AWS Security Incident Response User Guide

Version December 1, 2024

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Security Incident Response User Guide:

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is AWS Security Incident Response?

AWS Security Incident Response helps you quickly prepare for, respond to, and receive guidance to help recover from security incidents. This includes incidents like account takeovers, data breaches, and ransomware attacks.

AWS Security Incident Response triages findings, escalates security events, and manages cases that require your immediate attention. Additionally, you have access to the AWS Customer Incident Response Team (CIRT), who will investigate impacted resources.

> **Note**
>
> There is no guarantee impacted resources can be recovered. We recommend establishing and maintaining backups for resources that could impact your business requirements.

AWS Security Incident Response works with other AWS Detection and Response services, guiding you through the entire incident lifecycle – from detection to recovery.

Contents

Supported configurations

AWS Security Incident Response supports the following language and region configurations:

- Language: AWS Security Incident Response is available in English.
- Supported AWS Regions:

  AWS Security Incident Response is available in a subset of AWS Regions. In these supported Regions, you create a membership, create and view cases, and access the dashboard.

  - US East (Ohio)
  - US West (Oregon)
  - US East (Virginia)
  - EU (Frankfurt)

- EU (Ireland)
- EU (London)
- EU (Stockholm)
- Asia Pacific (Singapore)
- Asia Pacific (Seoul)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)

When you enable the monitoring and investigation feature, AWS Security Incident Response monitors Amazon GuardDuty findings from all active commercial AWS Regions. As a security best practice, AWS recommends enabling GuardDuty in all supported AWS Regions. This configuration allows GuardDuty to generate findings about unauthorized or unusual activity, even in AWS Regions where you don't actively deploy resources. By doing so, you enhance your overall security posture and maintain comprehensive threat detection coverage across your AWS environment.

> **Note**
>
> Amazon GuardDuty reports findings for configured regions. If you choose not to enable the service in a specific region, then alerts will not be available.

Feature Summary

Monitoring and investigation

AWS Security Incident Response rapidly reviews security alerts from Amazon GuardDuty and third-party integrations with AWS Security Hub, reducing the number your team needs to analyze. It configures suppression rules based on your environment to reduce low-priority alerts you need to triage and investigate.

Streamline incident response

Scale and execute incident response within minutes with relevant stakeholders, third-party services, and tools.

Self-service security solutions

AWS Security Incident Response provides APIs to integrate and allow you to build your own customized security solutions.

Dashboard for visibility

Monitor and measure incident response readiness.

Security posture

Access AWS best practices and vetted tools for security assessment and rapid incident response investigation.

Expedited assistance

Connect with AWS's Customer Incident Response Team (CIRT) to investigate, contain, and receive guidance on ways to recover from security events.

Preparedness and readiness

Implement streamlined notification by setting up your Incident Response team that triggers alerts to designated individuals or groups, with predefined permission policies.

Concepts and Terminology

The following terms and concepts are important for understanding the AWS Security Incident Response service and how it works.

Scope: AWS Security Incident Response aligns with the National Institute of Standards and Techology (NIST) 800-61 Computer Security Incident Handling Guide, providing a consistent approach to security event management as related to industry best practices.

Analysis: The detailed investigation and examination of a security event to understand its scope, impact, and root cause.

AWS Security Incident Response service portal: A self-service portal for you to initiate and manage security event cases. Ongoing communication and reporting facilitated through the ticketing system, automated notifications, and direct engagement with the service team.

Communication: The ongoing dialog and information sharing between the AWS Security Incident Response team and the customer during the incident response process.

Containment, Eradication, and Recovery: The prevention of additional unauthorized activity (containment), coupled with the removal of unauthorized resources and the original vulnerability (eradication), and recovering resources to get back to business as normal.

Continuous Improvement: AWS Security Incident Response incorporates feedback and lessons learned from prior engagements to enhance its detection capabilities, investigative processes, and remediation actions. AWS Security Incident Response also stays up-to-date with the latest security threats and best practices to address evolving security challenges.

Cybersecurity Event: Any observable occurrence in a system or network that violates or threatens to violate security policies, acceptable use policies, or standard security practices.

Incident Response Team: A group of individuals who provide support during active security events. For AWS supported cases, this is the AWS Customer Incident Response Team (CIRT).

Incident Response Workflow: The defined sequence of steps and activities involved in the end-to-end management of a security event, aligned with the NIST 800-61 standard.

Investigative Tooling: AWS Security Incident Response tools and service-linked roles used to review the operational health of your account and resources.

Lessons Learned: The review and documentation of a security event response to identify areas for improvement and inform future incident response planning.

Monitoring and Investigation: AWS Security Incident Response rapidly reviews security alerts from Amazon GuardDuty, bringing to the forefront the most important alerts your team needs to analyze. It configures suppression rules based on the specifics of your environment to prevent unnecessary alerts.

Preparation: The activities undertaken to get an organization ready to effectively respond to and manage security events, such as developing incident response plans and testing procedures.

Reporting and Communication: The processes used to keep you informed throughout the incident response process, including automated notifications, call bridges, and the delivery of investigation artifacts. AWS Security Incident Response provides a single, centralized dashboard in the AWS Management Console to manage all your AWS Security Incident Response efforts.

Responder Generated Intelligence: indicators of compromise; tactics, techniques, and procedures; and associated patterns observed by AWS CIRT investigations.

Security Event Expertise: The specialized knowledge and skills required to effectively respond to and manage security events, particularly in the context of the AWS cloud.

Shared Responsibility Model: The division of security responsibilities between AWS and the customer, where AWS is responsible for security of the cloud, and the customer is responsible for security in the cloud.

Threat Intelligence: Internal and external data feeds containing details of unauthorized activity to help identify and respond to evolving security threats.

Ticketing System: A dedicated case management platform that allows you to onboard and manage security event cases, add attachments, and track the incident response lifecycle.

Triage: The initial assessment and prioritization of a security event to determine the appropriate response and next steps.

Workflow: The defined sequence of steps and activities involved in the end-to-end management of a security event.

Getting Started

Contents

Select a membership account

You have two options for selecting your AWS Security Incident Response membership account using AWS Organizations. You can either create a membership in the Organizations management account or in an Organizations delegated administrator account.

Use the delegated administrator account: AWS Security Incident Response administrative tasks and case management are located in the delegated administrator account. We recommend using the same delegated administrator you've set for other AWS security and compliance services. Provide the 12-digit delegated administrator account ID and then log in to that account to proceed.

Use the currently logged in account: Selecting this account means the current account will be central membership account for your AWS Security Incident Response membership. Individuals within your organization will need to access the service through this account to create, access, and manage active and resolved cases.

Ensure you have sufficient permissions to administer AWS Security Incident Response.

Refer to Adding and removing IAM identity permissions for specific steps to add permissions.

Refer to AWS Security Incident Response managed policies.

To verify IAM permissions, you can follow these steps:

- *Check the IAM Policy:* Review the IAM policy attached to your user, group, or role to ensure it grants the necessary permissions. You can do this by navigating to the https://console.aws.amazon.com/iam/, select the `Users` option, choose the specific user, and then on their summary page, go to the `Permissions` tab where you can see a list of all attached policies; you can expand each policy row to view its details.

- *Test the Permissions:* Try to perform the action you need to verify the permissions. For example, if you need to access a case, try to `ListCases`. If you don't have the necessary permissions, you'll receive an error message.
- *Use the AWS CLI or SDK:* You can use the AWS Command Line Interface Command Line Interface (CLI) or an AWS SDK in your preferred programming language to test the permissions. For example, with the AWS Command Line Interface, you can run the `aws sts get-caller-identity` command to verify your current user permissions.
- *Check the AWS CloudTrail logs:* Review the CloudTrail logs to see if the actions you're trying to perform are being logged. This can help you identify any permission issues.
- *Use the IAM policy simulator:* The IAM policy simulator is a tool that allows you to test IAM policies and see the effect they have on your permissions.

> **Note**
>
> The specific steps may vary depending on the AWS service and the actions you're trying to perform.

Setup membership details

- Select an AWS Region where your membership and cases will be stored.

> **Warning**
>
> You can't change the default AWS Region after initial membership registration.

- You may optionally select a name for this membership.
- You have to supply a Primary and Secondary contact as part of the create membership workflow. These contacts are automatically included as part of you incident response team. At least two contacts must exist for a single membership which also ensures a minimum of two contacts are included in the incident response team.
- Define optional tags for your membership. Tags help you to track AWS costs and search for resources.

Associate accounts with AWS Organizations

Your membership entitles coverage on all linked AWS accounts in AWS Organizations. Associated accounts will automatically update as accounts get added or removed from your organization.

Setup proactive response and alert triaging workflows

Proactive response and alert triaging workflow is an optional feature to enable within your organization for monitoring enabled security services. Select the toggle next to the feature to enable.

If you experience any onboarding issues, then please create an AWS Support case for additional assistance. Make sure to include details including the AWS account ID and any errors you may have seen during the setup process.

Proactive response and alert triaging: AWS Security Incident Response monitors and investigates alerts generated from Amazon GuardDuty and Security Hub integrations. To use this feature, Amazon GuardDuty must be enabled. AWS Security Incident Response triages low-priority alerts with service automation so your team can focus on the most critical issues. For additional information on how AWS Security Incident Response works with Amazon GuardDuty and AWS Security Hub, please review the Detect and Analyze section of the user guide.

This feature enables AWS Security Incident Response to monitor and investigate findings across all accounts and active supported AWS Regions in your organization. To facilitate this functionality, AWS Security Incident Response automatically creates a service-linked role in all member accounts within your AWS Organizations. However, for the management account, you must manually create the service-linked role to enable monitoring.

The service cannot create the service-linked role in the management account. You must create this role manually in the management account by working with AWS CloudFormation stack sets.

Containment: In the event of a security incident, AWS Security Incident Response can execute containment actions to quickly mitigate the impact, such as isolating compromised hosts or rotating credentials. Security Incident Response does not enable containment capabilities by default. To execute these containment actions, you must first grant the necessary permissions to the service. This can be done by deploying an AWS CloudFormation StackSet, which creates the required roles.

User tasks

Contents

Dashboard

On the AWS Security Incident Response console, the dashboard provides you with an overview of your incident response team, your proactive response status, and a four-week rolling count of cases.

Select `View incident response team` to access details of your incident response teammates.

Select `proactive response` to identify if alert triaging is enabled. If you don't have the `alert triaging` workflow enabled, you can monitor its status and choose `Proactive Response` to enable it.

The *My Cases* section of the dashboard shows the number of opened and closed AWS supported cases, along with self-managed cases assigned to you within a defined period. It also shows the mean time it took to resolve the closed cases in hours.

Managing my Incident Response Team

Your incident response teams contains stakeholders for the incident response process. You can configure up to ten stakeholders as part of your membership.

Examples for internal stakeholders include members of your incident response team, security analysts, application owners, and your security leadership team.

Examples for external stakeholders include individuals from independent software vendors (ISV) and managed service providers (MSP) that you want to include in an incident response process.

> **Note**
>
> Setting up your incident response team does not automatically grant teammates access to service resources such as membership and cases. You can use AWS managed policies for AWS Security Incident Response to grant read and write access to resources. Click here to learn more.

Your incident response teammates specified on a membership level will be automatically added to any case. You can add or remove individual teammates at any time after a case has been created.

The incident response team will receive an email notification on the following events:

- Case (create, delete, update)
- Comment (create, delete, update)
- Attachment (create, delete, update)
- Membership (create, update, cancel, resume)

Account association to AWS Organizations

When you enable AWS Security Incident Response, the membership will be created and aligned to your AWS Organizations. All accounts within your Organizations are aligned to your AWS Security Incident Response membership.

For more detail, please see Managing AWS Security Incident Response accounts with AWS Organizations.

Monitoring and investigation

AWS Security Incident Response reviews and triages security alerts from Amazon GuardDuty and AWS Security Hub, then configures suppression rules based on your environment to prevent unnecessary alerts. The AWS CIRT team investigates non-triaged findings and quickly escalates and guides your team to rapidly contain potential issues. If desired, you can grant AWS Security Incident Response permission to implement containment actions on your behalf.

AWS Security Incident Response aligns to the NIST 800-61r2 Computer Security event Handling Guide for Security event Response. By aligning to this industry standard, AWS Security Incident Response provides a consistent approach to security event management and adhere to best practices in securing and responding to security events in your AWS environment.

When the AWS Security Incident Response service identifies a security alert or you request security assistance, the AWS CIRT investigates. The team collects log events and service data such as GuardDuty alerts, triages and analyzes that data, performs remediation and containment activities, and provides post-incident reporting.

