletop exercises) and progress to more complex simulation types (Red Team exercises). You should select a simulation type based on your security maturity, resources, and your desired outcomes. Some customers might not choose to perform Red Team exercises due to complexity and cost.

Exercise lifecycle

Regardless of the type of simulation you choose, simulations generally follow these steps:

1. **Define core exercise elements** – Define the simulation scenario and the objectives of the simulation. Both of these should have leadership acceptance.
2. **Identify key stakeholders** – At a minimum, an exercise needs exercise facilitators and participants. Depending on the scenario, additional stakeholders such as legal, communications, or executive leadership might be involved.
3. **Build and test the scenario** – The scenario might need to be redefined as it is being built if specific elements aren't feasible. A finalized scenario is expected as the output of this stage.
4. **Facilitate the simulation** – The type of simulation determines the facilitation used (paper-based scenario compared to highly technical, simulated scenario). The facilitators should align their facilitation tactics to the exercise objects and they should engage all exercise participants wherever possible to provide the most benefit.
5. **Develop the after action report (AAR)** – Identify areas that went well, those that can use improvement, and potential gaps. The AAR should measure the effectiveness of the simulation as well as the team's response to the simulated event so that progress can be tracked over time with future simulations.

Technology

If you develop and implement the appropriate technologies before a security incident, your incident response staff will be able to investigate, understand scope, and take action in a timely manner.

Develop AWS account structure

AWS Organizations helps centrally manage and govern an AWS environment as you grow and scale AWS resources. An AWS organization consolidates your AWS accounts so that you can administer

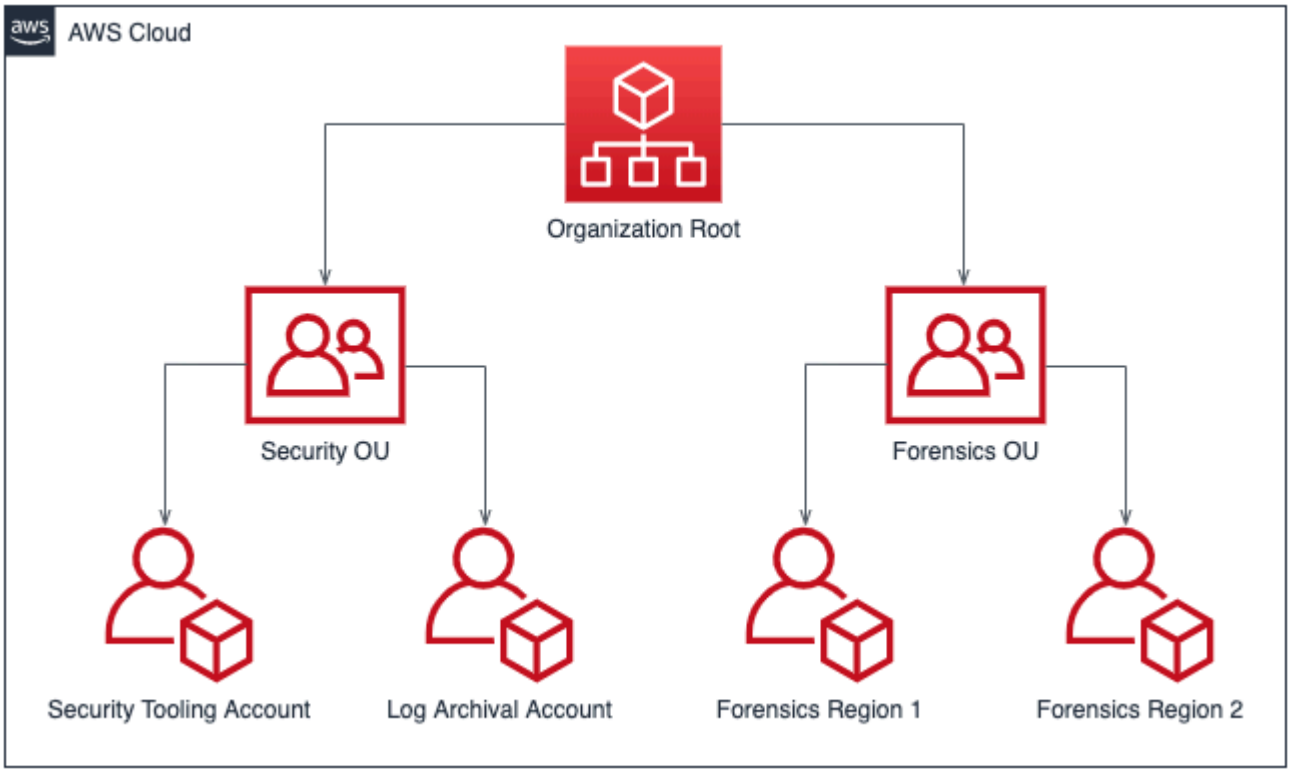
them as a single unit. You can use organizational units (OUs) to group accounts together to administer as a single unit.

For incident response, it's helpful to have an AWS account structure that supports the functions of incident response, which includes a *security OU* and a *forensics OU*. Within the security OU, you should have accounts for:

- **Log archival** – Aggregate logs in a log archival AWS account.
- **Security tooling** – Centralize security services in a security tool AWS account. This account operates as the delegated administrator for security services.

Within the forensics OU, you have the option to implement a single forensics account or accounts for each Region that you operate in, depending on which works best for your business and operational model. For an example of a per-Region account approach, if you only operate in US East (N. Virginia) (us-east-1) and US West (Oregon) (us-west-2), then you would have two accounts in the forensics OU: one for us-east-1 and one for us-west-2. Because it takes time to provision new accounts, it is imperative to create and instrument the forensics accounts well ahead of an incident so that responders can be prepared to effectively use them for response.

The following diagram displays a sample account structure including a forensics OU with per-Region forensics accounts:



Per-region account structure for incident response

Develop and implement a tagging strategy

Obtaining contextual information on the business use case and relevant internal stakeholders surrounding an AWS resource can be difficult. One way to do this is in the form of tags, which assign metadata to your AWS resources and consist of a user-defined key and value. You can create tags to categorize resources by purpose, owner, environment, type of data processed, and other criteria of your choice.

Having a consistent tagging strategy can speed up response times by allowing you to quickly identify and discern contextual information about an AWS resource. Tags can also serve as a mechanism to initiate response automations. For further information on what to tag, refer to the documentation on tagging AWS resources. You'll want to first define the tags you want to implement across your organization. After that, you'll implement and enforce this tagging strategy. Details on implementation and enforcement can be found in the AWS blog Implement AWS resource tagging strategy using AWS Tag Policies and Service Control Policies (SCPs).

Update AWS account contact information

For each of your AWS accounts, it's important to have accurate and up-to-date contact information so that the correct stakeholders receive important notifications from AWS on topics like security, billing, and operations. For each AWS account, you have a primary contact and alternate contacts for security, billing and operations. Differences between these contacts can be found in the AWS Account Management Reference Guide.

For details on managing alternate contacts, refer to the AWS documentation on adding, changing, or removing alternate contacts. It's a best practice to use an email distribution list if your team manages billing, operations, and security-related issues. An email distribution list removes dependencies on one person, which can cause blockages if they are out of the office or leave the company. You should also verify that the email and account contact information, including the phone number, are well protected to defend against root account password resets and multi-factor authentication (MFA) resets.

For customers using AWS Organizations, organization administrators can centrally manage alternate contacts for member accounts using the management account or a delegated administrator account without requiring credentials for each AWS account. You will also need to verify that newly created accounts have accurate contact information. Refer to the Automatically update alternate contacts for newly created AWS accounts blog post.

Prepare access to AWS accounts

During an incident, your incident response teams must have access to the environments and resources involved in the incident. Ensure that your teams have appropriate access to perform their duties before an event occurs. To do that, you should know what level of access your team members require (for example, what kinds of actions they are likely to take) and should provision least-privilege access in advance.

To implement and provision this access, you should identify and discuss the AWS account strategy and cloud identity strategy with your organization's cloud architects to understand what authentication and authorization methods are configured. Due to the privileged nature of these credentials, you should consider using approval flows or retrieving credentials from a vault or safe as part of your implementation. After implementation, you should document and test the team members' access well before an event occurs to make sure they can respond without delays.

Lastly, users that are created specifically to respond to a security incident are often privileged in order to provide sufficient access. Therefore, use of these credentials should be restricted, monitored, and not used for daily activities.

Understand the threat landscape

Develop threat models

By developing threat models, organizations can identify threats and mitigations before an unauthorized user can. There are a number of strategies and approaches to threat modeling; refer to the How to approach threat modeling blog post. For incident response, a threat model can help identify the attack vectors a threat actor might have used during an incident. Understanding what you're defending against will be crucial in order to respond in a timely manner. You can also use an AWS Partner for threat modeling. To search for an AWS partner, use the AWS Partner Network.

Integrate and use cyber threat intelligence

Cyber threat intelligence is the data and analysis of a threat actor's intent, opportunity, and capability. Obtaining and using threat intelligence is helpful to detect an incident early and to better understand threat actor behavior. Cyber threat intelligence includes static indicators like IP addresses or file hashes of malware. It also includes high-level information, like behavioral patterns and intent. You can collect threat intelligence from a number of cyber security vendors and from open-source repositories.

To integrate and maximize threat intelligence for your AWS environment, you can use some out-of-the-box capabilities and integrate your own threat intelligence lists. Amazon GuardDuty uses AWS internal and third-party threat intelligence sources. Other AWS services, such as a DNS firewall and AWS WAF rules, also take inputs from AWS' advanced threat intelligence group. Some GuardDuty findings are mapped to the MITRE ATT&CK Framework, which provides information on real-world observations on adversary tactics and techniques.

Select and set up logs for analysis and alerting

During a security investigation, you need to be able to review relevant logs to record and understand the full scope and timeline of the incident. Logs are also required for alert generation, indicating certain actions of interest have happened. It is critical to select, enable, store, and set up querying and retrieval mechanisms, and set up alerting. Each of these actions are reviewed in this section. For more details, see the Logging strategies for security incident response AWS blog post.

