# Incident Response Playbook Template

## Incident Type

GuardDuty Finding: Impact:IAMUser/AnomalousBehavior

## Introduction

This playbook is provided as a template to customers using AWS products and who are building their incident response capability. You should customize this template to suit your particular needs, risks, available tools and work processes.

Security and Compliance is a shared responsibility between you and AWS. AWS is responsible for “Security of the Cloud”, while you are responsible for “Security in the Cloud”. For more information on the shared responsibility model, please review our documentation (https://aws.amazon.com/compliance/shared-responsibility-model/).

You are responsible for making your own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) references 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. This document is 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.

# Summary

## This Playbook

This playbook outlines response steps for incidents involving anomolous API usage from an AWS IAM resource. These steps are based on the NIST Computer Security Incident Handling Guide (https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf) (Special Publication 800-61 Revision 2) that can be used to:

- Gather evidence
- Contain and then eradicate the incident
- Recover from the incident
- Conduct post-incident activities, including post-mortem and feedback processes

Interested readers may also refer to the AWS Security Incident Response Guide (https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/welcome.html) which contains additional resources.

Once you have customized this playbook to meet your needs, it is important that you test the playbook (e.g., Game Days) and any automation (functional tests), update as necessary to achieve the desired results, and then publish to your knowledge management system and train all responders.

Note that some of the incident response steps noted in each scenario may incur costs in your AWS account(s) for services used in either preparing for, or responding to incidents. Customizing these scenarios and testing them will help you to determine if additional costs will be incurred. You can use AWS Cost Explorer (https://aws.amazon.com/aws-cost-management/aws-cost-explorer/) and look at costs incurred over a particular time frame (such as when running Game Days) to establish what the possible impact might be.

In reviewing this playbook, you will find steps that involve processes that you may not have in place today. Proactively preparing for incidents means you need the right resource configurations, tools and services in place that allow you to respond to an incident. The next section will provide a summary of this incident type, and then cover the five steps (parts 1 - 5) for handling credential compromise.

# This Incident Type

An Amazon GuardDuty finding represents a potential security issue detected within your network. GuardDuty generates a finding whenever it detects unexpected and potentially malicious activity in your AWS environment.

**Impact:IAMUser/AnomalousBehavior (https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-iam.html#impact-iam-anomalousbehavior)**

**An API commonly used to tamper with data or processes in an AWS environment was invoked in an anomalous way.**

This finding informs you that an anomalous API request was observed in your account. This finding may include a single API or a series of related API requests made in proximity by a single user identity. The API observed is commonly associated with impact tactics where an adversary is trying to disrupt operations and manipulate, interrupt, or destroy data in your account. APIs for this finding type are typically delete, update, or put operations, such as, DeleteSecurityGroup, UpdateUser, or PutBucketPolicy.

Details on which factors of the API request are unusual for the user identity that invoked the request can be found in the finding details (https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings-summary.html#finding-anomalous).

Reasons for unexpected API usage could include the following:

- The user is new and their behavior has no established baseline
- The user has new responsibilities and requires activity outside the established baseline
- The user has broad permissions and was acting on a request to perform the detected activity
- *The user has broad permissions and was engaged in unauthorized or malicious activity*

**Note** This process focuses on AWS IAM Users as principals, but the same process is valid for AWS IAM Roles.

# Incident Response Process

## **Part 1**: Acquire, Preserve, Document Evidence

### For Any Incident

1. You become aware of potential indicators of compromise (IoCs). These could come in various forms, but the original source is a GuardDuty finding:
    - An internal ticketing system (the sources of the ticket are varied and could include any of the means below)
    - From an alert in one of your monitoring systems either inside or external to AWS (that are ingesting GuardDuty Findings, in AWS, this could include AWS Security Hub)
    - Alarms or observations that resources have been created or deleted in your account that cannot be accounted for in your CMDB, exist in regions that you do not operate infrastructure in, or themselves have generated alerts (Amazon Detective (https://aws.amazon.com/detective/getting-started/) is a useful tool for understanding these relationships)
2. Confirm a ticket/case has been raised for the incident. If not, manually raise one
3. Determine and begin to document any end-user impact/experience of the issue. Findings should be documented in the ticket/case related to the incident
4. Open an incident war room using your standard communication tools, and page in the relevant stakeholders
5. In the case of automatically created tickets/cases, verify the finding in GuardDuty (what caused the ticket to be created?)