Contents

Prepare

The AWS Security Incident Response team investigates and partners with you throughout the security event response lifecycle. It is recommended that you set up this team and assign the necessary permissions before a security event occurs.

Detect and Analyze

AWS Security Incident Response monitors, triages, investigates security findings from Amazon GuardDuty and integrations through AWS Security Hub. Additional dctions that can significantly enhance the scope and effectiveness of AWS Security Incident Response's monitoring and investigation capabilities include:

Enabling supported sources of detection

> **Note**
>
> AWS Security Incident Response service costs do not include usage and other costs and fees associated with supported sources of detection or use of other AWS services. Please refer to individual feature or service pages for cost details.

Amazon GuardDuty

GuardDuty is a threat detection service that continuously monitors, analyzes, and processes data sources and logs in your AWS environment. Enabling GuardDuty is not required to use AWS Security Incident Response; however, to use the proactive response and and alert triaging feature Amazon GuardDuty must be enabled.

To enable GuardDuty across your organization, please see the `Setting up GuardDuty` section of the Amazon GuardDuty User Guide.

We highly recommend that you enable GuardDuty in all supported AWS Regions. This enables GuardDuty to generate findings about unauthorized or unusual activity even in regions that you are not actively using. For more information, reference Amazon GuardDuty Regions and endpoints

Enabling GuardDuty provides AWS Security Incident Response access to critical threat detection data, enhancing its ability to identify and respond to potential security issues in your AWS environment.

AWS Security Hub

Security Hub can ingest security findings from several AWS services and supported third-party security solutions. These integrations can help AWS Security Incident Response monitor and investigate findings coming from other detection tools.

To enable Security Hub with Organizations integration please refer to the AWS Security Hub User Guide.

There are multiple ways of enabling integrations on Security Hub. For third-party product integrations, you may need to purchase the integration from the AWS Marketplace, and then configure the integration. The integration information provides links to complete these tasks. Learn more about how to enable AWS Security Hub integrations.

AWS Security Incident Response can monitor and investigate findings from the following tools when they're integrated with AWS Security Hub:

- *CrowdStrike – CrowdStrike Falcon*
- *Lacework – Lacework*
- *Trend Micro – Cloud One*

By enabling these integrations, you can significantly enhance the scope and effectiveness of AWS Security Incident Response's monitoring and investigation capabilities.

Analyzing findings.

AWS Security Incident Response automations and AWS CIRT service team will analyze all findings from the supported tools. We will start learning about your environment by communicating with you using AWS Support Cases. For example, when we need to understand whether a finding is expected behavior or should be escalated to an incident. As we learn more from your environment, we will customize the service and to reduce the number of communications.

Reporting an event.

You can raise a security event through the AWS Security Incident Response service portal. It's important not to wait during a security event. AWS Security Incident Response uses automated and manual techniques to investigate security events, analyze logs, and look for anomalous patterns. Your partnership and understanding of your environment accelerates this analysis.

Communicate.

AWS Security Incident Response keeps you informed during the investigation by engaging your security contacts through the event ticket. Multiple teammates may support your event, all using the event ticket for customer-provided content and AWS updates.

Communication may include automated notifications when a security alert is generated; communication during event analysis; establishing call bridges; the ongoing analysis of artifacts such as log files; and getting investigation results to you during the security event.

AWS Security Incident Response uses two different cases types to communicate with you: AWS Support for outbound communications to notify you of an event, and AWS Security Incident Response cases to communicate on a case you have opened to us.

AWS Support Cases: The service will use AWS Support Cases to communicate with your teams. We will create support cases on each AWS account in which the finding is generated. This approach facilitates communication with the multiple teams that own the specific workloads, as they will have more knowledge about the events occurring in their areas of responsibility.

AWS Security Incident Response Cases: If we determine that a finding needs to be escalated into a security incident, we will create a AWS Security Incident Response case. This ensures that critical security issues receive the appropriate level of attention and response.

By engaging actively with these communications and providing timely responses, you can help the AWS Security Incident Response service to:

- Better understand your environment and expected behaviors.
- Reduce false positives over time.
- Improve the accuracy and relevance of alerts.
- Ensure rapid response to genuine security incidents.
- Remember, the effectiveness of the AWS Security Incident Response service improves with your collaboration, leading to a more secure and efficiently monitored AWS environment.

Contain

AWS Security Incident Response partners with you to contain events. You can configure a service role for AWS Security Incident Response to take automated and manual actions in your account as a response to alerts. You can also perform containment yourself or in partnership with your third party relationships by using SSM documents.

An essential part of containment is decision-making; such as whether to shut down a system, isolate a resource from the network, turn off access, or end sessions. These decisions are made easier when there are predetermined strategies and procedures to contain the event. AWS Security Incident Response provides the containment strategy, informs you of potential impact, and guides you on impementing the solution only after you have considered and agreed to the risks involved.

AWS Security Incident Response executes supported containment actions on your behalf to expedite response and reduce the time a threat actor has to potentially cause damage in your environment. This capability allows for faster mitigation of identified threats, minimizing potential impact and enhancing your overall security posture. There are different containment options depending on the resources under analysis. The supported containment actions are:

- *EC2 Containment:* The `AWSSupport-ContainEC2Instance` containment automation performs a reversible network containment of an EC2 instance, leaving the instance intact and running, but isolating it from any network activity and preventing it from communicating with resources within and outside your VPC.

- *IAM Containment:* The `AWSSupport-ContainIAMPrincipal` containment automation performs a reversible network containment of an IAM user or role, leaving the user or role in IAM, but isolating it from communicating with resources within your account.
- *S3 Containment:* The `AWSSupport-ContainS3Resource` containment automation performs a reversible containment of a S3 bucket, leaving the objects in the bucket, and isolating the Amazon S3 bucket or object by modifying its access policies.

> **Important**
>
> AWS Security Incident Response does not enable containment capabilities by default, to execute these containment actions, you must first grant the necessary permissions to the service using roles. You can create these roles individually per account or across your entire organization by Working with AWS CloudFormation stacksets, which create the required roles.

AWS Security Incident Response encourages you to consider containment strategies for each major event type that fit within your risk appetite. Document clear criteria to help with decision-making during an event. Criteria to consider include:

- Potential damage to resources
- Preservation of evidence and regulatory requirements
- Service unavailability (for example, network connectivity, services provided to external parties)
- Time and resources needed to implement the strategy
- Effectiveness of the strategy (for example, partial vs. full containment)
- Permanence of the solution (for example, reversible vs. irreversible)
- Duration of the solution (for example, emergency workaround, temporary workaround, permanent solution) Apply security controls that can lower risk and allow time to define and implement a more effective containment strategy.

AWS Security Incident Response advises a staged approach to achieve efficient and effective containment, involving short-term and long-term strategies based on the resource type.

- Containment strategy
  - Can AWS Security Incident Response identify the scope of the security event?

    - If yes, identify all the resources (users, systems, resources).
    - If no, investigate in parallel with executing the next step on identified resources.
  - Can the resource be isolated?
    - If yes, then proceed to isolate the affected resources.
    - If no, then work with system owners and managers to determine further actions necessary to contain the problem.
  - Are all affected resources isolated from non-affected resources?
    - If yes, then continue to the next step.
    - If no, then continue to isolate affected resources to complete short-term containment and prevent the event from escalating further.
- System backup
  - Were backup copies of affected systems created for further analysis?
  - Are the forensic copies encrypted and stored in a secure location?
    - If yes, then continue to the next step.
    - If no, encrypt the forensic images, then store them in a secure location to prevent accidental usage, damage, and tampering.

Eradicate

During the eradication phase, it is important to identify and address all affected accounts, resources, and instances - such as by deleting malware, removing compromised user accounts, and mitigating any discovered vulnerabilities - to apply uniform remediation across the environment.

It is a best practice to use a phased approach to eradication and recovery, and to prioritize remediation steps. The purpose of the early phases is to increase the overall security quickly (days to weeks) with high-value changes to prevent future events. The later phases can focus on longer-term changes (for example, infrastructure changes), and ongoing work to keep the enterprise as secure as possible. Each case is unique and AWS CIRT will work with you to assess necessary actions.

Consider the following:

- Can you re-image the system and harden it with patches or other countermeasures to prevent or reduce the risk of attacks?

- Can you replace the infected system with a new instance or resource, enabling a clean baseline while terminating the infected item?
- Have you removed all malware and other artifacts left behind by the unauthorized use, and hardened the affected systems against further attacks?
- Is there a requirement for forensics on the impacted resources?

Recover

AWS Security Incident Response provides you guidance to help restore systems to normal operation, confirm they are functioning properly, and remediate any vulnerabilities to prevent similar events in the future. AWS Security Incident Response does not directly help with recovery of systems. Key considerations include:

- Are the affected systems patched and hardened against the recent attack?
- What is the feasible timeline to restore the systems to production?
- What tools will you use to test, monitor, and verify the restored systems?

Post incident report

AWS Security Incident Response provides a summary of the event after the conclusion of security activities between your team and ours.

At the end of each month, the AWS Security Incident Response service will send monthly reports to the primary point of contact for each customer via email. The reports will be delivered in a PDF format using the metrics described below. Customers will receive one report per AWS Organizations.

Case metrics

- Cases created
  - Dimension name: Type
  - Dimension values: AWS supported, self supported
  - Unit: Count
  - Description: The number of cases created.
- Cases closed
  - Dimension name: Type

- Dimension values: AWS supported, self-managed
- Unit: Count
- Description: A measure of the total number of cases closed.

- Opened cases
  - Dimension name: Type
  - Dimension values: AWS supported, self supported
  - Unit: Count
  - Description: The number of open cases.

Triaging metrics

- Findings received
  - Unit: Count
  - Description: The number of findings sent to triaging.
- Findings archived
  - Unit: Count
  - Description: The number of findings archived after processed without manual investigation.
- Findings Manually investigated
  - Unit: Count
  - Description: The number of findings with manual investigation performed.
- Investigations archived
  - Unit: Count
  - Description: The number of manual investigations resulting in false positive and sent for archiving
- Investigations escalated
  - Unit: Count
  - Description: The number of manual investigations resulting in a security incident

Cases

AWS Security Incident Response allows you to create two types of cases - AWS supported or self-managed cases.

Create an AWS supported case

You can create an AWS supported case from the AWS Security Incident Response, the API, or the AWS Command Line Interface. AWS supported cases allow you to receive support from the AWS Customer Incident Response Team (CIRT).

> **Note**
>
> AWS CIRT will respond to your case within 15 minutes. Response time is for a first response from AWS CIRT. We will make every reasonable effort to respond to your initial request within this time frame. This response time does not apply to subsequent responses.

The following example covers use of the console.

1. Sign in to the AWS Management Console. Open the Security Incident Response console at https://console.aws.amazon.com/security-ir/.
2. Choose **Create Case**
3. Choose **Resolve case with AWS**
4. Select the type of request
   a. **Active Security Incident**: This type is for urgent incident response support and services.
   b. **Investigations**: Investigations allow you to get support for perceived security incidents where the AWS CIRT can support in log dive and secondary confirmation of incident response investigation.
5. Set the start date estimate to the date of your earliest indicator of the incident. For example, when you experienced abnormal behavior for the first time or when you received the first related security alert.
6. Define a title for the case
7. Provide a detailed description of the case. Consider the following aspects which can help incident responders with the case resolution:
   a. What happened?
   b. Who discovered and reported the incident?
   c. Who is affected by the case?
   d. What is the known impact?
   e. What is the urgency for this case?

f. Add one or multiple AWS account IDs that are in scope of the case.

8. Add optional case details:
   a. Select the main services that are impacted from the drop-down list.
   b. Select the main regions that are impacted from the drop-down list.
   c. Add one or many threat actor IP addresses that you identified as part of this case.
9. Add optional additional incident responders to the case that will receive notifications. To add an individual, do the following:
   a. Add an email address.
   b. Add an optional first and last name.
   c. Choose **Add new** to add another individual.
   d. To remove an individual, choose the **Remove** option for an individual.
   e. Choose **Add** to add all listed individuals to the case.
      i. You can select multiple individuals and choose **Remove** to delete them from the list.
10. Add optional tags to the case.
   a. To add a tag, do the following:
   b. Choose **Add new tag**.
   c. For **Key**, enter the name of the tag.
   d. For **Value**, enter the value of the tag.
   e. To remove a tag, choose the **Remove** option for that tag.

After a AWS supported case has been created, the AWS CIRT and your incident response team are immediately notified.

Create a self-managed case

You can create a self-managed from the AWS Security Incident Response, the API, or the AWS Command Line Interface. This type of case *DOES NOT* engage the AWS CIRT. The following example covers use of the console.