Select and enable log sources

Ahead of a security investigation, you need to capture relevant logs to retroactively reconstruct activity in an AWS account. Select and enable log sources relevant to their AWS account workloads.

AWS CloudTrail is a logging service that tracks API calls made against an AWS account capturing AWS service activity. It is enabled by default with 90-day retention of management events that can be retrieved through CloudTrail's Event History facility using AWS Management Console, the AWS CLI, or an AWS SDK. For longer retention and visibility of data events, you need to create a CloudTrail Trail and associated with an Amazon S3 bucket, and optionally, with a CloudWatch log group. Alternatively, you can create a CloudTrail Lake, which retains CloudTrail logs for up to seven years and provides a SQL-based querying facility.

AWS recommends that customers using a VPC enable network traffic and DNS logs using, respectively, VPC Flow Logs and Amazon Route 53 resolver query logs, streaming them to either an Amazon S3 bucket or a CloudWatch log group. You can create a VPC flow log for a VPC, a subnet, or a network interface. For VPC Flow Logs, you can be selective on how and where you enable Flow Logs to reduce cost.

AWS CloudTrail Logs, VPC Flow Logs, and Route 53 resolver query logs are the *basic logging trifecta* to support security investigations in AWS.

AWS services can generate logs not captured by the basic logging trifecta, such as Elastic Load Balancing logs, AWS WAF logs, AWS Config recorder logs, Amazon GuardDuty findings, Amazon Elastic Kubernetes Service (Amazon EKS) audit logs, and Amazon EC2 instance operating system and application logs. Refer to *Appendix A: Cloud capability definitions* for the full list of logging and monitoring options.

Select log storage

The choice of log storage is generally related to which querying tool you use, retention capabilities, familiarity, and cost. When you enable AWS service logs, provide a storage facility; usually an Amazon S3 bucket or CloudWatch log group.

An Amazon S3 bucket provides cost-effective durable storage with an optional lifecycle policy. Logs stored in Amazon S3 buckets can be natively queried using services such as Amazon Athena. A CloudWatch log group provides durable storage and a built-in query facility through CloudWatch Logs Insights.

Identify appropriate log retention

When you use an S3 bucket or CloudWatch log group to store logs, you must establish adequate lifecycles for each log source to optimize storage and retrieval costs. Customers generally have between 3 and 12 months of logs readily available for querying, with retention of up to seven years. The choice of availability and retention should align with your security requirements and a composite of statutory, regulatory, and business mandates.

Select and implement querying mechanisms for logs

In AWS, the main services you can use to query logs are CloudWatch Logs Insights for data stored in CloudWatch log groups, and Amazon Athena and Amazon OpenSearch Service for data stored in Amazon S3. You can also use third-party querying tools such as a security information and event management (SIEM).

The process for selecting a log querying tool should consider the people, process, and technology aspects of your security operations. Select a tool that fulfills operational, business, and security requirements, and is both accessible and maintainable in the long term. Keep in mind that log querying tools work optimally when the number of logs to be scanned is kept within the tool's limits. It is not uncommon for customers to have multiple querying tools because of cost or technical constraints. For example, customers might use a third-party SIEM to perform queries for the last 90 days of data, and use Athena to perform queries beyond 90 days because of the log ingestion cost of a SIEM. No matter the implementation, verify that your approach minimizes the number of tools required to maximize operational efficiency, especially during a security event investigation.

Use logs for alerting

AWS natively provides alerting through security services, such as Amazon GuardDuty, AWS Security Hub, and AWS Config. You can also use custom alert generation engines for security alerts not covered by these services or for specific alerts relevant to your environment. Building these alerts and detections is covered in the section called the section called "Detection" in this document.

Develop forensics capabilities

Ahead of a security incident, consider developing forensics capabilities to support security event investigations. The Guide to Integrating Forensic Techniques into Incident Response by NIST provides such guidance.

Forensics on AWS

Concepts from traditional on-premises forensics apply to AWS. The Forensic investigation environment strategies in the AWS Cloud blog post provides you with key information to start migrating their forensic expertise to AWS.

Once you have your environment and AWS account structure set up for forensics, you'll want to define the technologies required to effectively perform forensically sound methodologies across the four phases:

- **Collection** – Collect relevant AWS logs, such as AWS CloudTrail, AWS Config, VPC Flow Logs, and host-level logs. Collect snapshots, backups, and memory dumps of impacted AWS resources.
- **Examination** – Examine the data collected by extracting and assessing the relevant information.
- **Analysis** – Analyze the data collected in order to understand the incident and draw conclusions from it.
- **Reporting** – Present the information resulting from the analysis phase.

Capture backups and snapshots

Setting up backups of key systems and databases are critical for recovering from a security incident and for forensics purposes. With backups in place, you can restore your systems to their previous safe state. On AWS, you can take snapshots of various resources. Snapshots provide you with point-in-time backups of those resources. There are many AWS services that can support you in backup and recovery. Refer to the Backup and Recovery Prescriptive Guidance for details on these services and approaches for backup and recovery. For more details, see the Use backups to recover from security incidents blog post.

Especially when it comes to situations such as ransomware, it's critical for your backups to be well protected. Refer to the Top 10 security best practices for securing backups in AWS blog post for guidance on securing your backups. In addition to securing your backups, you should regularly test your backup and restore processes to verify that the technology and processes you have in place work as expected.

Automation of forensics on AWS

During a security event, your incident response team must be able to collect and analyze evidence quickly while maintaining accuracy for the time period surrounding the event. It's both challenging

and time consuming for the incident response team to manually collect the relevant evidence in a cloud environment, especially across a large number of instances and accounts. Additionally, manual collection can be prone to human error. For these reasons, customers should develop and implement automation for forensics.

AWS offers a number of automation resources for forensics, which are consolidated in the Appendix under the section called "Forensic resources". These resources are examples of forensics patterns that we have developed and customers have implemented. While they might be a useful reference architecture to start with, consider modifying them or creating new forensics automation patterns based on your environment, requirements, tools, and forensics processes.

Summary of preparation items

Thorough preparation for responding to security events is critical for timely and effective incident response. Incident response preparation involves people, processes, and technology. All three of these domains are equally important to preparation. You should prepare and evolve your incident response program across all three domains.

Table 2 summarizes the preparation items detailed in this section.

Table 2 – Incident response preparation items

Domain	Preparation item	Action items
People	Define roles and responsib ilities.	• Identify relevant incident response stakeholders. • Develop a responsible, accountable, informed, consulted (RACI) chart for an incident.
People	Train incident response staff on AWS.	• Train incident response stakeholders on AWS foundations. • Train incident response stakeholders on AWS security and monitoring services.

Domain	Preparation item	Action items
		• Train incident response stakeholders on your AWS environment and how it is architected.
People	Understand AWS support options.	• Understand differences in AWS support, Customer Incident Response Team (CIRT), DDoS response team (DRT) and AMS. • Understand triage and escalation path to reach the CIRT during an active security event if needed.
Process	Develop an incident response plan.	• Create a high-level document that defines your incident response program and strategy. • Include a RACI, communica tion plan, incident definitio ns, and phases of incident response to the incident response plan.
Process	Document and centralize architecture diagrams.	• Document details on how your AWS environment is configured across account structure, service usages, IAM patterns, and other core functionality to your AWS configuration. • Develop architecture diagrams of your cloud architectures.

Domain	Preparation item	Action items
Process	Develop incident response playbooks.	• Create a template for structure of your playbooks. • Build playbooks for expected security events. • Build playbooks for known security alerts, such as GuardDuty findings.
Process	Run regular simulations.	• Develop a regular cadence to run incident simulations. • Use the outputs and lessons learned to iterate on your incident response program.
Technology	Develop AWS account structure.	• Plan an account structure for how workloads are separated by AWS accounts. • Create a security OU with a security tooling and log archival account. • Create a forensics OU with forensics accounts for each Region in which you operate.
Technology	Develop and implement a tagging strategy that helps responders to identify ownership and context for findings.	• Plan a strategy for tagging and what tags you want associated with your AWS resources. • Implement and enforce the tagging strategy.

Domain	Preparation item	Action items
Technology	Update AWS account contact information.	• Verify that AWS accounts have contact information listed. • Create email distribut ion lists for the contact information to remove single points of failure. • Protect the email accounts that are associated with the AWS account information.
Technology	Prepare access to AWS accounts.	• Define what access incident responders will require to respond to an incident. • Implement, test and monitor the access.
Technology	Understand threat landscape.	• Develop threat models of your environment and applications. • Integrate and use cyber threat intelligence.
Technology	Select and set up logs.	• Identify and enable logs for investigations. • Select log storage. • Identify and implement log retention. • Develop mechanism to retrieve and query logs and artifacts. • Use logs for alerting.

Domain	Preparation item	Action items
Technology	Develop forensics capabilities.	• Identify artifacts required for forensics collection. • Capture and secure backups of key systems. • Define mechanisms for analysis of identified logs and artifacts. • Implement automation for forensics analysis.

An iterative approach is recommended for incident response preparation. All of these preparation items cannot be done overnight; you should create a plan to start small and continuously improve your incident response capabilities over time.

Operations

Operations is the core of performing incident response. This is where the actions of responding and remediating security incidents occur. Operations includes the following five phases: *detection*, *analysis*, *containment*, *eradication*, and *recovery*. Descriptions of these phases and the goals can be found in Table 3.

Table 3 – Operations phases

Phase	Goal
Detection	Identify a potential security event.
Analysis	Determine if security event is an incident and assess the scope of the incident.
Containment	Minimize and limit the scope of the security event.
Eradication	Remove unauthorized resources or artifacts related to the security event. Implement mitigations that caused the security incident.
Recovery	Restore systems to known safe state and monitor these systems to verify that the threat does not return.