1. Sign in to the AWS Management Console. Open the Security Incident Response console at https://console.aws.amazon.com/security-ir/.
2. Choose **Create Case.**
3. Choose **Resolve case with my own incident response team.**

4. Set the start date estimate to the date of your earliest indicator of the incident. For example, when you experienced abnormal behavior for the first time or when you received the first related security alert.
5. Define a title for the case. It is recommended to include the data into the case title as suggested when selecting the **Generate Title** option.
6. Enter AWS account IDs that are part of the case. To add an account ID, do the following:
    a. Enter the 12-digit account ID and choose **Add account**.
    b. To remove an account, choose **Remove** next to the account you want to remove from the case.
7. Provide a detailed description of the case.
    a. Consider the following aspects which can help incident responders with the case resolution:
        i. What happened?
        ii. Who discovered and reported the incident?
        iii. Who is affected by the case?
        iv. What is the known impact?
        v. What is the urgency for this case?
8. Add optional case details:
    a. Select the main services that are impacted from the drop-down list.
    b. Select the main regions that are impacted from the drop-down list.
    c. Add one or many threat actor IP addresses that you identified as part of this case.
9. Add optional additional incident responders to the case that will receive notifications. To add an individual, do the following:
    a. Add an email address.
    b. Add an optional first and last name.
    c. Choose **Add new** to add another individual.
    d. To remove an individual, choose the **Remove** option for an individual.
    e. Choose **Add** to add all listed individuals to the case. You can select multiple individuals and choose **Remove** to delete them from the list.
10. Add optional tags to the case. To add a tag, do the following:
    a. Choose **Add new tag**.
    b. For **Key**, enter the name of the tag.
    c. For **Value**, enter the value of the tag.

d. To remove a tag, choose the **Remove** option for that tag.

The incident response team will be notified by e-mail after the case is created.

Responding to an AWS generated case

AWS Security Incident Response may create an outbound notification or case when you need to act on or be aware of something that might impact your account or resources. This will only occur if you have enabled the proactive response and alert triaging workflows enabled as part of your subscription.

These notifications will appear in AWS Support Center. The AWS Support user guide has information and detailed steps for updating, resolving, and reopening these cases.

Managing Cases

Contents

Changing the case status

A case will be in one of the following states:

- Submitted: This is the initial status of a case. Cases in this status have been submitted by a requested, but are not yet being worked on.

- Detection and Analysis: This status indicates an incident responder has started work on the case. This phase includes data gathering, triaging the event, and performing analysis to create data driven conclusions.
- Containment, Eradication and Recovery: In this status the incident responder has identified suspicious activity that requires additional effort to remove. The incident responder will provide recommendations to you for business risk analysis and additional actions. If you have enabled the opt-in features for the service, then an AWS incident responder will seek your consent to perform containment actions with SSM documents in the impacted account(s).
- Post-incident activities: In this status the primary security event has been contained. The focus now is to recover and return business operations to normal. A summary and root cause analysis is provided if the resolver for the case is AWS-supported.
- Closed: This is the final status of the workflow. Cases in a closed status indicate work has been completed. Closed cases cannot be reopened, so ensure all actions are complete before transitioning to this status.

Choose **Action/Update Status** to change the status of the case for self-managed cases. For AWS supported cases, the status is set by the AWS CIRT responder.

Changing the resolver

For self-managed cases, your incident response team can request help from AWS. Choose **Get help from AWS** to change the resolver for this case to AWS. Once the case is updated to AWS supported, the status is changed to **Submitted**. The existing case history will be available to AWS CIRT. Once you have requested help from AWS you will not be able to change it back to self-managed.

Action Items

An AWS CIRT responder working on the case may request actions from your internal team.

Action items that appear after a case has been created include:

- Request to provide permissions for an incident responder to access a case
- Request to provide more information about the case

Action item when a customer action is pending:

- Request to act on a new comment to proceed the case

Action items when a case is ready to close:

- Request to review the case report
- Request to close the case

Edit a case

Choose **Edit** to change the details of a case.

For AWS supported and self-managed cases:

You can change the following case details after a case has been created:

- Title
- Description

For AWS supported cases only:

You can change the additional fields:

- **Request type:**
  - **Active Security Incident**: This type is for urgent incident response support and services.
  - **Investigations**: Investigations allow you to get support for perceived security incidents where the AWS CIRT can support in log dive and secondary confirmation of incident response investigation. event.
- **Start date estimate**: Change this field if you received indicators for this case that pre-date the initial provided start date. Consider providing additional details in regards to the newly detected indicator in the description field or add a comment in the communications tab.

Communications

AWS CIRT can add comments to document their activities when working on a case. Different AWS CIRT responders can work on a case at the same time. They are represented as **AWS Responder** within the communication log.

Permissions

The permissions tab lists all individuals that will be notified for any change to the case. You can add and remove individuals from the list until the case is closed.

> **Note**
>
> Individual cases allow you to include up to 30 total stakeholders. Additional permission configuration is required to grant case-level access to these stakeholders.

Provide access to a case in the console

To provide access to the case in the AWS Management Console, you can copy the IAM permission policy template and add this permission to a user or role.

Adding the IAM policy to a user or a role:

1. Copy the IAM permission policy.
2. Open IAM in the via https://console.aws.amazon.com/iam/.
3. In the navigation pane, choose **User** or **Roles.**
4. Select a user or role to open the details page.
5. In the permissions tab, choose **Add permissions.**
6. Choose **Attach policy**.
7. Select the appropriate AWS Security Incident Response managed policy.
8. Choose **Add policy**.

Attachments

Your incident responders can add attachments to a case that help other incident responders with their investigation for self-managed cases.

> **Note**
>
> If you choose an AWS supported case, AWS cannot view attachments. All details for AWS supported cases must be shared via case comments or through you providing a screenshare using your preferred communications technology.

Choose **Upload** to select a file from your computer to be added to the case.

> **Note**
>
> Any uploaded attachments are deleted seven days after a case has been `Closed`.

Tags

A tag is an optional label that you can assign to your cases to hold metadata about that resource. Each tag is a label consisting of a key and an optional value. You can use tag to search, allocate costs, and authenticate permissions for the resource.

To add a tag, do the following:

1. Choose **Add new tag**.
2. For **Key**, enter the name of the tag.
3. For **Value**, enter the value of the tag.

To remove a tag, choose the **Remove** option for that tag.

Case activities

Audit trails provide detailed chronological records of all case activities. They provide important information in post-event activities and help to identify potential improvements. The time, user, action, and details of any case change are logged in the case audit trail.

Closing a case

For AWS supported cases, choose **Close Case** on the case details page to permanently close the case at any status. A case typically reaches the status **Ready to Close** before it is permanently closed. If you close a case prematurely at any other status than **Ready to Close**, you are requesting that AWS CIRT will stop working on this AWS supported case.

If your incident response team is the responder, select **Action/Close Case** on the case details page.

> **Note**
>
> The "Ready to Close" status signifies that a case can be permanently closed and that there is no additional work to be done on a case.

A case cannot be re-opened again after it has been permanently closed. All information will be available read-only. To prevent accidental closure, you will be asked to confirm that you want to close the case.

Working with AWS CloudFormation stacksets

> **Important**
>
> AWS Security Incident Response does not enable containment capabilities by default, to execute these containment actions, you must first grant the necessary permissions to the service using roles. You can create these roles individually per account or across your entire organization by deploying AWS CloudFormation StackSets, which create the required roles.

You can find specific instructions on how to Create a stack set with service-managed permissions.

Following are template stacksets to create the *AWSSecurityIncidentResponseContainment* and *AWSSecurityIncidentResponseContainmentExecution* roles.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'Template for AWS Security Incident Response containment roles'

Resources:
  AWSSecurityIncidentResponseContainment:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: AWSSecurityIncidentResponseContainment
      AssumeRolePolicyDocument:
        {
          'Version': '2012-10-17',
          'Statement':
            [
              {
                'Effect': 'Allow',
```

```
                'Principal': { 'Service': 'containment.security-ir.amazonaws.com' },
                'Action': 'sts:AssumeRole',
                'Condition': { 'StringEquals': { 'sts:ExternalId': !Sub
 '${AWS::AccountId}' } },
              },
              {
                'Effect': 'Allow',
                'Principal': { 'Service': 'containment.security-ir.amazonaws.com' },
                'Action': 'sts:TagSession',
              },
            ],
        }
      Policies:
        - PolicyName: AWSSecurityIncidentResponseContainmentPolicy
          PolicyDocument:
            {
              'Version': '2012-10-17',
              'Statement':
                [
                  {
                    'Effect': 'Allow',
                    'Action': ['ssm:StartAutomationExecution'],
                    'Resource':
                      [
                        !Sub 'arn:${AWS::Partition}:ssm:*:*:automation-definition/
AWSSupport-ContainEC2Instance:$DEFAULT',
                        !Sub 'arn:${AWS::Partition}:ssm:*:*:automation-definition/
AWSSupport-ContainS3Resource:$DEFAULT',
                        !Sub 'arn:${AWS::Partition}:ssm:*:*:automation-definition/
AWSSupport-ContainIAMPrincipal:$DEFAULT',
                      ],
                  },
                  {
                    'Effect': 'Allow',
                    'Action':
                      ['ssm:DescribeInstanceInformation', 'ssm:GetAutomationExecution',
 'ssm:ListCommandInvocations'],
                    'Resource': '*',
                  },
                  {
                    'Effect': 'Allow',
                    'Action': ['iam:PassRole'],
                    'Resource': !GetAtt
 AWSSecurityIncidentResponseContainmentExecution.Arn,
```

```
                    'Condition': { 'StringEquals': { 'iam:PassedToService':
'ssm.amazonaws.com' } },
                  },
                ],
            }
 AWSSecurityIncidentResponseContainmentExecution:
   Type: 'AWS::IAM::Role'
   Properties:
     RoleName: AWSSecurityIncidentResponseContainmentExecution
     AssumeRolePolicyDocument:
       {
         'Version': '2012-10-17',
         'Statement':
           [{ 'Effect': 'Allow', 'Principal': { 'Service': 'ssm.amazonaws.com' },
'Action': 'sts:AssumeRole' }],
       }
     ManagedPolicyArns:
       - !Sub arn:${AWS::Partition}:iam::aws:policy/SecurityAudit
     Policies:
       - PolicyName: AWSSecurityIncidentResponseContainmentExecutionPolicy
         PolicyDocument:
           {
             'Version': '2012-10-17',
             'Statement':
               [
                 {
                   'Sid': 'AllowIAMContainment',
                   'Effect': 'Allow',
                   'Action':
                     [
                       'iam:AttachRolePolicy',
                       'iam:AttachUserPolicy',
                       'iam:DeactivateMFADevice',
                       'iam:DeleteLoginProfile',
                       'iam:DeleteRolePolicy',
                       'iam:DeleteUserPolicy',
                       'iam:GetLoginProfile',
                       'iam:GetPolicy',
                       'iam:GetRole',
                       'iam:GetRolePolicy',
                       'iam:GetUser',
                       'iam:GetUserPolicy',
                       'iam:ListAccessKeys',
                       'iam:ListAttachedRolePolicies',
```

```
                'iam:ListAttachedUserPolicies',
                'iam:ListMfaDevices',
                'iam:ListPolicies',
                'iam:ListRolePolicies',
                'iam:ListUserPolicies',
                'iam:ListVirtualMFADevices',
                'iam:PutRolePolicy',
                'iam:PutUserPolicy',
                'iam:TagMFADevice',
                'iam:TagPolicy',
                'iam:TagRole',
                'iam:TagUser',
                'iam:UntagMFADevice',
                'iam:UntagPolicy',
                'iam:UntagRole',
                'iam:UntagUser',
                'iam:UpdateAccessKey',
                'identitystore:CreateGroupMembership',
                'identitystore:DeleteGroupMembership',
                'identitystore:IsMemberInGroups',
                'identitystore:ListUsers',
                'identitystore:ListGroups',
                'identitystore:ListGroupMemberships',
              ],
            'Resource': '*',
          },
          {
            'Sid': 'AllowOrgListAccounts',
            'Effect': 'Allow',
            'Action': 'organizations:ListAccounts',
            'Resource': '*',
          },
          {
            'Sid': 'AllowSSOContainment',
            'Effect': 'Allow',
            'Action':
              [
                'sso:CreateAccountAssignment',
                'sso:DeleteAccountAssignment',
                'sso:DeleteInlinePolicyFromPermissionSet',
                'sso:GetInlinePolicyForPermissionSet',
                'sso:ListAccountAssignments',
                'sso:ListInstances',
                'sso:ListPermissionSets',
```

```
                        'sso:ListPermissionSetsProvisionedToAccount',
                        'sso:PutInlinePolicyToPermissionSet',
                        'sso:TagResource',
                        'sso:UntagResource',
                      ],
                    'Resource': '*',
                  },
                  {
                    'Sid': 'AllowSSORead',
                    'Effect': 'Allow',
                    'Action': ['sso-directory:SearchUsers', 'sso-
directory:DescribeUser'],
                    'Resource': '*',
                  },
                  {
                    'Sid': 'AllowS3Read',
                    'Effect': 'Allow',
                    'Action':
                      [
                        's3:GetAccountPublicAccessBlock',
                        's3:GetBucketAcl',
                        's3:GetBucketLocation',
                        's3:GetBucketOwnershipControls',
                        's3:GetBucketPolicy',
                        's3:GetBucketPolicyStatus',
                        's3:GetBucketPublicAccessBlock',
                        's3:GetBucketTagging',
                        's3:GetEncryptionConfiguration',
                        's3:GetObject',
                        's3:GetObjectAcl',
                        's3:GetObjectTagging',
                        's3:GetReplicationConfiguration',
                        's3:ListBucket',
                        's3express:GetBucketPolicy',
                      ],
                    'Resource': '*',
                  },
                  {
                    'Sid': 'AllowS3Write',
                    'Effect': 'Allow',
                    'Action':
                      [
                        's3:CreateBucket',
                        's3:DeleteBucketPolicy',
```

```
                's3:DeleteObjectTagging',
                's3:PutAccountPublicAccessBlock',
                's3:PutBucketACL',
                's3:PutBucketOwnershipControls',
                's3:PutBucketPolicy',
                's3:PutBucketPublicAccessBlock',
                's3:PutBucketTagging',
                's3:PutBucketVersioning',
                's3:PutObject',
                's3:PutObjectAcl',
                's3express:CreateSession',
                's3express:DeleteBucketPolicy',
                's3express:PutBucketPolicy',
              ],
            'Resource': '*',
          },
          {
            'Sid': 'AllowAutoScalingWrite',
            'Effect': 'Allow',
            'Action':
              [
                'autoscaling:CreateOrUpdateTags',
                'autoscaling:DeleteTags',
                'autoscaling:DescribeAutoScalingGroups',
                'autoscaling:DescribeAutoScalingInstances',
                'autoscaling:DescribeTags',
                'autoscaling:EnterStandby',
                'autoscaling:ExitStandby',
                'autoscaling:UpdateAutoScalingGroup',
              ],
            'Resource': '*',
          },
          {
            'Sid': 'AllowEC2Containment',
            'Effect': 'Allow',
            'Action':
              [
                'ec2:AuthorizeSecurityGroupEgress',
                'ec2:AuthorizeSecurityGroupIngress',
                'ec2:CopyImage',
                'ec2:CreateImage',
                'ec2:CreateSecurityGroup',
                'ec2:CreateSnapshot',
                'ec2:CreateTags',
```

```
                    'ec2:DeleteSecurityGroup',
                    'ec2:DeleteTags',
                    'ec2:DescribeImages',
                    'ec2:DescribeInstances',
                    'ec2:DescribeSecurityGroups',
                    'ec2:DescribeSnapshots',
                    'ec2:DescribeTags',
                    'ec2:ModifyNetworkInterfaceAttribute',
                    'ec2:RevokeSecurityGroupEgress',
                ],
                'Resource': '*',
            },
            {
                'Sid': 'AllowKMSActions',
                'Effect': 'Allow',
                'Action':
                    [
                    'kms:CreateGrant',
                    'kms:DescribeKey',
                    'kms:GenerateDataKeyWithoutPlaintext',
                    'kms:ReEncryptFrom',
                    'kms:ReEncryptTo',
                    ],
                'Resource': '*',
            },
            {
                'Sid': 'AllowSSMActions',
                'Effect': 'Allow',
                'Action': ['ssm:DescribeAutomationExecutions'],
                'Resource': '*',
            },
        ],
    }
```

Cancel Membership

A role having the CancelMembership permission for AWS Security Incident Response can cancel the membership from the console, the API, or AWS Command Line Interface.

Important

Once a membership has been canceled, you will be unable to view historic case data. Cancellations occur at the end of the billing cycle. If you cancel during the month, your membership will be available until the end of the month. Any resources or investigations that are `Active` or `ready to close` will be terminated upon final membership cancellation at the end of the billing cycle.

Important

If you resubscribe to the service, a new membership will be created and the case resources that lived under the prior membership are only accessible if you downloaded them prior to cancellation.

After the membership has been canceled, everyone in the membership incident response team are notified by email.

Important

If you created a membership using a delegated administrator account and you use the AWS Organizations API to remove the delegated administrator designation from the account, the membership will be terminated immediately.

Tagging AWS Security Incident Response resources

A *tag* is a metadata label that you assign or that AWS assigns to an AWS resource. Each tag consists of a *key* and a *value*. For tags that you assign, you define the key and value. For example, you might define the key as `stage` and the value for one resource as `test`.

Tags help you do the following:

- Identify and organize your AWS resources. Many AWS services support tagging, so you can assign the same tag to resources from different services to indicate that the resources are related.
- Track your AWS costs. You activate these tags on the AWS Billing dashboard. AWS uses the tags to categorize your costs and deliver a monthly cost allocation report to you. For more information, see Use cost allocation tags in the AWS Billing User Guide.
- Control access to your AWS resources. For more information, see Controlling access using tags in the IAM User Guide.

Refer to the AWS Security Incident Response API reference for tagging.

Using AWS CloudShell to work with AWS Security Incident Response

AWS CloudShell is a browser-based, pre-authenticated shell that you can launch directly from the AWS Management Console. You can run AWS CLI commands against AWS services (including AWS Security Incident Response) using your preferred shell (Bash, PowerShell or Z shell). And you can do this without needing to download or install command line tools.

You launch AWS CloudShell from the AWS Management Console, and the AWS credentials you used to sign in to the console are automatically available in a new shell session. This pre-authentication of AWS CloudShell users allows you to skip configuring credentials when interacting with AWS services such as Security Incident Response using AWS CLI version 2 (pre-installed on the shell's compute environment).

Contents

Obtaining IAM permissions for AWS CloudShell

Using the access management resources provided by AWS Identity and Access Management, administrators can grant permissions to IAM users so they can access AWS CloudShell and use the environment's features.

The quickest way for an administrator to grant access to users is through an AWS managed policy. An AWS managed policy is a standalone policy that's created and administered by AWS. The following AWS managed policy for CloudShell can be attached to IAM identities:

- `AWSCloudShellFullAccess`: Grants permission to use AWS CloudShell with full access to all features.

If you want to limit the scope of actions that an IAM user can perform with AWS CloudShell, you can create a custom policy that uses the `AWSCloudShellFullAccess` managed policy as a template. For more information about limiting the actions that are available to users in CloudShell, see Managing AWS CloudShell access and usage with IAM policies in the *AWS CloudShell User Guide*.

> **Note**
>
> Your IAM identity also requires a policy that grants permission to make calls to Security Incident Response.

Interacting with Security Incident Response using AWS CloudShell

After you launch AWS CloudShell from the AWS Management Console, you can immediately start to interact with Security Incident Response using the command line interface.

> **Note**
>
> When using AWS CLI in AWS CloudShell, you don't need to download or install any additional resources. Moreover, because you're already authenticated within the shell, you don't need to configure credentials before making calls.

Working with AWS CloudShell and Security Incident Response

- From the AWS Management Console, you can launch CloudShell by choosing the following options available on the navigation bar:
  - Choose the CloudShell icon.
  - Start typing "cloudshell" in Search box and then choose the CloudShell option.

Logging AWS Security Incident Response API calls using AWS CloudTrail

AWS Security Incident Response is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Security Incident Response. CloudTrail captures all API calls for Security Incident Response as events. The calls captured include calls from the Security Incident Response console and code calls to the Security Incident Response API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Security Incident Response. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Security Incident Response, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Security Incident Response information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Security Incident Response, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail Event history.

For an ongoing record of events in your AWS account past 90 days, create a trail or a CloudTrail Lake event data store.

CloudTrail trails

A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. All trails created using the AWS Management Console are multi-Region. You can create a single-Region or a multi-Region trail by using the AWS CLI. Creating a multi-Region trail is recommended because you capture activity in all AWS Regions in your account. If you create a single-Region trail, you can view only the events logged in the trail's AWS Region. For more information about trails, see Creating a trail for your AWS account and Creating a trail for an organization in the *AWS CloudTrail User Guide*.

You can deliver one copy of your ongoing management events to your Amazon S3 bucket at no charge from CloudTrail by creating a trail, however, there are Amazon S3 storage charges. For

more information about CloudTrail pricing, see AWS CloudTrail Pricing. For information about Amazon S3 pricing, see Amazon S3 Pricing.

CloudTrail Lake event data stores

CloudTrail Lake lets you run SQL-based queries on your events. CloudTrail Lake converts existing events in row-based JSON format to Apache ORC format. ORC is a columnar storage format that is optimized for fast retrieval of data. Events are aggregated into *event data stores*, which are immutable collections of events based on criteria that you select by applying advanced event selectors. The selectors that you apply to an event data store control which events persist and are available for you to query. For more information about CloudTrail Lake, see Working with AWS CloudTrail Lake in the *AWS CloudTrail User Guide*.

CloudTrail Lake event data stores and queries incur costs. When you create an event data store, you choose the pricing option you want to use for the event data store. The pricing option determines the cost for ingesting and storing events, and the default and maximum retention period for the event data store. For more information about CloudTrail pricing, see AWS CloudTrail Pricing.

All Security Incident Response actions are logged by CloudTrail and are documented in the AWS Security Incident Response API Reference. For example, calls to the `CreateMembership`, `CreateCase` and `UpdateCase` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding Security Incident Response log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of

the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the CreateCase action.


```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA0000000000000000:user",
        "arn": "arn:aws:sts::123412341234:assumed-role/Admin/user",
        "accountId": "123412341234",
        "accessKeyId": "****",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA0000000000000000",
                "arn": "arn:aws:iam::123412341234:role/Admin",
                "accountId": "123412341234",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2024-10-13T06:32:53Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-10-13T06:40:45Z",
    "eventSource": "security-ir.amazonaws.com",
    "eventName": "CreateCase",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "1.2.3.4",
    "userAgent": "aws-cli/2.17.23 md/awscrt#0.20.11 ua/2.0 os/macos#23.6.0 md/
arch#x86_64 lang/python#3.11.9 md/pyimpl#CPython cfg/retry-mode#standard md/
installer#exe md/prompt#off md/command#security-ir.create-case",
    "requestParameters": {
        "impactedServices": [
            "Amazon GuardDuty"
        ],
        "impactedAccounts": [],
        "clientToken": "testToken112345679",
        "resolverType": "Self",
        "description": "***",
        "engagementType": "Investigation",
```