The phases should serve as guidance when you respond to and operate on security incidents in order to respond in an effective and robust way. The actual actions you take will vary depending on the incident. An incident involving ransomware, for example, will have a different set of response steps to follow than an incident involving a public Amazon S3 bucket. Additionally, these phases do not necessarily happen sequentially. After containment and eradication, you might need to return to analysis to understand if your actions were effective.

Detection

An alert is the main component of the detection phase. It generates a notification to initiate the incident response process based on AWS account activity of interest.

Alerting accuracy is challenging; it's not always possible to determine with complete certainty if an incident has occurred, is in progress, or if it will happen in the future. Here are a few reasons:

- Detection mechanisms are based on baseline deviation, known patterns, and notification from internal or external entities.
- Because of the unpredictable nature of technology and people, respectively *the means* and *the actors* of security incidents, baselines change over time. Rogue patterns emerge through novel or modified threat actor *tactics*, *techniques*, and *procedures* (TTPs).
- Changes to people, technology, and processes are not immediately incorporated into the incident response process. Some are discovered during the progress of an investigation.

Alert sources

You should consider using the following sources to define alerts:

- **Findings** – AWS services such as Amazon GuardDuty, AWS Security Hub, Amazon Macie, Amazon Inspector, AWS Config, IAM Access Analyzer, and Network Access Analyzer generate findings that can be used to craft alerts.
- **Logs** – AWS service, infrastructure, and application logs stored in Amazon S3 buckets and CloudWatch log groups can be parsed and correlated to generate alerts.
- **Billing activity** – A sudden change in billing activity can indicate a security event. Follow the documentation on Creating a billing alarm to monitor your estimated AWS charges to monitor for this.
- **Cyber threat intelligence** – If you subscribe to a third-party cyber threat intelligence feed, you can correlate that information with other logging and monitoring tools to identify potential indicators of events.
- **Partner tools** – Partners in the AWS Partner Network (APN) offer top-tier products that can help you meet your security objectives. For incident response, partner products with endpoint detection and response (EDR) or SIEM can help support your incident response objectives. For more information, see Security Partner Solutions and Security Solutions in the AWS Marketplace.

- **AWS trust and safety** – AWS Support might contact customers if we identify abusive or malicious activity.
- **One-time contact** – Because it can be your customers, developers, or other staff in your organization who notice something unusual, it's important to have a well-known, well-publicized method of contacting your security team. Popular choices include ticketing systems, contact email addresses, and web forms. If your organization works with the general public, you might also need a public-facing security contact mechanism.

For more information about cloud capabilities that you can use during your investigations, refer to *Appendix A: Cloud capability definitions* in this document.

Detection as part of security control engineering

Detection mechanisms are an integral part of security control development. As *directive* and *preventative* controls are defined, related *detective* and *responsive* controls should be constructed. As an example, an organization establishes a directive control related to the root user of an AWS account, which should only be used for specific and very well-defined activities. They associate it with a preventative control implemented by using an AWS organization's service control policy (SCP). If root user activity beyond the expected baseline happens, a detective control implemented with an EventBridge rule and SNS topic will alert the security operations center (SOC). The responsive control entails the SOC selecting the appropriate playbook, performing analysis, and working until the incident is resolved.

Security controls are best defined by threat modeling of workloads running in AWS. The criticality of detective controls will be set by looking at the business impact analysis (BIA) for the particular workload. Alerts generated by detective controls are not handled as they come in, but rather based on its initial criticality, to be adjusted during analysis. The initial criticality set is an aid for prioritization; the context in which the alert happened will determine its true criticality. As an example, an organization uses Amazon GuardDuty as a component of the detective control used for EC2 instances that are part of a workload. The finding `Impact:EC2/SuspiciousDomainRequest.Reputation` is generated, informing you that the listed Amazon EC2 instance within your workload is querying a domain name that is suspected of being malicious. This alert is set by default as low severity, and as the analysis phase progresses, it was determined that several hundred EC2 instances of type `p4d.24xlarge` have been deployed by an unauthorized actor, significantly increasing the organization's operating cost. At this point, the incident response team makes the decision to adjust the criticality of this alert to *high*, increasing the sense of urgency and expediting further actions. Note that the GuardDuty finding severity

cannot be changed. Rather, the organization's alert based on the finding will have to be criticality adjusted.

Detective control implementations

It is important to understand how detective controls are implemented because they help determine how the alert will be used for the particular event. There are two main implementations of technical detective controls:

- **Behavioral detection** relies on mathematical models commonly referred to as machine learning (ML) or artificial intelligence (AI). The detection is made by inference; therefore, the alert might not necessarily reflect an actual event.
- **Rule-based detection** is deterministic; customers can set the exact parameters of what activity to be alerted on, and that is certain.

Modern implementations of detective systems, such as an intrusion detection system (IDS), generally come with both mechanisms. Following are some examples for rule-based and behavioral detections with GuardDuty.

- When the finding `Exfiltration:IAMUser/AnomalousBehavior` is generated, it informs you that "an anomalous API request was observed in your account." As you look further into the documentation, it tells you that "The ML model evaluates all API requests in your account and identifies anomalous events that are associated with techniques used by adversaries," indicating that this finding is of behavioral nature.
- For the finding `Impact:S3/MaliciousIPCaller`, GuardDuty is analyzing API calls from the Amazon S3 service in CloudTrail, comparing the `SourceIPAddress` log element with a table of public IP addresses that includes threat intelligence feeds. Once it finds a direct match to an entry, it generates the finding.

We recommend implementing a mix of both behavioral and rule-based alerting because it is not always possible to implement rule-based alerting for every activity within your threat model.

People-based detection

Up to this point, we have discussed technology-based detection. The other important source of detection comes from people inside or outside the customer's organization. *Insiders* can be

defined as an employee or contractor, and *outsiders* are entities such as security researchers, law enforcement, the news, and social media.

Though technology-based detection can be systematically configured, people-based detection comes in a variety of forms such as emails, tickets, mail, news posts, telephone calls, and in-person interactions. Technology-based detection notifications can be expected to be delivered in near real-time, but there are no timeline expectations for people-based detection. It is imperative that the security culture incorporates, facilitates, and empowers people-based detection mechanisms for a defense-in-depth approach to security.

Summary

With detection, it's important to have a mix of rule-based and behavioral driven alerting. Additionally, you should have mechanisms in place for people both internally and externally to submit a ticket about a security issue. Humans can be one of the most valuable sources for security events, so it's important to have processes in place for people to escalate concerns. You should use threat models of your environment to get started with building detections. Threat models will help you build alerts based on threats that are most relevant to your environment. Lastly, you should use frameworks such as MITRE ATT&CK to understand threat actor tactics, techniques, and procedures (TTPs). The MITRE ATT&CK framework can be helpful to use as a common language across your various detection mechanisms.

Analysis

Logs, query capabilities, and threat intelligence are a few of the supporting components required by the analysis phase. Many of the same logs used for detection are also used for analysis and will require onboarding and configuration of querying tools.

Validate, scope, and assess impact of alert

During the analysis phase, comprehensive log analysis is performed with the goal to validate alerts, define scope, and assess impact of the possible compromise.

- *Validation* of the alert is the entry point of the analysis phase. Incident responders will be looking for log entries from various sources and directly engaging with owners of the affected workload.
- *Scoping* is the next step, when all resources involved are inventoried and alert criticality is adjusted after stakeholders agree that it is unlikely to be a false-positive.

- Finally, *impact analysis* details the actual business disruption.

Once the affected workload components are identified, scoping results can be correlated with the related workload's recovery point objective (RPO) and recovery time objective (RTO), adjusting for alert criticality, which will initiate resource allocation and all activity happening next. Not all incidents will directly disrupt operations of a workload supporting a business process. Incidents such as sensitive data disclosure, intellectual property theft, or resource hijacking (as in cryptocurrency mining) might not stop or debilitate a business process immediately, but can result in consequences at a later time.

Enrich security logs and findings

Enrichment with threat intelligence and organizational context

During the course of analysis, observables of interest require enrichment for enhanced contextualization of the alert. As stated in the Preparation section, integrating and leveraging cyber threat intelligence can be helpful to understand more about a security finding. Threat intelligence services are used to assign reputation and attribute ownership to public IP addresses, domain names, and file hashes. These tools are available as paid and no charge services.

Customers adopting Amazon Athena as a log querying tool gain the advantage of AWS Glue jobs to load threat intelligence information as tables. The threat intelligence tables can be used in SQL queries to correlate log elements such as IP addresses and domain names, providing an enriched view of the data to be analyzed.

AWS does not provide threat intelligence directly to customers, but services such as Amazon GuardDuty makes use of threat intelligence for enrichment and finding generation. You can also upload custom threat lists to GuardDuty based on your own threat intelligence.

Enrichment with automation

Automation is an integral part of AWS Cloud governance. It can be used throughout the various phases of the incident response lifecycle.

For the detection phase, rule-based automation matches patterns of interest from the threat model in logs and takes appropriate action, such as sending notifications. The analysis phase can leverage the detection mechanism and forward the alert body to an engine capable of querying logs and enriching observables for contextualization of the event.

The alert body, in its fundamental form, is comprised of a *resource* and an *identity*. As an example, you could implement an automation to query CloudTrail for AWS API activity performed by the alert body's identity or resource around the time of the alert, providing additional insights including `eventSource`, `eventName`, `SourceIPAddress`, and `userAgent` of identified API activity. By performing these queries in an automated way, responders can save time during triage and get additional context to help make better informed decisions.

Refer to the How to enrich AWS Security Hub findings with account metadata blog post for an example on how to use automation to enrich security findings and simplify analysis.

Collect and analyze forensic evidence

Forensics, as mentioned in the *Preparation* section of this document, is the process of collecting and analyzing artifacts during incident response. On AWS, it is applicable to infrastructure domain resources such as network traffic packet captures, operating system memory dump, and for service domain resources such as AWS CloudTrail logs.

The forensics process has the following fundamental characteristics:

- **Consistent** – It follows the exact steps documented, without deviations.
- **Repeatable** – It produces the exact same results when repeated against the same artifact.
- **Customary** – It's publicly documented and widely adopted.

It is important to maintain chain of custody for artifacts collected during incident response. Using automation and having automatic documentation of this collection generated can help, in addition to storing the artifacts in read-only repositories. Analysis should only be performed on exact replicas of the collected artifacts to maintain integrity.

Collect relevant artifacts

With these characteristics in mind, and based on the relevant alerts and assessment of impact and scope, you will need to collect the data that will be relevant to further investigation and analysis. Various types and sources of data that might be relevant to investigation, including service/control plane logs (CloudTrail, Amazon S3 data events, VPC Flow Logs), data (Amazon S3 metadata and objects), and resources (databases, Amazon EC2 instances).

Service/control plane logs can be collected for local analysis or, ideally, directly queried using native AWS services (where applicable). Data (including metadata) can be directly queried to obtain relevant information or to acquire the source objects; for example, use the AWS CLI to acquire

Amazon S3 bucket and object metadata and directly acquire source objects. Resources need to be collected in a manner consistent with the resource type and intended method of analysis. For example, databases can be collected by creating a copy/snapshot of the system running the database, creating a copy/snapshot of the entire database itself, or querying and extracting certain data and logs from the database relevant to the investigation.

For Amazon EC2 instances, there is a specific set of data that should be collected and a specific order to collection that should be performed in order to acquire and preserve the most amount of data for analysis and investigation.

Specifically, the order for response to acquire and preserve the most amount of data from an Amazon EC2 instance is the following:

1. **Acquire instance metadata** – Acquire instance metadata relevant to the investigation and data queries (instance ID, type, IP address, VPC/subnet ID, Region, Amazon Machine Image (AMI) ID, security groups attached, launch time).
2. **Enable instance protections and tags** – Enable instance protections like termination protection, setting shutdown behavior to stop (if set to terminate), disabling Delete on Termination attributes for the attached EBS volumes, and applying appropriate tags for both visual denotation and use in possible response automations (for example, upon applying a tag with name of `Status` and value of `Quarantine`, perform forensic acquisition of data and isolate the instance).
3. **Acquire disk (EBS snapshots)** – Acquire an EBS snapshot of the attached EBS volumes. Each snapshot contains the information that you need to restore your data (from the moment when the snapshot was taken) to a new EBS volume. See the step to perform live response/artifact collection if you're using instance store volumes.
4. **Acquire memory** – Because EBS snapshots only capture data that has been written to your Amazon EBS volume, which might exclude data that is stored or cached in memory by your applications or OS, it is imperative to acquire a system memory image using an appropriate third-party open-source or commercial tool in order to acquire available data from the system.
5. **(Optional) Perform live response/artifact collection** – Perform targeted data collection (disk/memory/logs) through live response on the system only if disk or memory is unable to be acquired otherwise, or there is a valid business or operational reason. Doing this will modify valuable system data and artifacts.
6. **Decommission the instance** – Detach the instance from auto scaling groups, deregister the instance from load balancers, and adjust or apply a pre-built instance profile with minimized or no permissions.

7. **Isolate or contain the instance** – Verify that the instance is effectively isolated from other systems and resources within the environment by ending and preventing current and future connections to and from the instance. Refer to the the section called “Containment” section of this document for more details.
8. **Responder’s choice** – Based on the situation and goals, select one of the following:
   - Decommission and shut down the system (recommended).

     Shut the system down once the available evidence has been acquired in order to verify the most effective mitigation against a possible future impact to the environment by the instance.
   - Continue running the instance within an isolated environment instrumented for monitoring.

     Though it is not recommended as a standard approach, if a situation merits continued observation of the instance (such as when additional data or indicators are needed to perform comprehensive investigation and analysis of the instance), you might consider shutting down the instance, creating an AMI of the instance, and re-launching the instance in your dedicated forensics account within a sandbox environment that is pre-instrumented to be completely isolated and configured with instrumentation to facilitate continuous monitoring of the instance (for example, VPC Flow Logs or VPC Traffic Mirroring).

> **Note**
>
> It is essential to capture memory before live response activities or system isolation or shutdown in order to capture available volatile (and valuable) data.

Develop narratives

During analysis and investigation, document the actions taken, analysis performed, and information identified, to be used by the subsequent phases and ultimately a final report. These narratives should be succinct and precise, confirming that relevant information is included to verify effective understanding of the incident and to maintain an accurate timeline. They are also helpful when you engage people outside of the core incident response team. Here is an example:

> *The marketing and sales department received a ransom note on March 15th, 2022 demanding payment in cryptocurrency to avoid public posting of possible sensitive data. The SOC determined that the Amazon RDS database belonging to marketing and sales was*

publicly accessible on February 20th, 2022. The SOC queried RDS access logs and determined that IP address 198.51.100.23 was used on February 20th, 2022 with the credentials `mm03434` belonging to Major Mary, *one of the web developers. The SOC queried VPC Flow Logs and determined that approximately 256MB of data egressed to the same IP address at the same date (time stamp 2022-02-20T15:50+00Z). The SOC determined through open-source threat intelligence that the credentials are currently available in plain text in the public repository `https[:]//example[.]com/majormary/rds-utils`.*

Containment

One definition of containment, as it relates to incident response, is the process or implementation of a strategy during the handling of a security event that acts to minimize the scope of the security event and contain the effects of unauthorized usage within the environment.

A containment strategy depends on a myriad of factors and can be different from one organization to another in terms of application of containment tactics, timing, and purpose. The NIST SP 800-61 Computer Security Incident Handling Guide outlines several criteria for determining the appropriate containment strategy, which include:

- Potential damage to and theft of resources
- Need for evidence preservation
- Service availability (network connectivity, services provided to external parties)
- Time and resources needed to implement the strategy
- Effectiveness of the strategy (partial or full containment)
- Duration of the solution (emergency workaround to be removed in four hours, temporary workaround to be removed in two weeks, permanent solution)

Regarding services on AWS, however, the fundamental containment steps can be distilled down to three categories:

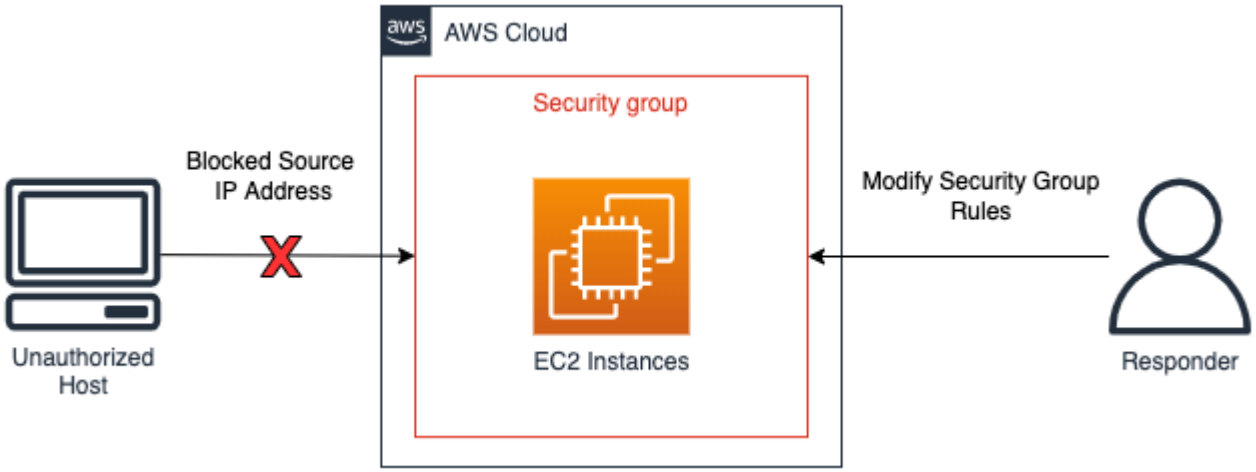
- **Source containment** – Use filtering and routing to prevent access from a certain source.
- **Technique and access containment** – Remove access to prevent unauthorized access to the affected resources.
- **Destination containment** – Use filtering and routing to prevent access to a target resource.

Source containment

Source containment is the use and application of filtering or routing within an environment to prevent access to resources from a specific source IP address or network range. Examples of source containment using AWS services are highlighted here:

- **Security groups** – Creating and applying isolation security groups to Amazon EC2 instances or removing rules from an existing security group can help to contain unauthorized traffic to an Amazon EC2 instance or AWS resource. It is important to note that existing tracked connections won't be shut down as a result of changing security groups – only future traffic will be effectively blocked by the new security group (refer to this Incident Response Playbook and Security group connection tracking for additional information on tracked and untracked connections).
- **Policies** – Amazon S3 bucket policies can be configured to block or allow traffic from an IP address, a network range, or a VPC endpoint. Policies create the ability to block suspicious addresses and access to the Amazon S3 bucket. Additional information on bucket policies can be found at Adding a bucket policy using the Amazon S3 console.
- **AWS WAF** – Web access control lists (web ACLs) can be configured on AWS WAF to provide fine-grained control over web requests that resources respond to. You can add an IP address or network range to an IP set configured on AWS WAF, and apply match conditions, such as block, to the IP set. This will block web requests to a resource if the IP address or network ranges from the originating traffic match those configured in the IP set rules.

An example of source containment can be seen in the following diagram with an incident response analyst modifying a security group of an Amazon EC2 instance in order to restrict new connections to only certain IP addresses. As stated in the security groups bullet, existing tracked connections won't be shut down as a result of changing security groups.