```
        "watchers": [
            {
                "email": "***",
                "name": "***",
                "jobTitle": "***"
            }
        ],
        "membershipId": "m-r1abcdabcd",
        "title": "***",
        "impactedAwsRegions": [
            {
                "region": "ap-southeast-1"
            }
        ],
        "reportedIncidentStartDate": 1711553521,
        "threatActorIpAddresses": [
            {
                "ipAddress": "***",
                "userAgent": "browser"
            }
        ]
    },
    "responseElements": {
        "caseId": "0000000001"
    },
    "requestID": "2db4b08d-94a9-457a-9474-5892e6c8191f",
    "eventID": "b3fa3990-db82-43be-b120-c81262cc2f19",
    "readOnly": false,
    "resources": [
        {
            "accountId": "123412341234",
            "type": "AWS::SecurityResponder::Case",
            "ARN": "arn:aws:security-ir:us-east-1:123412341234:case/*"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123412341234",
    "eventCategory": "Management"
}
```

Managing AWS Security Incident Response accounts with AWS Organizations

AWS Security Incident Response is integrated with AWS Organizations. The AWS Organizations management account for the organization can designate an account as the delegated administrator for AWS Security Incident Response. This action enables AWS Security Incident Response as a trusted service in AWS Organizations. For information about how these permissions are granted, see Using AWS Organizations with other AWS services.

The following sections will walk you through various tasks that you may perform as a delegated Security Incident Response administrator account.

Contents

Considerations and recommendations for using AWS Security Incident Response with AWS Organizations

The following considerations and recommendations can help you understand how a delegated Security Incident Response administrator account operates in AWS Security Incident Response:

A delegated Security Incident Response administrator account is regional.

The delegated Security Incident Response administrator account and member accounts must be added through AWS Organizations.

Delegated administrator account for AWS Security Incident Response.

You may designate one member account as the delegated Security Incident Response administrator account. For example, if you designate a member account *111122223333* in

Europe (Ireland), you can't designate another member account *555555555555* in *Canada (Central)*. It is required that you use the same account as delegated Security Incident Response administrator account in all other Regions.

It is not recommended to set your organization's management as the delegated Security Incident Response administrator account.

Your organization's management can be the delegated Security Incident Response administrator account. However, the AWS security best practices follow the principle of least privilege and doesn't recommend this configuration.

Removing a delegated Security Incident Response administrator account from a live subscription cancels the subscription immediately.

If you remove a delegated Security Incident Response administrator account, AWS Security Incident Response removes all the member accounts associated with this delegated Security Incident Response administrator account. AWS Security Incident Response will not longer be enabled for all these member accounts.

Enabling trusted access for AWS Account Management

Enabling trusted access for AWS Security Incident Response allows the delegated administrator of the management account to modify the information and metadata (for example, primary or alternate contact details) specific to each member account in AWS Organizations.

Use the following procedure to enable trusted access for AWS Security Incident Response in your organization.

Minimum permissions

To perform these tasks, you must meet the following requirements:

- You can perform this only from the organization's management account.
- Your organization must have all features enabled.

Console

To enable trusted access for AWS Security Incident Response

1. Sign in to the AWS Organizations console. You must sign in as an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization's management account.
2. Choose **Services** in the navigation pane.
3. Choose **AWS Security Incident Response** in the list of services.
4. Choose **Enable trusted access**.
5. In the **Enable trusted access for AWS Security Incident Response** dialog box, type **enable** to confirm it, and then choose **Enable trusted access**.

API/CLI

To enable trusted access for AWS Account Management

After running the following command, you can use credentials from the organization's management account to call Account Management API operations that use the `--accountId` parameter to reference member accounts in an organization.

- AWS CLI: enable-aws-service-access

  The following example enables trusted access for AWS Security Incident Response in the calling account's organization.

```
$ aws organizations enable-aws-service-access \
                                        --service-principal security-
ir.amazonaws.com
```

  This command produces no output if it's successful.

Permissions required to designate a delegated Security Incident Response administrator account

You can chose to set up your AWS Security Incident Response membership using delegated administrator for AWS Organizations. For information about how these permissions are granted, see Using AWS Organizations with other AWS services.

> **Note**
>
> AWS Security Incident Response automatically enables the AWS Organizations trusted relationship when using the console for setup and management. If you use the CLI/SDK then you have to manually enable this by using the EnableAWSServiceAccess API to trust `security-ir.amazonaws.com`.

As the AWS Organizations manager, before you designate the delegated Security Incident Response administrator account for your organization, verify that you can perform the following AWS Security Incident Response actions: `sir:CreateMembership` and `sir:UpdateMembership`. These actions allow you to designate the delegated Security Incident Response administrator account for your organization by using AWS Security Incident Response. You must also ensure that you are allowed to perform the AWS Organizations actions that help you retrieve information about your organization.

To grant these permissions, include the following statement in an AWS Identity and Access Management (IAM) policy for your account:

```
{
    "Sid": "PermissionsForSIRAdmin",
    "Effect": "Allow",
    "Action": [
        "security-ir:CreateMembership",
        "security-ir:UpdateMembership",
        "organizations:EnableAWSServiceAccess",
        "organizations:RegisterDelegatedAdministrator",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:DescribeOrganizationalUnit",
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
```

```
        "organizations:ListAccounts"
    ],
    "Resource": "*"
}
```

If you want to designate your AWS Organizations management as the delegated Security Incident Response administrator account, your account will also need the IAM action: `CreateServiceLinkedRole`. This action allows you to initialize AWS Security Incident Response for the management. However, review Considerations and recommendations for using AWS Security Incident Response with AWS Organizations before you proceed to add the permissions.

To continue with designating the management as the delegated Security Incident Response administrator account, add the following statement to the IAM policy and replace *111122223333* with the AWS account ID of your organization's management:

```
{
 "Sid": "PermissionsToEnablesir"
 "Effect": "Allow",
 "Action": [
  "iam:CreateServiceLinkedRole"
 ],
 "Resource": "arn:aws:iam::111122223333:role/aws-service-role/security-
ir.amazonaws.com/AWSServiceRoleForAmazonsir",
 "Condition": {
  "StringLike": {
   "iam:AWSServiceName": "security-ir.amazonaws.com"
  }
 }
}
```

Designating a delegated administrator for AWS Security Incident Response

This section provides steps to designate a delegated administrator in the AWS Security Incident Response organization.

As a manager of the AWS organization, make sure that you read through the Considerations and recommendations on how a delegated Security Incident Response administrator account operates.

Before proceeding, ensure that you have Permissions required to designate a delegated Security Incident Response administrator account.

Choose a preferred access method to designate a delegated Security Incident Response administrator account for your organization. Only a management can perform this step.

Console

1. Open the Security Incident Response console at https://console.aws.amazon.com/security-ir/

   To sign in, use the management credentials for your AWS Organizations organization.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region in which you want to designate the delegated Security Incident Response administrator account for your organization.
3. Follow the setup wizard to create your membership, including the delegated administrator account.

API/CLI

- Run CreateMembership using the credentials of the AWS account of the organization's management.
  - Alternatively, you can use AWS Command Line Interface to do this. The following AWS CLI command designates a delegated Security Incident Response administrator account. Following are the string options available for configuring your membership:

```
{
  "customerAccountId": "stringstring",
  "membershipName": "stringstring",
  "customerType": "Standalone",
  "organizationMetadata": {
    "organizationId": "string",
    "managementAccountId": "stringstring",
    "delegatedAdministrators": [
      "stringstring"
    ]
  },
  "membershipAccountsConfigurations": {
    "autoEnableAllAccounts": true,
```

```
      "organizationalUnits": [
        "string"
      ]
    },
    "incidentResponseTeam": [
      {
        "name": "string",
        "jobTitle": "stringstring",
        "email": "stringstring"
      }
    ],
    "internalIdentifier": "string",
    "membershipId": "stringstring",
    "optInFeatures": [
      {
        "featureName": "RuleForwarding",
        "isEnabled": true
      }
    ]
  }
```

If AWS Security Incident Response is not enabled for your delegated Security Incident Response administrator account, it won't be able to take any action. If not already done so, make sure to enable AWS Security Incident Response for the newly designated delegated Security Incident Response administrator account.

Adding members to AWS Security Incident Response

There is a one to one relationship with AWS Organizations and your AWS Security Incident Response membership. As accounts are added (or removed) from your Organizations, this will be reflected in the covered accounts for your AWS Security Incident Response membership.

To add an account to your membership, follow one of the options for Managing accounts in an organization with AWS Organizations.

Removing members from AWS Security Incident Response

To remove an account from your membership, follow the procedures for removing a member account from an organization.

Troubleshooting

When you experience issues related to performing an action specific to AWS Security Incident Response, consult the topics in this section.

An ERROR is a status of an operation denoting a fault in some or all of the operations. Alternatively, you receive warnings when an issue occurs but the task still completes.

Contents

Issues

Not sending requests from the correct context.

All calls to AWS Security Incident Response APIs must originate from an IAM principal in the service delegated administrator or membership account. Ensure that you are operating from the correct IAM principal in the AWS account that is your organization's AWS Security Incident Response delegated administrator or membership account.

Errors

AccessDeniedException

You do not have sufficient access to perform this action.

Please work with your AWS administrator to ensure that you have permission to assume an IAM Role in your AWS Security Incident Response delegated administrator or membership account. Also check the role has a an IAM policy that permits the requested action. For more information see AWS Security Incident Response IAM.

ConflictException

The request causes an inconsistent state.

Please check that any case attachment file names or default response team members that you have specified are unique. Also check that your AWS Security Incident Response service

membership has not already been configured. Open the Security Incident Response console at https://console.aws.amazon.com/security-ir/ and navigate to `Membership Details`.

InternalServerException

An unexpected error occurred during the processing of the request. Please try again in a few minutes. If the issue persists, raise a case with AWS Support.

ResourceNotFoundException

The request references a resource that does not exist.

One or more of the resources specified in your request does not exist. Please check that all given resource ARNs or IDs are correct. This applies to AWS Organizations IDs, account IDs, IAM roles, memberships, cases, response team members, cases, case responders, case attachments, and case comments.

ThrottlingException

The request was denied due to request throttling.

Too many requests have been made by your IAM principal to that API function in a specified period. Wait a minute and try again. If the issue persists, please consider implementing an exponential backoff and retry algorithm.

ValidationException

The input fails to satisfy the constraints specified by an AWS service.

One or more of the data fields in your request did not meet validation and/or logical combination requirements. Please check that all resource ARNs complete, and that text values meet size and format constraints from the AWS Security Incident Response API Reference Guide. Also check that any value updates are permitted. For example, changing a case from AWS supported to self-managed is not possible.

AWS Support

If you need additional assistance, contact AWS Support Center for troubleshooting purposes. Please have the following information available:

- The AWS Region that you used

- The AWS account ID of the membership
- Your source content, if applicable and available
- Any other details about the problem that might assist with troubleshooting

Security

Contents

Data Protection in AWS Security Incident Response

Contents

The AWS shared responsibility model applies to data protection for the AWS Security Incident Response service. As described in this model, AWS is responsible for protecting the infrastructure that runs the services offered in the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, AWS security best practices state that you should protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and

Access Management (IAM). This way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- FIPS 140-3 is currently not supported by the service.

You should never put confidential or sensitive information, such as your email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with AWS Support or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we ***strongly*** recommend that you do not include credentials information in the URL to validate your request to that server.

Data encryption

Contents

Encryption at rest

Data is encrypted at rest using transparent server-side encryption. This helps reduce the operational burden and complexity involved in protecting sensitive data. With encryption at rest, you can build security-sensitive applications that meet encryption compliance and regulatory requirements.

Encryption in transit

Data gathered and accessed by AWS Security Incident Response is exclusively over a Transport Layer Security (TLS) protected channel.

Key management

AWS Security Incident Response implements integrations with AWS KMS to provide encryption at rest for case and attachment data.

AWS Security Incident Response does not support customer managed keys.

Inter-network traffic privacy

Traffic between service and on-premises clients and applications

You have two connectivity options between your private network and AWS:

- An AWS Site-to-Site VPN connection. For more information, see What is AWS Site-to-Site VPN? in the *AWS Site-to-Site VPN User Guide*.
- An AWS Direct Connect connection. For more information, see What is AWS Direct Connect? in the *AWS Direct Connect User Guide*.

Access to AWS Security Incident Response via the network is through AWS published APIs. Clients must support Transport Layer Security (TLS) 1.2. We recommend TLS 1.3. Clients must also support cipher suites with Perfect Forward Secrecy (PFS), such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). Most modern systems such as Java 7 and later support these modes. Additionally, you must sign requests using an access key ID and a secret access key that are associated with an IAM principal, or you can use the AWS Security Token Service (STS) to generate temporary security credentials to sign requests.

Traffic between AWS resources in the same Region

An Amazon Virtual Private Cloud (Amazon VPC) endpoint for AWS Security Incident Response is a logical entity within a VPC that allows connectivity only to AWS Security Incident Response. The Amazon VPC routes requests to AWS Security Incident Response and routes responses back to the VPC. For more information, see VPC endpoints in the *Amazon VPC User Guide*. For example policies that you can use to control access from VPC endpoints, see Using IAM policies to control access to DynamoDB.

> **Note**
>
> Amazon VPC endpoints are not accessible via AWS Site-to-Site VPN or AWS Direct Connect.

Identity and Access Management

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator control access to AWS resources. IAM administrators control *authenticated* (signed in) and *authorized* (have permissions) principals to use AWS Security Incident Response resources. IAM is an AWS service that you can use with no additional charge.

Contents

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in AWS Security Incident Response.

Security Administrators

These users are suggested to use the AWSSecurityIncidentResponseFullAccess managed policy to ensure they have read and write access to membership and case resources.

Case Watchers

These individuals do not have authoritative access to all cases but individual cases that you grant explicit permission for.

Incident Response Team members

Members of the team can be given both full membership and case access. It is recommended that not all individuals have authoritative action on service membership but should have access to any and all cases that are created and managed through the service. For more information, refer to AWS Security Incident Response managed policies.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on

authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signing AWS API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you may be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the 8 address and password that you used to create the account. Never use the root user for your everyday tasks and take steps to safeguard your root user credentials. Only use them to perform tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

It is best practice to require human users, including those that need administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. We recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. If you have a specific use case that requires long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control

what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.

- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

How AWS Security Incident Response Works with IAM

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS Security Incident Response resources. IAM is an AWS service that you can use with no additional charge.

IAM features that you can use with AWS Security Incident Response	
IAM feature	Service alignment
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy conditions keys	Yes (global)
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Forward acces sessions (FAS)	Yes
Service roles	No
Service-linked roles	Yes

Contents

Identity-based policies for AWS Security Incident Response

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Contents

Identity-based policy examples

By default, users and roles don't have permission to create or modify AWS Security Incident Response resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. An IAM administrator can create IAM policies to grant users permission to perform actions on the resources they need. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the *IAM User Guide*.

For details about actions and resource types defined by AWS Security Incident Response, including the format of the ARNs for each of the resource types, see *Actions, resources, and condition keys for AWS Security Incident Response* in the *Service Authorization Reference*.

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Security Incident Response resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

Get started with AWS managed policies and move toward least-privilege permissions – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.

Apply least-privilege permissions – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.

Use conditions in IAM policies to further restrict access – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.

Require multi-factor authentication (MFA) – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the AWS Security Incident Response console

To access https://console.aws.amazon.com/security-ir/, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS Security Incident Response resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

Attach the AWS Security Incident Response Access or ReadOnly AWS managed policy to ensure that users and roles can use the service console. For more information, see Adding permissions to a user in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "ViewOwnUserInfo",
"Effect": "Allow",
"Action": [
"iam:GetUserPolicy",
"iam:ListGroupsForUser",
"iam:ListAttachedUserPolicies",
"iam:ListUserPolicies",
"iam:GetUser"
],
"Resource": ["arn:AWS:iam::*:user/${AWS:username}"]
},
{
"Sid": "NavigateInConsole",
"Effect": "Allow",
"Action": [
"iam:GetGroupPolicy",
"iam:GetPolicyVersion",
```

```
"iam:GetPolicy",
"iam:ListAttachedGroupPolicies",
"iam:ListGroupPolicies",
"iam:ListPolicyVersions",
"iam:ListPolicies",
"iam:ListUsers"
],
"Resource": "*"
}
]
}
```