Source containment example

Technique and access containment

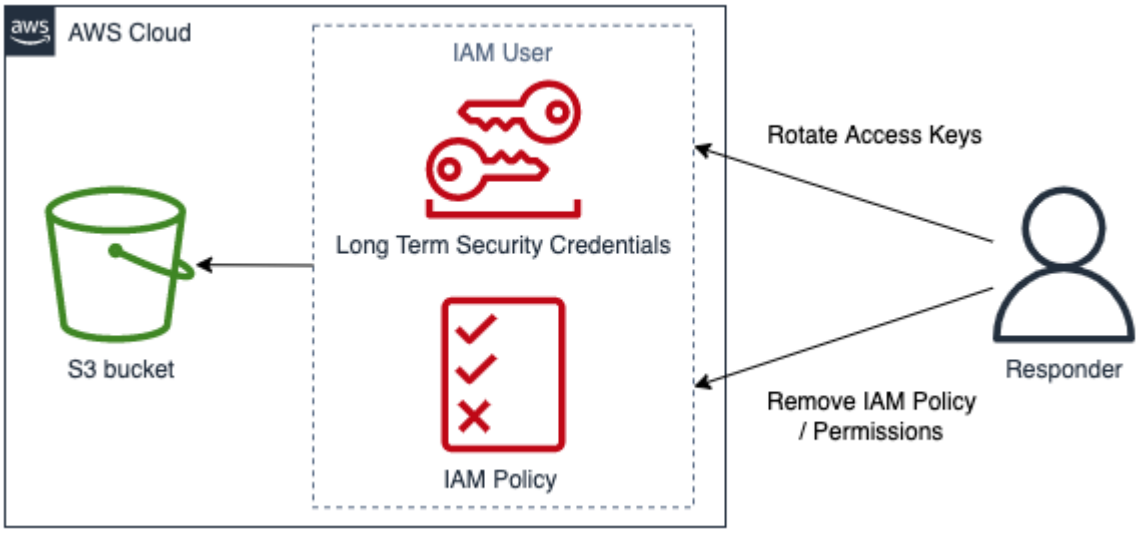
Prevent unauthorized use of a resource by limiting the functions and IAM principals with access to the resource. This includes restricting the permissions of IAM principals that have access to the resource; it also includes temporary security credentials revocation. Examples of technique and access containment using AWS services are highlighted here:

- **Restrict permissions** – Permissions assigned to an IAM principal should follow the Principle of Least Privilege. However, during an active security event, you might need to restrict access to a targeted resource from a specific IAM principal even further. In this case, it is possible to contain access to a resource by removing the permissions from the IAM principal to be contained. This is done with the IAM service and can be applied using AWS Management Console, the AWS CLI, or an AWS SDK.
- **Revoke keys** – IAM access keys are used by IAM principals to access or manage resources. These are long-term static credentials to sign programmatic requests to the AWS CLI or AWS API and begin with the prefix *AKIA* (for additional information, refer to the *Understanding unique ID prefixes* section in IAM identifiers). To contain access for an IAM principal where an IAM access key has been compromised, the access key can be deactivated or deleted. It is important to note the following:
  - An access key can be reactivated after it has been deactivated.
  - An access key is not recoverable once it has been deleted.
  - An IAM principal can have up to two access keys at any given time.

  - Users or applications using the access key will lose access once the key is either deactivated or deleted.

- **Revoke temporary security credentials** – Temporary security credentials can be employed by an organization to control access to AWS resources and begin with the prefix *ASIA* (for additional information, see the *Understanding unique ID prefixes* section in IAM identifiers). Temporary credentials are typically used by IAM roles and do not have to be rotated or explicitly revoked because they have a limited lifetime. In cases where a security event occurs involving a temporary security credential before the temporary security credential expiration, you might need to alter the effective permissions of the existing temporary security credentials. This can be completed using the IAM service within AWS Management Console. Temporary security credentials can also be issued to IAM users (as opposed to IAM roles); however, as of the time of this writing, there is no option to revoke the temporary security credentials for an IAM user within the AWS Management Console. For security events where a user's IAM access key is compromised by an unauthorized user who created temporary security credentials, the temporary security credentials can be revoked using two methods:
  - Attach an inline policy to the IAM user that prevents access based on the security token issue time (refer to the *Denying access to temporary security credentials issued before a specific time* section in Disabling permissions for temporary security credentials for more detail).
  - Delete and recreate the IAM user with the compromised access keys.
- **AWS WAF** - Certain techniques employed by unauthorized users include common malicious traffic patterns, such as requests that contain SQL injection and cross-site scripting (XSS). AWS WAF can be configured to match and deny traffic employing these techniques using the AWS WAF built-in rule statements.

An example of technique and access containment can be seen in the following diagram, with an incident responder rotating access keys or removing an IAM policy to prevent an IAM user from accessing an Amazon S3 bucket.



Technique and access containment example

Destination containment

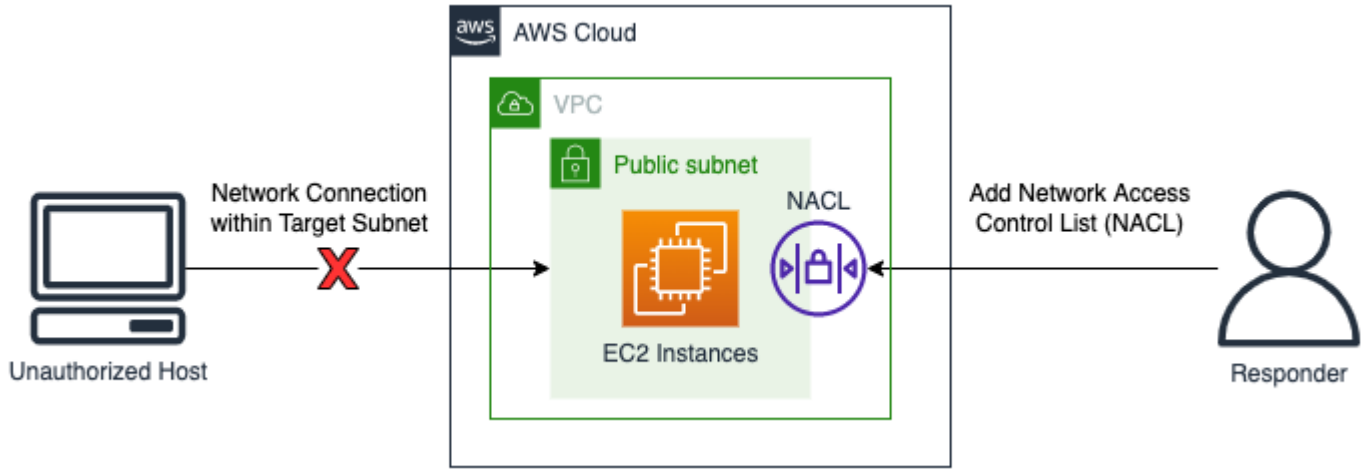
Destination containment is the application of filtering or routing within an environment to prevent access to a targeted host or resource. In some cases, destination containment also involves a form of resiliency to verify that legitimate resources are replicated for availability; resources should be detached from these forms of resiliency for isolation and containment. Examples of destination containment using AWS services include:

- **Network ACLs** – Network ACLs (NACLs) that are configured on subnets that contain AWS resources can have deny rules added. These deny rules can be applied to prevent access to a particular AWS resource; however, applying an NACL will affect every resource on the subnet, not only the resources that are being accessed without authorization. Rules listed within an NACL are processed in top-down order, so the first rule in an existing NACL should be configured to deny unauthorized traffic to the targeted resource and subnet. Alternatively, a completely new NACL can be created with a single deny rule for both inbound and outbound traffic and associated with the subnet containing the targeted resource to prevent access to the subnet using the new NACL.
- **Shutdown** – Shutting down a resource completely can be effective at containing the effects of unauthorized use. Shutting down a resource will also prevent legitimate access for business needs and prevent volatile forensic data from being obtained, so this should be a purposeful decision and should be judged against an organization's security policies.
- **Isolation VPCs** – Isolation VPCs can be used to provide effective containment of resources while providing access to legitimate traffic (such as anti-virus (AV) or EDR solutions that require access

to the internet or an external management console). Isolation VPCs can be preconfigured in advance of a security event to permit valid IP addresses and ports, and targeted resources can immediately be moved into this isolation VPC during an active security event to contain the resource while allowing legitimate traffic to be sent and received by the targeted resource during subsequent phases of incident response. An important aspect of using an isolation VPC is that resources, such as EC2 instances, need to be shut down and relaunched in the new isolation VPC prior to use. Existing EC2 instances cannot be moved to another VPC or another Availability Zone. To do so, follow the steps outlined in How do I move my Amazon EC2 instance to another subnet, Availability Zone, or VPC?

- **Auto scaling groups and load balancers** – AWS resources attached to auto scaling groups and load balancers should be detached and deregistered as part of destination containment procedures. Detachment and deregistration of AWS resources can be performed using the AWS Management Console, AWS CLI, and AWS SDK.

An example of destination containment is demonstrated in the following diagram with an incident response analyst adding an NACL to a subnet in order to block a network connection request from an unauthorized host.



Destination containment example

Summary

Containment is one step of the incident response process and can be manual or automated. The overall containment strategy should align with an organization's security policies and business needs, and verify that negative effects are mitigated as efficiently as possible prior to eradication and recovery.

Eradication

Eradication, in relation to security incident response, is the removal of suspicious or unauthorized resources in efforts to return the account to a known safe state. The eradication strategy depends on multiple factors, which depend on the business requirements for your organization.

The NIST SP 800-61 Computer Security Incident Handling Guide provides several steps for eradication:

1. Identify and mitigate all vulnerabilities that were exploited.
2. Remove malware, inappropriate materials, and other components.
3. If more affected hosts are discovered (for example, new malware infections), repeat the detection and analysis steps to identify all other affected hosts, then contain and eradicate the incident for them.

For AWS resources, this can be further refined through those events detected and analyzed through available logs or automated tooling such as CloudWatch Logs and Amazon GuardDuty. Those events should be the basis to determine which remediations should be performed to properly restore the environment to a known safe state.

The first step of eradication is determining which resources have been affected within the AWS account. This is accomplished through analysis of your available log data sources, resources, and automated tooling.

- Identify unauthorized actions taken by the IAM identities in your account.
- Identify unauthorized access or changes to your account.
- Identify the creation of unauthorized resources or IAM users.
- Identify systems or resources with unauthorized changes.

Once the list of resources is identified, you should assess each to determine the business impact if the resource is deleted or restored. As an example, if a web server is hosting your business application and deleting it would cause down time, then you should consider recovering the resource from verified safe backups or re-launching the system from a clean AMI before deleting the impacted server.

Once you have concluded your business impact analysis, then, using the events from your log analysis, you should go into the accounts and perform the appropriate remediations, such as:

- Rotate or delete keys - this step removes the ability of the actor to continue performing activities within the account.
- Rotate potentially unauthorized IAM user credentials.
- Delete unrecognized or unauthorized resources.

> **Important**
>
> If you must keep resources for your investigation, consider backing up those resources. For example, if you must retain an Amazon EC2 instance for regulatory, compliance, or legal reasons, then create an Amazon EBS snapshot before removing the instance.

- For malware infections, you might need to reach out to an AWS Partner or other vendor. AWS does not offer native tools for malware analysis or removal. If you're using the GuardDuty Malware module for Amazon EBS, then recommendations might be available for provided findings.

Once you have eradicated the identified affected resources, AWS recommends you perform a security review of your account. This can be done using AWS Config rules, using open-source solutions such as Prowler and ScoutSuite, or through other vendors. You should also consider performing vulnerability scans against your public-(internet) facing resources to assess residual risk.

Eradication is one step of the incident response process and can be manual or automated, depending on the incident and affected resources. The overall strategy should align with an organization's security policies and business needs, and verify that negative effects are mitigated as inappropriate resources or configurations are removed.

Recovery

Recovery is the process of restoring systems to a known safe state, validating that backups are safe or unaffected by the incident prior to restoration, testing to verify that the systems are working properly post-restoration, and addressing vulnerabilities associated with the security event.

The order of recovery depends on your organization's requirements. As part of the process of recovery, you should perform a business impact analysis to determine, at minimum:

- Business or dependency priorities

- The restoration plan
- Authentication and authorization

The NIST SP 800-61 Computer Security Incident Handling Guide provides several steps to recover systems, including:

- Restoring systems from clean backups.
  - Verify that backups are evaluated before restoring to systems to make sure that the infection is not present and to prevent resurgence of the security event.

    Backups should be evaluated on a regular basis as part of disaster recovery testing to verify that the backup mechanism is working properly and the data integrity meets recovery point objectives.
  - If possible, use backups from before the first event timestamp identified as part of root cause analysis.
- Rebuilding systems from scratch, including redeploying from trusted source using automation, sometime in a new AWS account.
- Replacing compromised files with clean versions.

  You should exercise great caution when doing this. You must be absolutely certain the file you are recovering is known safe and unaffected by the incident
- Installing patches.
- Changing passwords.
  - This includes passwords for IAM principals that might have been abused.
  - If possible, we recommend using roles for IAM principals and federation as part of a least privilege strategy.
- Tightening network perimeter security (firewall rulesets, boundary router access control lists).

Once the resources have been recovered, it is important to capture lessons learned to update incident response policies, procedures, and guides.

In summary, it is imperative to implement a recovery process that facilitates a return to known safe operations. Recovery can take a long time and requires a close linkage with containment strategies to balance business impact against risk of reinfection. Recovery procedures should include steps for

restoring resources and services, IAM principals, and performing a security review of the account to assess residual risk.

Conclusion

Each operations phase has unique goals, techniques, methodologies, and strategies. Table 4 summarizes these phases and some of the techniques and methodologies covered in this section.

Table 4 – Operations phases: Goals, techniques, and methodologies

Phase	Goal	Techniques and methodologies
Detection	Identify a potential security event.	• Security controls for detection • Behavior and rule-based detection • People-based detection
Analysis	Determine if the security event is an incident and assess the scope of the incident.	• Validate and scope alert • Query logs • Threat intelligence • Automation
Containment	Minimize and limit the impact of the security event.	• Source containment • Technique and access containment • Destination containment
Eradication	Remove unauthorized resources or artifacts related to the security event.	• Compromised or unauthorized credential rotation or deletion • Unauthorized resource deletion • Malware removal • Security scans

Phase	Goal	Techniques and methodologies
Recovery	Restore systems to a known good state and monitor these systems to ensure the threat does not return.	• System restoration from backups • Systems rebuilt from scratch • Compromised files replaced with clean versions

Post-incident activity

The threat landscape is constantly changing and it is important to be equally dynamic in your organization's ability to effectively protect your environments. Key to continuous improvement is iterating on the outcomes of your incidents and simulations in order to improve your capabilities to effectively detect, respond to, and investigate possible security incidents, reducing your possible vulnerabilities, time to response, and return to safe operations. The following mechanisms can help you verify that your organization remains prepared with the latest capabilities and knowledge to effectively respond, no matter the situation.

Establish a framework for learning from incidents

Implementing a *lessons learned* framework and methodology will not only help to improve incident response capabilities, but also help to prevent the incident from recurring. By learning from each incident, you can help to avoid repeating the same mistakes, exposures, or misconfigurations, not only improving your security posture, but also minimizing time lost to preventable situations.

It's important to implement a lessons learned framework that establishes and achieves, at a high level, the following points:

- When is a lessons learned held?
- What is involved in the lessons learned process?
- How is a lessons learned performed?
- Who is involved in the process and how?
- How will areas of improvement be identified?
- How will you ensure improvements are effectively tracked and implemented?

Aside from these high-level outcomes listed, it is important to make sure that you ask the right questions to derive the most value (information that leads to actionable improvements) from the process. Consider these questions to help get you started in fostering your lessons learned discussions:

- What was the incident?
- When was the incident first identified?
- How was it identified?

- What systems alerted on the activity?
- What systems, services, and data were involved?
- What specifically occurred?
- What worked well?
- What didn't work well?
- Which process or procedures failed or failed to scale to respond to the incident?
- What can be improved within the following areas:
  - **People**
    - Were the people who were needed to be contacted actually available and was the contact list up to date?
    - Were people missing training or capabilities needed to effectively respond and investigate the incident?
    - Were the appropriate resources ready and available?
  - **Process**
    - Were processes and procedures followed?
    - Were processes and procedures documented and available for this (type of) incident?
    - Were required processes and procedures missing?
    - Were the responders able to gain timely access to the required information to respond to the issue?
  - **Technology**
    - Did existing alerting systems effectively identify and alert on the activity?
    - Do existing alerts need improvement or new alerts need to be built for this (type of) incident?
    - Did existing tooling allow for effective investigation (search/analysis) of the incident?
- What can be done to help identify this (type of) incident sooner?
- What can be done to help prevent this (type of) incident from occurring again?
- Who owns the improvement plan and how will you test that it has been implemented?
- What is the timeline for the additional monitoring/preventative controls/process to be implemented and tested?

This list isn’t all-inclusive; it is intended to serve as a starting point for identifying what the organization and business needs are and how you can analyze them in order to most effectively

learn from incidents and continuously improve your security posture. Most important is getting started by incorporating lessons learned as a standard part of your incident response process, documentation, and expectations across the stakeholders.

Establish metrics for success

Metrics are necessary to effectively measure, assess, and improve your incident response capabilities. Without metrics, there is no reference against which to accurately measure or even identify how well your organization is (or is not) performing. There are a few metrics common to incident response that are a good starting point for an organization looking to establish expectations and references for working toward operational excellence.

Mean time to detect

Mean time to detect is the average time it takes to discover a possible security incident. Specifically, this is the time between the occurrence of the first indicator of compromise and the initial identification or alert.

You can use this metric to track how effective your detection and alerting systems are performing. Effective detection and alerting mechanisms are key to verifying that possible security incidents don't linger within your environments.

The higher the mean time to detect, the greater the need to build additional or more effective alerts and mechanisms to identify and discover possible security incidents. The lower the mean time to detect, the better your detection and alerting mechanisms are functioning.

Mean time to acknowledge

Mean time to acknowledge is the average time it takes to acknowledge and prioritize a possible security incident. Specifically, this is the time between the generation of an alert and a member of your SOC or incident response staff identifying and prioritizing the alert for processing.

You can use this metric to track how well your team is processing and prioritizing alerts. If your team is unable to effectively identify and prioritize alerts, then responses will be delayed and ineffective.

The higher the mean time to acknowledge, the greater the need to verify that your team is both properly resourced and trained to quickly acknowledge and prioritize a possible security incident for response. The lower the mean time to acknowledge, the better your team is at responding to security alerts, showing that they are effectively prepared and able to prioritize them well.

Mean time to respond

Mean time to respond is the average time it takes to begin the initial response to a possible security incident. Specifically, this is the time between initial alert or discovery of a possible security incident and first actions taken to respond. This is similar to mean time to acknowledge, but is the measurement of specific responsive actions (for example, acquire system data, contain the system) compared to simple recognition or acknowledgement of the situation.

You can use this metric to track your preparedness to respond to security incidents. As mentioned, preparation is key to effective response. Refer to the *Preparation* section of this document.

The higher the mean time to respond, the greater the need to verify that your team is both properly trained on how to respond so that response processes are effectively documented and utilized. The lower the mean time to respond, the better your team is at identifying an appropriate response to identified alerts and performing the required responsive actions to begin the journey back to safe operations.

Mean time to contain

Mean time to contain is the average time it takes to contain a possible security incident. Specifically, this is the time between initial alert or discovery of a possible security incident and the completion of responsive actions that effectively prevents the attacker or compromised systems from doing further harm.