Resource-based policies within AWS Security Incident Response

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

For more information, refer to Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for AWS Security Incident Response

Support policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Action element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS Security Incident Response actions, see Actions defined by AWS Security Incident Response in the *Service Authorization Reference*.

Policy actions in AWS Security Incident Response use the following prefix before the action:

AWS Security Incident Response -identity

To specify multiple actions in a single statement, separate them with commas.

"Action": ["AWS Security Incident Response -identity:action1", "AWS Security Incident Response -identity:action2"]

Policy resources for Amazon AWS Security Incident Response

Supports policy resources: Yes Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Resource JSON policy element specifies the object or objects to which the action applies. Statements must include either a Resource or a NotResource element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

"Resource": "*"

Policy condition keys for AWS Security Incident Response

Supports service-specific policy condition keys: No

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Condition element (or Condition *block*) lets you specify conditions in which a statement is in effect. The Condition element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple Condition elements in a statement, or multiple keys in a single Condition element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

Access control lists (ACLs) in AWS Security Incident Response

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Attribute-based access control (ABAC) with AWS Security Incident Response

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access. ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the AWS:ResourceTag/key-name, AWS:RequestTag/key-name, or AWS:TagKeys condition keys. If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**. For more information about ABAC, see What is ABAC? in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Temporary credentials with Amazon AWS Security Incident Response

Supports temporary credentials: Yes

AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work

with IAM in the *IAM User Guide*. You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Forward access sessions for AWS Security Incident Response

Supports forward access sessions (FAS): Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Troubleshooting AWS Security Incident Response identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS Security Incident Response and IAM.

Topics

- I am not authorized to perform an action
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my AWS Security Incident Response resources

I am not authorized to perform an action

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the mateojackson IAM user tries to use the console to view details about a fictional my-example-widget resource but doesn't have the fictional AWS Security Incident Response :GetWidget permissions.

User: arn:AWS:iam::123456789012:user/mateojackson is not authorized to perform: AWS Security Incident Response :GetWidget on resource: *my*-example-widget

In this case, the policy for the mateojackson user must be updated to allow access to the my-example-widget resource by using the AWS Security Incident Response :GetWidget action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole If you receive an error that you're not authorized to perform the iam:PassRole action, your policies must be updated to allow you to pass a role to AWS Security Incident Response .

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named marymajor tries to use the console to perform an action in AWS Security Incident Response . However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

User: arn:AWS:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole

In this case, Mary's policies must be updated to allow her to perform the iam:PassRole action. If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my AWS Security Incident Response resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role.

To learn more, consult the following:

- To learn whether Amazon AWS Security Incident Response supports these features, see How AWS Security Incident Response works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Using service roles

Supports service roles: No

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

Using service-linked roles

Service-linked roles for AWS Security Incident Response

Contents

- AWS SLR: AWSServiceRoleForSecurityIncidentResponse
- AWS SLR: AWSServiceRoleForSecurityIncidentResponse_Triage
- Supported regions for AWS Security Incident Response service-linked roles

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

A service-linked role makes setting up AWS Security Incident Response easier because you don't have to manually add the necessary permissions. AWS Security Incident Response defines

the permissions of its service-linked roles, and unless defined otherwise, only AWS Security Incident Response can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

For information about other services that support service-linked roles, see AWS services that work with IAM and look for the services that have **Yes** in the Service-linked roles column. Choose a Yes with a link to view the service-linked role documentation for that service.

AWS SLR: AWSServiceRoleForSecurityIncidentResponse

AWS Security Incident Response uses the service-linked role (SLR) named AWSServiceRoleForSecurityIncidentResponse – AWS Security Incident Response policy to identify accounts subscribed, create cases, and tag related resources.

Permissions

The AWSServiceRoleForSecurityIncidentResponse service-linked role trusts the following service to assume the role:

- `triage.security-ir.com`

Attached to this role is the AWS managed policy named AWSSecurityIncidentResponseServiceRolePolicy. The service uses the role to perform actions on the following resources:

- *AWS Organizations:* Allows the service to lookup membership accounts for use with the service.
- *CreateCase:* Allows the service create service cases on behalf of membership accounts.
- *TagResource:* Allows the service tag resources configured as part of the service.

Managing the role

You don't need to manually create a service-linked role. When you onboard to to AWS Security Incident Response in the AWS Management Console, the AWS CLI, or the AWS API, the service creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you onboard to the service it creates the service-linked role for you again.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

AWS SLR: AWSServiceRoleForSecurityIncidentResponse_Triage

AWS Security Incident Response uses the service-linked role (SLR) named AWSServiceRoleForSecurityIncidentResponse_Triage – AWS Security Incident Response policy to continuously monitor your environment for security threats, tune security services to reduce alert noise, and gather information to investigate potential incidents.

Permissions

The AWSServiceRoleForSecurityIncidentResponse_Triage service-linked role trusts the following service to assume the role:

- `triage.security-ir.com`

Attached to this role is the AWS managed policy AWSSecurityIncidentResponseTriageServiceRolePolicy. The service uses the role to perform actions on the following resources:

- *Events:* Allows the service to create an Amazon EventBridge managed rule. This rule is the infrastructure required in your AWS account to deliver events from your account to the service. This action is performed on any AWS resource managed by `triage.security-ir.amazonaws.com`.
- *Amazon GuardDuty:* Allows the service to tune security services to reduce alert noise and gather information to investigate potential incidents. This action is performed on any AWS resource.
- *AWS Security Hub:* Allows the service to tune security services to reduce alert noise and gather information to investigate potential incidents. This action is performed on any AWS resource.

Managing the role

You don't need to manually create a service-linked role. When you onboard to to AWS Security Incident Response in the AWS Management Console, the AWS CLI, or the AWS API, the service creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you onboard to the service it creates the service-linked role for you again.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

Supported regions for AWS Security Incident Response service-linked roles

AWS Security Incident Response supports using service-linked roles in all of the regions where the service is available.

- US East (Ohio)
- US West (Oregon)
- US East (Virginia)
- EU (Frankfurt)
- EU (Ireland)
- EU (London)
- EU (Stockholm)
- Asia Pacific (Singapore)
- Asia Pacific (Seoul)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)

AWS Managed Policies

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that

provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update their associated AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

Contents

AWS managed policy: AWSSecurityIncidentResponseServiceRolePolicy

AWS Security Incident Response uses the AWSSecurityIncidentResponseServiceRolePolicy AWS managed policy. This AWS managed policy is attached to the AWSServiceRoleForSecurityIncidentResponse service-linked role. The policy provides access for AWS Security Incident Response to identify accounts subscribed, create cases, and tag related resources.

> **Important**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. AWS Security Incident Response uses tags to provide you with administration services. Tags are not intended to be used for private or sensitive data

Permissions details

The service uses this policy to perform actions on the following resources:

- *AWS Organizations:* Allows the service to lookup membership accounts for use with the service.
- *CreateCase:* Allows the service create service cases on behalf of membership accounts.
- *TagResource:* Allows the service tag resources configured as part of the service.

You can view the permissions associated with this policy in AWS managed policies for AWSSecurityIncidentResponseServiceRolePolicy.

AWS managed policy: AWSSecurityIncidentResponseFullAccess

AWS Security Incident Response uses the AWSSecurityIncidentResponseAdmin AWS managed policy. This policy grants full access to service resources and access to related AWS services. You can use this policy with your IAM principals to quickly add permissions for AWS Security Incident Response.

> **Important**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. AWS Security Incident Response uses tags to provide you with administration services. Tags are not intended to be used for private or sensitive data

Permissions details

The service uses this policy to perform actions on the following resources:

- *IAM principal read-only access:* Grants a service user the ability to perform read-only actions against existing AWS Security Incident Response resources.

- *IAM principal write access:* Grants a service user the ability to update, modify, delete, and create AWS Security Incident Response resources.

You can view the permissions associated with this policy in AWS managed policies for AWSSecurityIncidentResponseFullAccess.

AWS managed policy: AWSSecurityIncidentResponseReadOnlyAccess

AWS Security Incident Response uses the AWSSecurityIncidentResponseReadOnlyAccess AWS managed policy. The policy grants read-only access to service case resources. You can use this policy with your IAM principals to quickly add permissions for AWS Security Incident Response.

> **Important**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. AWS Security Incident Response uses tags to provide you with administration services. Tags are not intended to be used for private or sensitive data

Permissions details

The service uses this policy to perform actions on the following resources:

- *IAM principal read-only access:* Grants a service user the ability to perform read-only actions against existing AWS Security Incident Response resources.

You can view the permissions associated with this policy in AWS managed policies for AWSSecurityIncidentResponseReadOnlyAccess.

AWS managed policy: AWSSecurityIncidentResponseCaseFullAccess

AWS Security Incident Response uses the AWSSecurityIncidentResponseCaseFullAccess AWS managed policy. The policy grants full access to service case resources. You can use this policy with your IAM principals to quickly add permissions for AWS Security Incident Response.

> **Important**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. AWS Security Incident Response uses tags to provide you with administration services. Tags are not intended to be used for private or sensitive data

Permissions details

The service uses this policy to perform actions on the following resources:

- *IAM principal case read-only access:* Grants a service user the ability to perform read-only actions against existing AWS Security Incident Response cases.
- *IAM principal case write access:* Grants a service user the ability to update, modify, delete, and create AWS Security Incident Response cases.

You can view the permissions associated with this policy in AWS managed policies for AWSSecurityIncidentResponseCaseFullAccess.

AWS managed policy: AWSSecurityIncidentResponseTriageServiceRolePolicy

AWS Security Incident Response uses the AWSSecurityIncidentResponseTriageServiceRolePolicy AWS managed policy. This AWS managed policy is attached to the AWSServiceRoleForSecurityIncidentResponse_Triage service-linked role.

The policy provides access to AWS Security Incident Response to continuously monitor your environment for security threats, tune security services to reduce alert noise, and gather information to investigate potential incidents. You can't attach this policy to your IAM entities.

> **Important**
>
> Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. AWS Security Incident Response uses tags to provide you with administration services. Tags are not intended to be used for private or sensitive data

Permissions details

The service uses this policy to perform actions on the following resources:

- *Events:* Allows the service to create an Amazon EventBridge managed rule. This rule is the infrastructure required in your AWS account to deliver events from your account to the service. This action is performed on any AWS resource managed by `triage.security-ir.amazonaws.com`.
- *Amazon GuardDuty:* Allows the service to tune security services to reduce alert noise and gather information to investigate potential incidents. This action is performed on any AWS resource.
- *AWS Security Hub:* Allows the service to tune security services to reduce alert noise and gather information to investigate potential incidents. This action is performed on any AWS resource.