You can use this metric to track how well your team is able to mitigate or contain possible security incidents. Inability to quickly and effectively contain possible security incidents increases the impact, scope, and exposure to possible further compromise.

The higher the mean time to contain, the greater the need to build both knowledge and capabilities to quickly and effectively mitigate and contain the security incidents you are experiencing. The lower the mean time to contain, the better your team is at understanding and employing the necessary measures to mitigate and contain identified threats to reduce impact, scope, and risk to the business.

Mean time to recover

Mean time to recover is the average time it takes to fully return so safe operations from a possible security incident. Specifically, this is the time between initial alert or discovery of a possible

security incident and when the business is back to operating normally and safely without being affected by the incident.

You can use this metric to track how effective your teams are at returning systems, accounts, and environments back to safe operations after a security incident. Inability to return to safe operations swiftly or effectively can not only have an impact on security but can also increase impact and expense to the business and its operations.

The higher the mean time to recover, the greater the need to prepare your teams and environments to have the appropriate mechanisms (for example, failover processes and CI/CD pipelines to safe redeploy clean systems) to minimize the impact of security incidents to operations and the business. The lower the mean time to recover, the more effective your teams are at minimizing the impact of security incidents on your operations and business.

Attacker dwell time

Attacker dwell time is the average time an unauthorized user has access to a systems or environment. This is similar to mean time to contain, except the time frame begins with the initial time the attacker gained access to the system or environments, which might be earlier than the initial alert or discovery.

You can use this metric to track how well many of your systems and mechanisms are all working together to reduce the amount of time, access, and opportunity an attacker or threat has to impact your environment. Reducing attacker dwell time should be a top priority for your teams and business.

The higher the attacker dwell time, the greater the need to identify which parts of the incident response process need improvement to ensure your teams' abilities to minimize the impact and scope of threats or attacks in your environments. The lower the attacker dwell time, the better your teams are at minimizing the time and opportunity that a threat or attacker has within your environments, ultimately reducing the risk and impact to your operations and business.

Metrics summary

Establishing and tracking metrics for incident response allows you to effectively measure, assess, and improve your incident response capabilities. To achieve this, there are number of common incident response metrics that were highlighted in this section. Table 5 summarizes these metrics.

Table 5 – Incident response metrics

Metric	Description
Mean time to detect	Average time it takes to discover a possible security incident
Mean time to acknowledge	Average time it takes to acknowledge (and prioritize) a possible security incident
Mean time to respond	Average time it takes to begin the initial response to a possible security incident
Mean time to contain	Average time it takes to contain a possible security incident
Mean time to recover	Average time it takes to fully return so safe operations from a possible security incident
Attacker dwell time	Average time an attacker has access to a systems or environment

Use indicators of compromise (IOCs)

An *indicator of compromise* (IOC) is an artifact observed in or on a network, system, or environment that can (with a high level of confidence) identify malicious activity or a security incident. IOCs can exist in a variety of forms, including IP addresses, domains, network-level artifacts such as TCP flags or payloads, system or host-level artifacts such as executables, file names and hashes, log file entries, or registry entries, and more. They can also be a combination of items or activities, such as the existence of specific items or artifacts on a system (a certain file or set of files and registry items), actions performed in certain order (a login to a system from a certain IP followed by specific anomalous commands), or network activity (anomalous inbound or outbound traffic to or from certain domains) that can indicate a specific threat, attack, or attacker methodology.

As you work to iteratively improve your incident response program, you should implement a framework to collect, manage, and utilize IOCs as a mechanism to continuously build and improve detections and alerting and improve the speed and efficacy of investigations. You can start by incorporating the collection and management of IOCs into the analysis and investigation phases of your incident response processes. By proactively identifying, collecting, and storing IOCs as a

standard part of your process, you can build a repository of data (as part of a more comprehensive threat intelligence program) that in turn can be used to improve existing detections and alerts, build additional detections and alerts, identify where and when an artifact was seen before, build and reference documentation of how investigations were previously done involving matching IOCs, and more.

Continue education and training

Education and training are both evolving and continual efforts that should be purposefully pursued and maintained. There are a variety of mechanisms to verify that your team is maintaining awareness, knowledge, and capabilities commensurate with the evolving state of technology as well as the threat landscape.

One mechanism is to employ continuing education as a standard part of your teams' goals and operations. As mentioned in the Preparation section, your incident response staff and stakeholders must be effectively trained on detecting, responding to, and investigating incidents within AWS. However, education isn't a "one and done" effort. Education must be continuously pursued to verify that your team maintains awareness of the latest technological advances, updates, and improvements that can be leveraged to improve the efficacy and efficiency of response, as well as additions or updates to data that can be leveraged for improving investigation and analysis.

Another mechanism is to verify that simulations are performed on a regular basis (for example, quarterly) and focused on specific outcomes for the business. Refer to the the section called "Run regular simulations" section of this document.

Though running initial tabletop exercise exercises are an excellent way to generate an initial baseline for improvement, continuous testing is key to sustained improvements and maintaining an up-to-date and accurate reflection of the current state of operations. Testing against the latest and most critical security situations and the most important or newest capabilities for response, and incorporating the lessons learned back into education, operations, and processes/procedures will verify that you are able to continuously improve your response processes and program as a whole.

Conclusion

As you continue your cloud journey, it is important for you to consider the fundamental security incident response concepts for your AWS environment. You can combine the available controls, cloud capabilities, and remediation options to help you improve the security of your cloud environment. You can also start small and iterate as you adopt automation capabilities that improve your response speed, so you are better prepared when security events occur.

Contributors

Current and past contributors to this document include:

- Anna McAbee, Senior Security Solutions Architect, Amazon Web Services
- Freddy Kasprzykowski, Senior Security Consultant, Amazon Web Services
- Jason Hurst, Senior Security Consultant, Amazon Web Services
- Jonathon Poling, Principal Security Consultant, Amazon Web Services
- Josh Du Lac, Senior Manager, Security Solutions Architecture, Amazon Web Services
- Paco Hope, Principal Security Consultant, Amazon Web Services
- Ryan Tick, Senior Security Engineer, Amazon Web Services
- Steve de Vera, Senior Security Consultant, Amazon Web Services

Appendix A: Cloud capability definitions

AWS offers over 200 cloud services and thousands of features. Many of these provide native detective, preventative, and responsive capabilities, and others can be used to architect custom security solutions. This section includes a subset of those services that are most relevant to incident response in the cloud.

Topics

Logging and events

AWS CloudTrail – AWS CloudTrail service enabling governance, compliance, operational auditing, and risk auditing of AWS accounts. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across AWS services. CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. This event history simplifies security analysis, resource change tracking, and troubleshooting. CloudTrail logs two different types of AWS API actions:

- **CloudTrail management events** (also known as control plane operations) show management operations that are performed on resources in your AWS account. This includes actions such as creating an Amazon S3 bucket and setting up logging.
- **CloudTrail data events** (also known as data plane operations) show the resource operations performed on or within a resource in your AWS account. These operations are often high-volume activities. This includes actions such as Amazon S3 object-level API activity (for example, `GetObject`, `DeleteObject`, and `PutObject` API operations) and Lambda function invocation activity.

Amazon CloudWatch Events – Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in AWS resources, or when API calls are published by AWS

CloudTrail. Using simple rules that you can quickly set up, you can match events and route them to one or more target functions or streams. CloudWatch Events becomes aware of operational changes as they occur. CloudWatch Events can respond to these operational changes and take corrective action as necessary, by sending messages to respond to the environment, activating functions, making changes, and capturing state information. Some security services, such as Amazon GuardDuty, produce their output in the form of CloudWatch Events. Many security services also provide an option to send their outputs to Amazon S3.

AWS Config – AWS Config is a service enabling customers assess, audit, and evaluate the configurations of your AWS resources. AWS Config continuously monitors and records your AWS resource configurations and enables you to automate the evaluation of recorded configurations against desired configurations. With AWS Config, customers can review changes in configurations and relationships between AWS resources, manually or automatically, detailed resource configuration history, and determine overall compliance against the configurations specified in customer's guidelines. This enables simplification of compliance auditing, security analysis, change management, and operational troubleshooting.

Amazon S3 access logs – If sensitive information is stored in an Amazon S3 bucket, customers can enable Amazon S3 access logs to record every upload, download, and modification to that data. This log is separate from, and in addition to, the CloudTrail logs that record changes to the bucket itself (such as changing access policies and lifecycle policies). It's worth noting that access log records are delivered on a best effort basis. Most requests for a bucket that is properly configured for logging result in a delivered log record. The completeness and timeliness of server logging is not guaranteed.

Amazon CloudWatch Logs – Customers can use Amazon CloudWatch Logs to monitor, store, and access log files originating from operating systems, applications, and other sources running in Amazon EC2 instances with a CloudWatch Logs agent. CloudWatch Logs can be a destination for AWS CloudTrail, Route 53 DNS Queries, VPC Flow Logs, Lambda functions, and others. Customers can then retrieve the associated log data from CloudWatch Logs.

Amazon VPC Flow Logs – VPC Flow Logs enables customers to capture information about IP traffic going to and from network interfaces in VPCs. After enabling flow logs, they can be streamed to Amazon CloudWatch Logs and Amazon S3. VPC Flow Logs helps customers with a number of tasks such as troubleshooting why specific traffic is not reaching an instance, diagnosing overly restrictive security group rules, and using it as a security tool to monitor the traffic to EC2 instances. Use the most current version of VPC flow logging to get the most robust fields.

AWS WAF Logs – AWS WAF supports full logging of all web requests inspected by the service. Customers can store these in Amazon S3 to fulfil compliance and auditing requirements, as well as debugging and forensics. These logs help customers determine root cause of initiated rules and blocked web requests. Logs can be integrated with third-party SIEM and log analysis tools.

Route 53 Resolver query logs – Route 53 Resolver query logs will let you log all DNS queries made by resources within Amazon Virtual Private Cloud (Amazon VPC). Whether it's an Amazon EC2 instance, an AWS Lambda function, or a container, if it lives in your Amazon VPC and makes a DNS query, then this feature will log it; you are then able to explore and better understand how your applications are operating.