You can view the permissions associated with this policy in AWS managed policies for AWSSecurityIncidentResponseTriageServiceRolePolicy.

AWS Security Incident Response updates to SLRs and managed policies

View details about updates to AWS Security Incident Response SLRs and managed policies roles since this service began tracking these changes.

Change	Description	Date
New SLR – AWSServiceRoleForSecurityIncidentResponse New managed policy – AWSSecurityIncidentResponseServiceRolePolicy.	New service linked role and attached policy allowing service access into your AWS Organizations accounts to identify membership.	December 1, 2024
New SLR – AWSServiceRoleForS	New service linked role and attached policy allowing service access into your AWS Organizations accounts to perform triage of security events.	December 1, 2024

Change	Description	Date
ecurityIncidentResponse_Triage New managed policy – AWSSecurityIncidentResponseTriageServiceRolePolicy		
New managed policy – AWSSecurityIncidentResponseFullAccess	AWS Security Incident Response add a new SLR to attach to IAM principals for read and write actions for the service.	December 1, 2024
New managed policy role – AWSSecurityIncidentResponseReadOnlyAccess	AWS Security Incident Response add a new SLR to attach to IAM principals for read actions	December 1, 2024
New managed policy role – AWSSecurityIncidentResponseCaseFullAccess	AWS Security Incident Response add a new SLR to attach to IAM principals for read and write actions for service cases.	December 1, 2024
Started tracking changes.	Started tracking changes for AWS Security Incident Response SLRs and managed policies	December 1, 2024

Incident response

Security and Compliance is a shared responsibility between AWS and the customer. This shared model can help relieve the customer's operational burden as 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. The customer assumes responsibility and management of the guest operating system (including updates and security patches), other associated application software as well as the configuration of the AWS provided security group firewall. For additional information refer to the AWS shared responsibility model.

By establishing a security baseline that meets the objectives for your applications running in the cloud, you're able to detect deviations that you can respond to. Since security incident response can be a complex topic, we encourage you to review the following resources so that you are better able to understand the impact that incident response and your choices have on your corporate goals: AWS Security Best Practices whitepaper, and the Security Perspective of the AWS Cloud Adoption Framework (CAF) white paper.

Compliance validation

Third-party auditors assess the security and compliance of AWS services as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

AWS Security Incident Response has not been evaluated for compliance with aforementioned programs.

For a list of AWS services in scope of specific compliance programs, see AWS services in scope by compliance program. For general information, see AWS compliance programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable lAWS and regulations. AWS provides the following resources to help with compliance:

- Security and compliance quick start guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.

- Architecting for HIPAA security and compliance whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS compliance resources – A collection of workbooks and guides that apply by industry and/or location.
- Evaluating resources with AWS Config Rules in the AWS Config Developer Guide – AWS Config; assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Logging and monitoring in AWS Security Incident Response

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Security Incident Response and your other AWS solutions. AWS Security Incident Response currently supports the following AWS services to monitor your organization and the activity that happens within it.

AWS CloudTrail – With CloudTrail you can capture API calls from the AWS Security Incident Response console. For example, when a user authenticates, CloudTrail can record details such as the IP address in the request, who made the request, and when it was made.

Amazon CloudWatch Metrics – With CloudWatch metrics you can monitor, report, and take automatic actions in case of an event in near real time. For example, you can create CloudWatch dashboards on the provided metrics to monitor your AWS Security Incident Response usage, or you can create CloudWatch alarms on the provided metrics to notify you on breach of a set threshold.

The namespace for the service is AWS/Usage/ServiceName. The metric names available are ActiveManagedCases and SelfManagedCases.

In accordance with the AWS Service Terms, The AWS Security Incident Response responder team will have access to your history of CloudTrail, VPC, DNS and S3 log data. This data may be utilized during active security incidents when a case is open in the AWS Security Incident Response service portal.

Resilience

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS global infrastructure.

Infrastructure security

AWS Security Incident Response is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access AWS Security Incident Response through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Configuration and vulnerability analysis

You are responsible for managing the service containment roles and the associated AWS CloudFormation stack sets.

AWS handles basic security tasks, such as guest operating system (OS) and database patching, firewall configuration, and disaster recovery. These procedures have been reviewed and certified by the appropriate third parties. For more details, see the following AWS resources:

- Shared responsibility model
- Best practices for security, identity, & compliance

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the AWS:SourceArn and AWS:SourceAccount global condition context keys in resource policies to limit the permissions that Amazon Connect gives another service to the resource. If you use both global condition context keys, the AWS:SourceAccount value and the account in the AWS:SourceArn value must use the same account ID when used in the same policy statement.

The most effective way to protect against the confused deputy problem is to use the exact Amazon Resource Name (ARN) of the resource you want to allow. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the AWS:SourceArn global context condition key with wildcards (*) for the unknown portions of the ARN. For example, arn:AWS:servicename::region-name::your AWS account ID:*.

For an example of an assume role policy that shows how you can prevent a confused deputy issue, see Confused deputy prevention policy.

Service Quotas

AWS Security Incident Response

The following tables list the quotas, for AWS Security Incident Response resources for your AWS-account;. Some quotas may be increased above those stated below with service manager approval. Unless indicated otherwise, these quotas are per Region.

	Name	Default	Adjustable	Comments
1	Active AWS supported cases	10	Yes (up to 50)	The number of active cases requesting assistance from AWS CIRT.
2	Active self-managed cases	50	Yes (up to 100)	The number number of active cases using the platform without assistance from AWS CIRT.
3	Service supported cases created within 24 hours	10	No	The number of cases created requesting assistance from AWS CIRT created in a 24-hour rolling window.
4	Maximum number of entities in	10	No	The maximum number of entities in the

	Name	Default	Adjustable	Comments
	default incident response team			default incident response team.
5	Maximum number of additional members on a case	30	No	The maximum number of entities associated with a case. This will initially be populated with entities from your default incident response team.
6	Maximum Number of Case Attachments	50	Yes (up to 100)	The maximum number of files that can be attached to a case.
7	Maximum case comment size	1000	No	The maximum number of characters in a case comment.
8	Maximum Case Attachment filename size	255	No	The maximum number of characters in a filename.

AWS Security Incident Response Technical Guide

Contents

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 guide 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

Well-Architected Tool 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 incident response for security events 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 prerequisite. 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

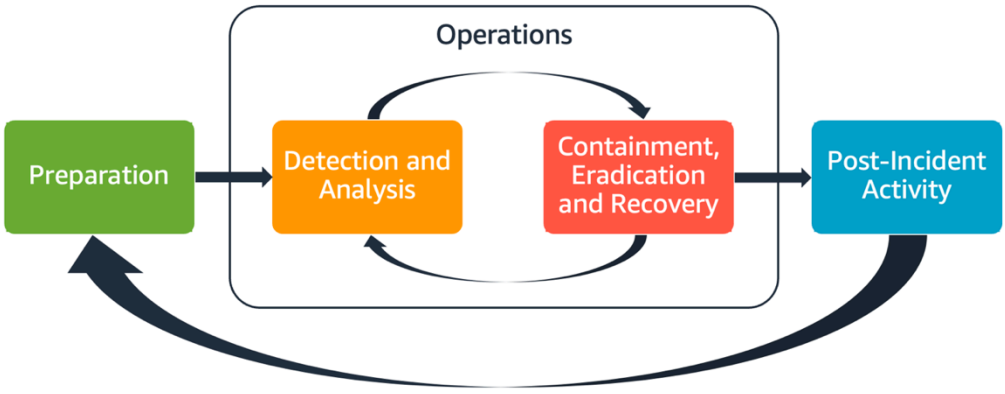
All AWS users within an organization should have a basic understanding of security incident response processes, and security staff should understand how to respond to security issues. Education, training, and experience are vital to a successful cloud incident response program and are ideally implemented 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 the 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 the section called “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 a 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 an 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.

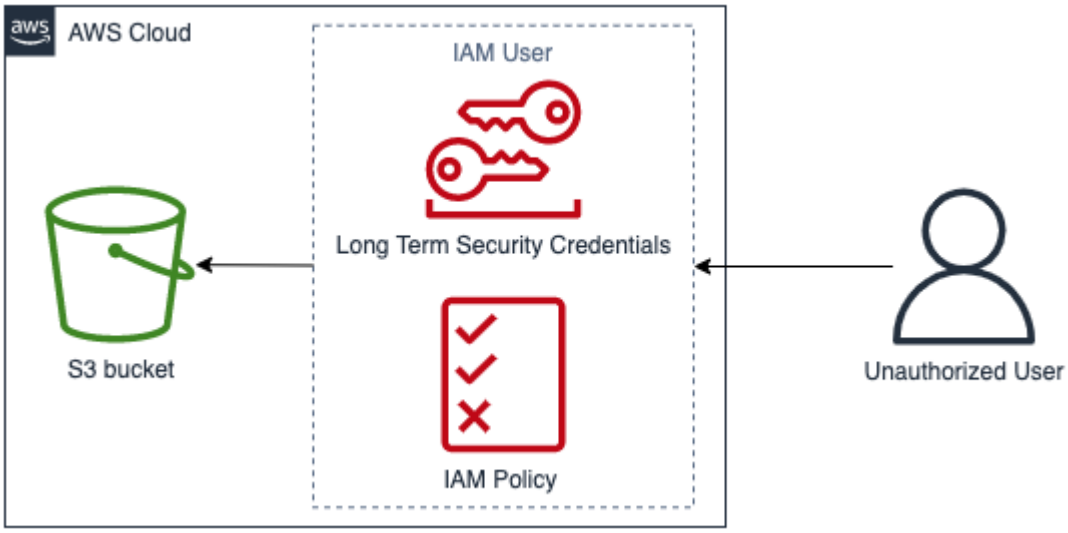
In addition to the direct relationship you have with AWS, there might be other entities that have responsibilities in your particular responsibility model. For example, you might have internal organizational units that take responsibility for some aspects of your operations. You might also have relationships with 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 automatic scaling, 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 relationships 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, see 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, AWS training 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 Tust and Safety team.

AWS Customer Incident Response Team (CIRT)

The AWS Customer Incident Response Team (CIRT) is a specialized always available 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.

- **All Customers**:
  1. Account and billing
  2. Service: Account
  3. Category: Security
  4. Severity: General question

- **Customers with Developer AWS Support plans**:
  1. Account and billing
  2. Service: Account
  3. Category: Security
  4. Severity: Important question

- **Customers with Business AWS Support plans**:
  1. Account and billing

2. Service: Account
3. Category: Security
4. Severity: Urgent business impacting question

- **Customers with Enterprise AWS Support plans**:
  1. Account and billing
  2. Service: Account
  3. Category: Security
  4. Severity: Critical business risk question

- **Customers with AWS Security Incident Response subscriptions**: Open the Security Incident Response console at https://console.aws.amazon.com/security-ir/

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 an every day 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 an 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 applications 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, a 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 know 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	A more realistic offering compared to tabletop exercises . During Purple Team exercises, facilitators	Generally a more advanced simulatio n offering. There is usually a high level of covertness, where

	Tabletop exercise	Purple Team exercise	Red Team exercise
	be either high-level or technical, and are driven by a series of paper injects.	work collaboratively with the participants to increase exercise engagement and offer training where necessary.	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 the scope, and take action in a timely manner.

Develop AWS account structure

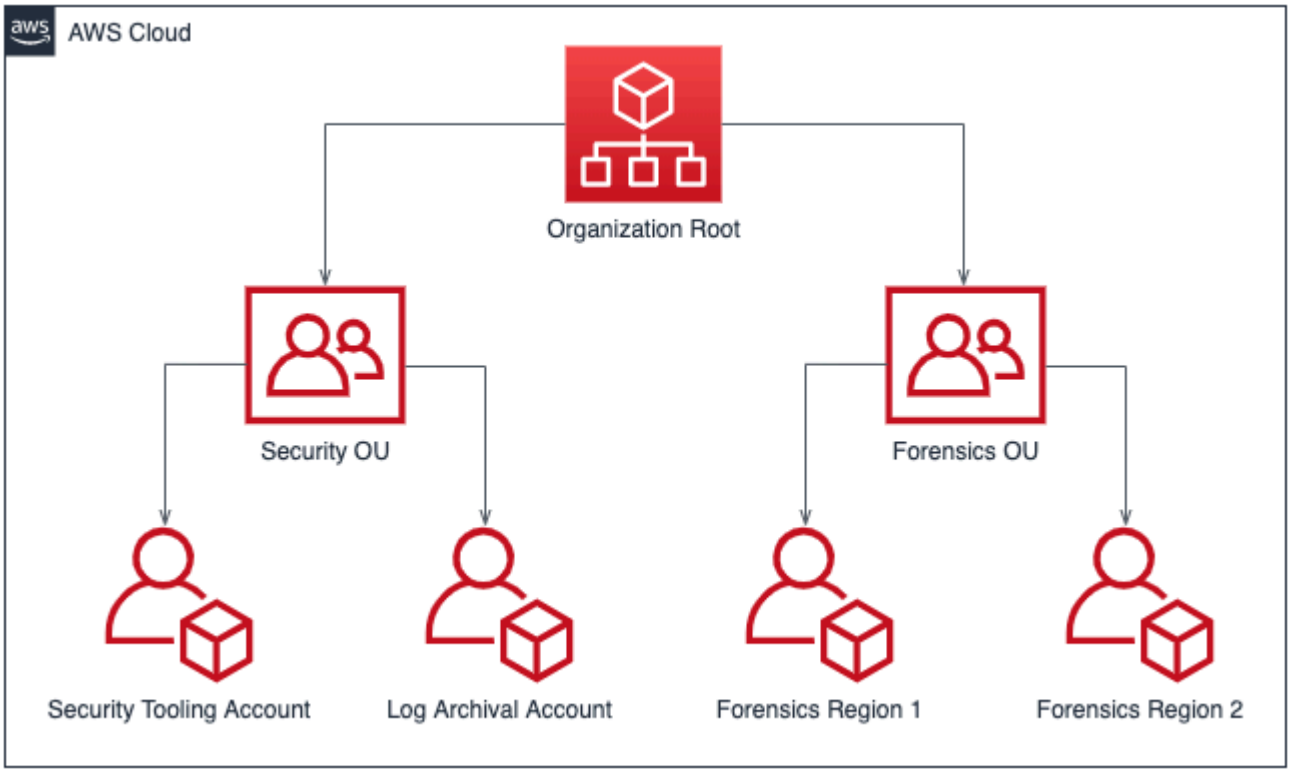
AWS Organizations helps centrally manage and govern an AWS environment as you grow and scale AWS resources. An AWS organization consolidates your AWS accounts so that you can administer them as a single unit. You can use organizational units (OUs) to group accounts together to administer as a single 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 the section called "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. • 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

Domain	Preparation item	Action items
		(CIRT), DDoS response team (DRT) and AMS. • Understand the 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, communication plan, incident definitions, 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.
Process	Develop incident response playbooks.	• Create a template for the structure of your playbooks. • Build playbooks for expected security events. • Build playbooks for known security alerts, such as GuardDuty findings.

Domain	Preparation item	Action items
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 an 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 the 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 a 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 a security event is an incident and assess the scope of the incident.
Containment	Minimize and limit the scope of the security event.

Phase	Goal
Eradication	Remove unauthorized resources or artifacts related to the security event. Implement mitigations that caused the security incident.
Recovery	Restore systems to a 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 the section called "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.

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 a 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 can 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 the 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 the section called "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 a 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 down the system 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 nearly 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 includes:

- 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:

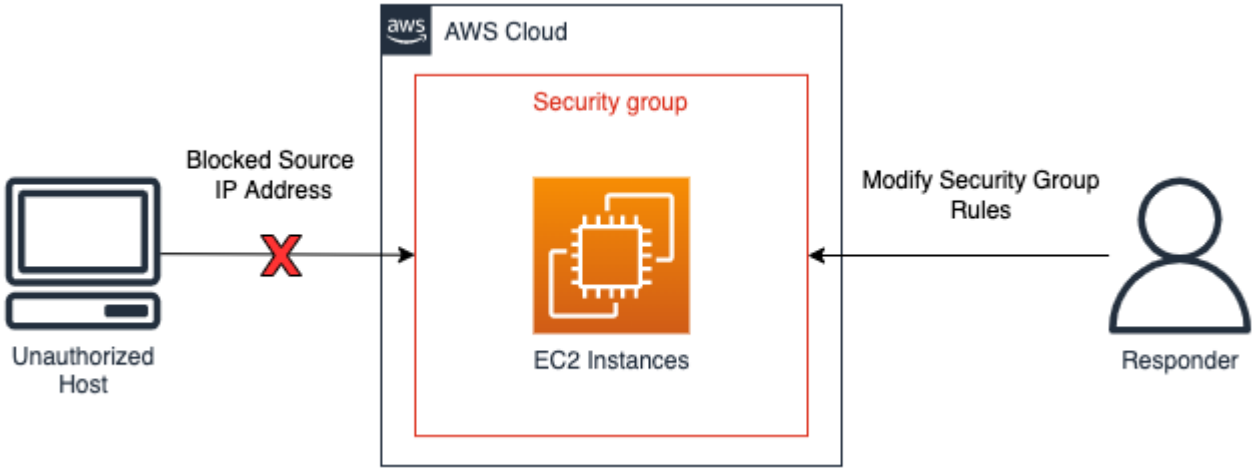
- **Source containment** – Use filtering and routing to prevent access from a certain source.
- **Technique and access containment** – Remove access to prevent unauthorized access to the affected resources.
- **Destination containment** – Use filtering and routing to prevent access to a target resource.

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

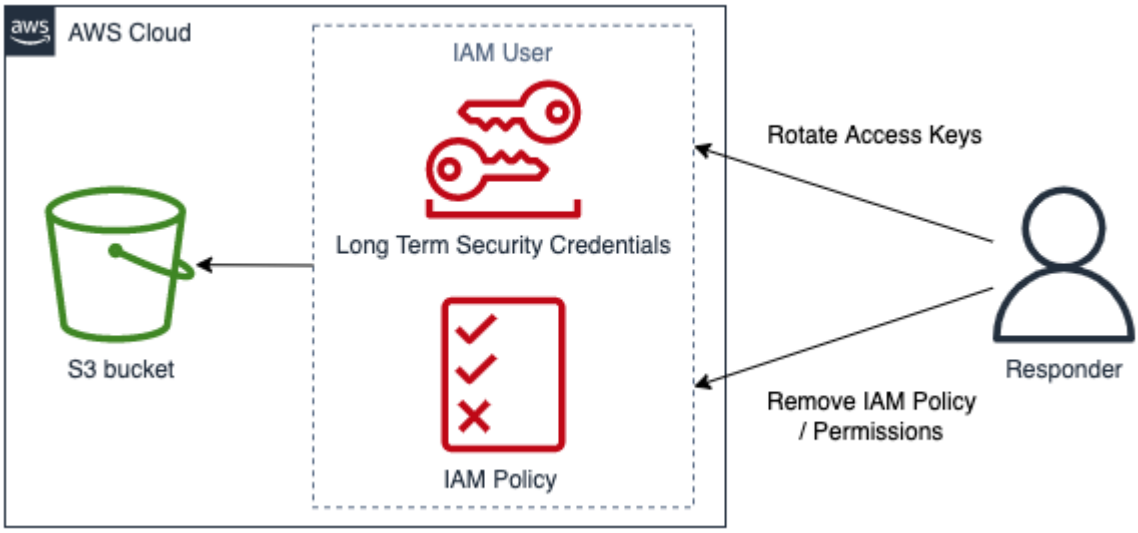
Technique and access containment

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 the 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 the IAM user that owns the compromised access keys. Re-create the user if needed.
- **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

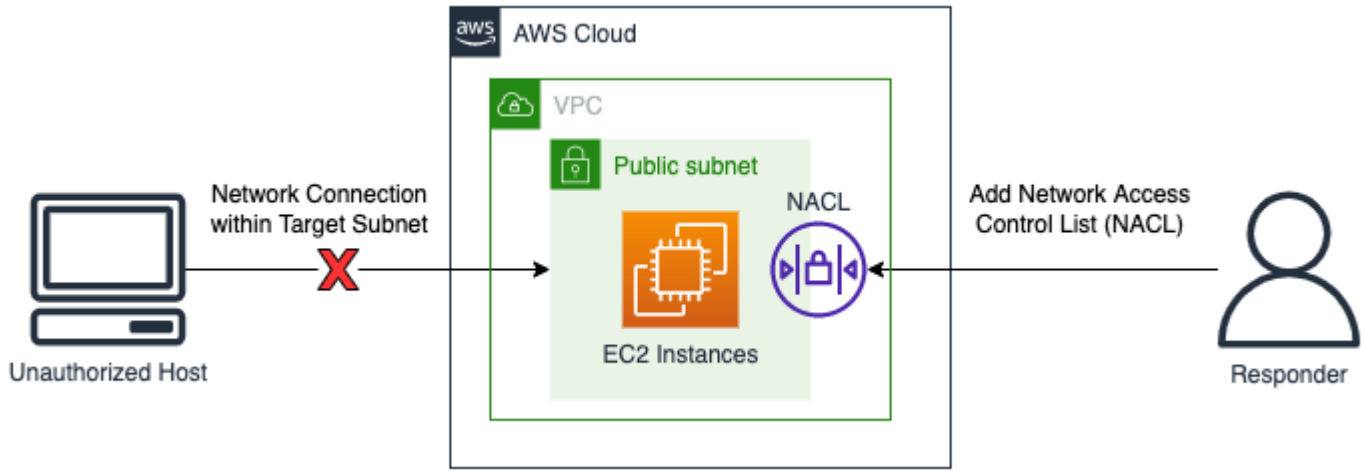
Destination containment is the application of filtering or routing within an environment to prevent access to a targeted host or resource. In some cases, destination containment also involves a form of resiliency to verify that legitimate resources are replicated for availability; resources should be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

- **Network ACLs** – Network ACLs (network ACLs) 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 network access control list (network ACL) will affect every resource on the subnet, not only the resources that are being accessed without authorization. Rules listed within an network ACL are processed in top-down order, so the first rule in an existing network ACL should be configured to deny unauthorized traffic to the targeted resource and subnet. Alternatively, a completely new network ACL 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 network ACL.
- **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 network ACL 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. However, 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 a 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
Recovery	Restore systems to a known good state and monitor these	• System restoration from backups

Phase	Goal	Techniques and methodologies
	systems to ensure the threat does not return.	• 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. The 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 the 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 the 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 an effective response. Refer to the the section called “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 the 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 the 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 that 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 a 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.

Continuous 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 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 Engineer, 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 Engineer, Amazon Web Services
- Ryan Tick, Senior Security Engineer, Amazon Web Services
- Steve de Vera, Senior Security Engineer, 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

- Future and Custom Security Capabilities

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.

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 EventBridge – Amazon EventBridge 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. EventBridge becomes aware of operational changes as they occur. EventBridge 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 EventBridge events. Many security services also provide an option to send their outputs to Amazon S3.

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 fulfill compliance and auditing requirements, as well as debugging and forensics. These logs help customers determine the 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 that 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.

Future and Custom Security Capabilities

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 Λ 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.

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.

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Document history

Change	Description	Date
Updated: Updates from customer comments on docs.	Removed multiple duplicate AWS AWS in text. Fixed broken URl links on https://docs.aws.amazon.com/security-ir/latest/userguide/sir_tagging.html and https://docs.aws.amazon.com/security-ir/latest/userguide/service-name-info-in-cloudtrail.html . Updates to https://docs.aws.amazon.com/security-ir/latest/userguide/contain.html. Removed the > from first paragraph . Replaced AWSSupport-ContainEC2Reversible with AWSSupport-ContainEC2Instance. Replaced AWSSupport-ContainIAMReversible with AWSSupport-ContainIAMPrincipal. Replaced AWSSupport-ContainS3Reversible with AWSSupport-ContainS3Resource. Updated formatting on https://docs.aws.amazon.com/en_us/security-ir/latest/userguide/issues.html	December 10, 2024

Change	Description	Date
	When telling customers to contact CIRT via a support ticket, https://docs.aws.amazon.com/security-ir/latest/userguide/understand-response-teams-and-support.html now provides options to select in the support forms. Removed CloudWatch Events and replaced with EventBridge on https://docs.aws.amazon.com/security-ir/latest/userguide/logging-and-events.html . Grammar updates on https://docs.aws.amazon.com/security-ir/latest/userguide/technique-access-containment.html . Removed publication date from https://docs.aws.amazon.com/security-ir/latest/userguide/security-incident-response-guide.html, replaced by updates in this table.	
Updated: AWS managed policies and service-linked roles.	Updates to managed policies and service-linked roles.	December 1, 2024

Change	Description	Date
Service Launch	Initial service docs for service launch at re:Invent 2024	December 1, 2024