Other AWS logs – AWS continuously releases service features and capabilities for customers with new logging and monitoring capabilities. For information about features available for each AWS service, refer to our public documentation.

Visibility and alerting

AWS Security Hub – AWS Security Hub provides customers with a comprehensive view of high-priority security alerts and compliance statuses across AWS accounts. Security Hub aggregates, organizes, and prioritizes findings from AWS services such as Amazon GuardDuty, Amazon Inspector, Amazon Macie, and AWS Partner solutions. Findings are visually summarized on integrated dashboards with actionable graphs and tables. You can also continuously monitor your environment using automated compliance checks based on the AWS best practices and industry standards your organization follows.

Amazon GuardDuty – Amazon GuardDuty is a managed threat detection service continuously monitoring malicious or unauthorized behavior to help customers protect AWS accounts and workloads. It monitors activity such as unusual API calls or potentially unauthorized deployments indicating possible account or resource compromise of Amazon EC2 instances, Amazon S3 buckets, or reconnaissance by bad actors.

GuardDuty identifies suspected bad actors through integrated threat intelligence feeds using machine learning to detect anomalies in account and workload activity. When a potential threat is detected, the service delivers a detailed security alert to the GuardDuty console and CloudWatch Events. This makes alerts actionable and simple to integrate into existing event management and workflow systems.

GuardDuty also offers two add-ons to monitor for threats with specific services: Amazon GuardDuty for Amazon S3 protection and Amazon GuardDuty for Amazon EKS protection. Amazon

S3 protection enables GuardDuty to monitor object-level API operations to identify potential security risks for data within Amazon S3 buckets. Kubernetes protection enables GuardDuty to detect suspicious activities and potential compromises of Kubernetes clusters within Amazon EKS.

Amazon Macie – Amazon Macie is an AI-powered security service that helps prevent data loss by automatically discovering, classifying, and protecting sensitive data stored in AWS. Macie uses machine learning (ML) to recognize sensitive data such as personally identifiable information (PII) or intellectual property, assign a business value, and provide visibility into where this data is stored and how it is being used in your organization. Amazon Macie continuously monitors data access activity for anomalies, and delivers alerts when it detects a risk of unauthorized access or inadvertent data leaks.

AWS Config Rules – An AWS Config rule represents the preferred configurations for a resource and is evaluated against configuration changes on the relevant resources, as recorded by AWS Config. You can see the results of evaluating a rule against the configuration of a resource on a dashboard. Using AWS Config rules, you can assess your overall compliance and risk status from a configuration perspective, view compliance trends over time, and find which configuration change caused a resource to be out of compliance with a rule.

AWS Trusted Advisor – AWS Trusted Advisor is an online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment. Trusted Advisor provides real time guidance to help you provision your resources by following AWS best practices. The full set of Trusted Advisor checks, including CloudWatch Events integration, is available to Business and Enterprise support plan customers.

Amazon CloudWatch – Amazon CloudWatch is a monitoring service for AWS Cloud resources and the applications you run on AWS. You can use CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. CloudWatch can monitor AWS resources, such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to get system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react accordingly and keep your application running smoothly.

Amazon Inspector – Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for vulnerabilities or deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed list of security findings

prioritized by level of severity. These findings can be reviewed directly or as part of detailed assessment reports which are available through the Amazon Inspector console or API.

Amazon Detective – Amazon Detective is a security service that automatically collects log data from your AWS resources and uses machine learning, statistical analysis, and graph theory to build a linked set of data that enables you to conduct faster and more efficient security investigations. Detective can analyze trillions of events from multiple data sources such as VPC Flow Logs, CloudTrail, and GuardDuty, and automatically creates a unified, interactive view of your resources, users, and the interactions between them over time. With this unified view, you can visualize all the details and context in one place to identify the underlying reasons for the findings, drill down into relevant historical activities, and quickly determine the root cause.

Automation

AWS Lambda – AWS Lambda is a serverless compute service that runs your code in response to events, and automatically manages the underlying compute resources for you. You can use Lambda to extend other AWS services with custom logic, or create your own backend services that operate at AWS scale, performance, and security. Lambda runs your code on high-availability compute infrastructure and performs the administration of the compute resources for you. This includes server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment, and code monitoring and logging. All you have to do is supply the code.

AWS Step Functions – AWS Step Functions makes it simple to coordinate the components of distributed applications and microservices using visual workflows. Step Functions provides a graphical console for you to arrange and visualize the components of your application as a series of steps. This makes it simple to build and run multistep applications. Step Functions automatically starts and tracks each step, and retries when there are errors, so your application runs in order and as expected.

Step Functions logs the state of each step, so when things do go wrong, you can diagnose and debug problems quickly. You can change and add steps without writing code, so you can evolve your application and innovate faster. AWS Step Functions is part of AWS Serverless, and makes it simple to orchestrate AWS Lambda functions for serverless applications. You can also use Step Functions for microservices orchestration using compute resources such as Amazon EC2 and Amazon ECS.

AWS Systems Manager – AWS Systems Manager gives you visibility and control of your infrastructure on AWS. Systems Manager provides a unified user interface so you can view

operational data from multiple AWS services, and enables you to automate operational tasks across your AWS resources. With Systems Manager, you can group resources by application, view operational data for monitoring and troubleshooting, and act on your groups of resources. Systems Manager can keep your instances in their defined state, perform on-demand changes, such as updating applications or running shell scripts, and perform other automation and patching tasks.

Secure storage

Amazon Simple Storage Service – Amazon S3 is object storage built to store and retrieve any amount of data from anywhere. It is designed to deliver 99.999999999% durability, and stores data for millions of applications used by market leaders in every industry. Amazon S3 provides comprehensive security and is designed to help you meet your regulatory requirements. It gives customers flexibility in the methods they use to manage data for cost optimization, access control, and compliance. Amazon S3 provides query-in-place functionality, which enables you to run powerful analytics directly on your data at rest in Amazon S3. Amazon S3 is a highly supported cloud storage service, with integration from one of the largest communities of third-party solutions, systems integrator partners, and other AWS services.

Amazon S3 Glacier – Amazon S3 Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup. It is designed to deliver 99.999999999% durability, provides comprehensive security, and is designed to help you meet your regulatory requirements. S3 Glacier provides query-in-place functionality, which enables you to run powerful analytics directly on your archive data at rest. To keep costs low yet suitable for varying retrieval needs, S3 Glacier provides three options for access to archives, from a few minutes to several hours.

Custom

The aforementioned services and features are not an exhaustive list. AWS is continuously adding new capabilities. For more information, we encourage you to review the What's New at AWS and AWS Cloud Security pages. In addition to the security services that AWS offers as native cloud services, you might be interested in building your own capabilities on top of AWS services.

Although we recommend enabling a base set of security services within your accounts, such as AWS CloudTrail, Amazon GuardDuty, and Amazon Macie, you might eventually want to extend these capabilities to derive additional value from your log assets. There are a number of partner tools available, such as those listed in our APN Security Competency program. You might also want to

write your own queries to search your logs. With the extensive number of managed services that AWS offers, this has never been easier. There are many additional AWS services that can assist you with investigation that are outside the scope of this paper, such as Amazon Athena, Amazon OpenSearch Service, Amazon QuickSight, Amazon Machine Learning, and Amazon EMR.

Appendix B: AWS incident response resources

AWS publishes resources to assist customers with developing incident response capabilities. Most example code and procedures can be found at the AWS external GitHub public repository. Following are some resources that provide examples of how to perform incident response.

Playbook resources

- Framework for Incident Response Playbooks - An example framework for customers to create, develop, and integrate security playbooks in preparation for potential attack scenarios when using AWS services.
- Develop your own Incident Response Playbooks - This workshop is designed to help you get familiar with developing incident response playbooks for AWS.
- Incident Response Playbook Samples - Playbooks covering common scenarios faced by AWS customers.
- Building an AWS incident response runbook using Jupyter playbooks and CloudTrail Lake - This workshop guides you through building an incident response playbook for your AWS environment using Jupyter notebooks and CloudTrail Lake.

Forensic resources

- Automated Incident Response and Forensics Framework – This framework and solution provides a standard digital forensic process, consisting of the following phases: containment, acquisition, examination, and analysis. It leverages AWS Lambda functions to trigger the incident response process in an automated repeatable way. It provides segregation of accounts to operate the automation steps, store artifacts and create forensic environments.
- Automated Forensics Orchestrator for Amazon EC2 – This implementation guide provides a self-service solution to capture and examine data from EC2 instances and attached volumes for forensic analysis in the event of a potential security issue being detected. There is an AWS CloudFormation template to deploy the solution.
- How to automate forensic disk collection in AWS – This AWS blog details how to set up an automation workflow to capture the disk evidence for analysis in order to determine the scope and the impact of potential security incidents. There is also an AWS CloudFormation template included to deploy the solution.

Document revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Minor updates	Several changes and additions to improve guidance.	May 11, 2023
Major revision	Nearly all content replaced to improve alignment with industry standards, such as NIST. Additional prescriptive guidance on how to prepare for and respond to security events in an AWS environme nt.	January 1, 2023
Minor updates	Bug fixes and numerous minor changes throughout.	April 1, 2022
Minor updates	Bug fixes and numerous minor changes throughout.	June 2, 2021
Minor update	Corrected broken links.	March 5, 2021
Whitepaper updated	Corrected broken links and numerous text changes to improve readability.	November 23, 2020
Minor update	Fixed link to "Incident Response with AWS Console and CLI".	June 30, 2020
Whitepaper updated	Updated for new security services, threat intelligence, shared responsibility for containers, automation, and	June 11, 2020

	CCPA. Added appendices with sample decision tree and runbook.	
Initial publication	Whitepaper first published	June 1, 2019

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.