### For This Incident Type

1. Identify the specific resources impacted. In GuardDuty, this will be "Resource ID" in the “Overview” section of the finding.
2. Identify the nature of the impact (modified, deleted, etc.). In GuardDuty, Look at the “Anomolous APIs” section of the finding. Look up each in the AWS API Reference (https://docs.aws.amazon.com/quicksight/latest/APIReference/API_Reference.html)
3. Determine if the Anomolous API call was successfull. In GuardDuty, Look at the “Anomolous APIs” section of the finding. There are two types of subcategories: Successfully called and Error Response
4. Identify the origination information from which the APIs were called. In GuardDuty, Look at the “Actor” section of the finding
    - Verify that the IP address is valid for your enterprise users.
    - Verify that the Location or Organization is known or valid
5. Identify the owner or custodian of the *impacted* resource(s) and add them to the ticket
6. Determine what, if any, data was modified or deleted, and the date and time.
7. Identify the person or system associated the principal that resulted in the GuardDuty finding being generated

- If an IAM user, query the user details in IAM (https://us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users)
- If an IAM role, verify the principal that assumed the role. For roles, this may involve linking the corporate identity to the assumed role session in the relevant CloudTrail entry (https://aws.amazon.com/blogs/security/how-to-relate-iam-role-activity-to-corporate-identity/)

8. Contact the person associated with the principal (**Actor**) to validate the activity. At this point we assume this incident is based on alert generated by GuardDuty and not confirmed malicious activity. Do these steps now if the activities identified by the alert are presumed to be potentially authorized. If the changes made by the Actor are presumed as unauthorized for any reason, delay this until **Part 5**. These presumption should be based up your organization's accepted change practices and risk tolerance.
    1. Collate the information obtained in previous steps.
    2. Notify the the Actor's manager (or note in the ticket) of your intent to inverview the person linked to the AWS principal
    3. Interview the Actor to validate whether further investigation is necessary.
        - The Actor should be able to say what they did and why they did it.
        - If it exists, obtain and document evidence that the activity was authorized. Such evidence may include:
            - Change records in the official change management system.
            - Written instructions from the actor's manager, the resource owners or their managers, etc.
            - Documented request from Incident responders including an Incident record number.
    4. Verify evidence provided with their respective sources.
    5. Notify the actor's management of the outcome of the interview.

# **Part 2**: Contain the Incident

If you determine that the activity is unauthorized, or decide it is prudent to assume so, the first priority is to prevent further compromise without impact to production workloads.

1. Verify that disabling the credentials of the IAM principal in question will not result in a service outage.
2. Record all permissions from the principal responsible for the activity.
3. Remove permissions from the principal responsible for the activity.
    1. If the Principal is an AWS Role, only remove permissions if no other entities need to use it. Instead, do one of the following:
        1. Remove the the Actor identified above from whatever authorization system approves the access (Active Directory security groups, etc.)
        2. Attach an inline policy with a Condition that excludes an ARN with the above Actors Session Identifier Use an IAM policy Conditions tp limit access (https://aws.amazon.com/blogs/security/iam-makes-it-easier-to-manage-permissions-for-aws-services-accessing-resources/)
4. Disable any current Access Keys for the principal responsible for the activity
5. Change the password (or remove console access) for the principal responsible for the activity
6. If required, retain user action logs (such as CloudTrail) further forensic analysis
7. Continue to update stakeholders on the current state of the incident response. These may include:
    - Technology Risk organization

- Data owner and management
- General Counsel and/or Privacy Officer
- Manager of the entity whose credentials were used in the incident

# **Part 3**: Eradicate the Incident

This is the stage for taking remedial action to minimize the impact of the unauthorized activities.

1. Using your preferred monitoring tool, access CloudTrail and search for all API actions performed by the principal in the last 90 days:

    1. If this tool is a third-party tool such as Splunk, Sumo Logic or others, follow the normal procedure for obtaining log information from that tool
    2. If you do not ingest CloudTrail logs into a third-party tool, but do send those logs to Amazon Simple Storage Service (Amazon S3), you will be able to use Amazon Athena to query the logs. The remaining steps will focus on AWS tools to retrieve the necessary information

2. Create an Athena table referencing the bucket containing your CloudTrail logs (https://aws.amazon.com/premiumsupport/knowledge-center/athena-tables-search-cloudtrail-logs/) that link also includes example queries that can be run in Athena

3. In the Athena console, run a query that shows all API actions taken by the compromised credentials post-compromise date/time. From the resulting list, determine which API calls:

    - Accessed sensitive data, such as S3 GetObject (NOTE! To do this, the trail will need to have data events configured for the relevant bucket(s))
    - Created new AWS resources, such as databases, Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS Lambda functions or S3 buckets, etc.
    - Services that create resources should also be carefully checked; for example, CloudFormation Stacks/StackSets, AWS Firewall Manager Security Policies, AWS Elastic Beanstalk resources, etc.
    - Created or modified AWS identity resources that could be used to extend a foothold into the account (or other accounts, for example with AWS Security Token Service (AWS STS) API methods such as AssumeRole). Within an account, API methods including (but not limited to) the following should also be investigated:
        - CreateUser
        - CreateRole
        - AssumeRole*
        - Get*Token
        - Attach*Policy
        - RunInstances (especially with PassRole)
        - *Image*
        - *Provider
        - Tag*
        - Untag*
        - Create*

        - Delete*
        - Update*
        - etc.
    - Deleted existing AWS resources
    - Modified existing AWS resources

4. Based on the results of the previous step, determine if any applications are potentially impacted:

    1. Obtain the ARN and/or tag information for each resource impacted (from step 4, above)
    2. Go back to the CMDB and determine which application that resource belongs to
    3. Notify the application owner based on the results of the above steps

5. Based on the above results, if additional credential-obtaining resources have been created (IAM users, roles, EC2 instances with roles, etc.), take the following steps:

    1. Disable and/or delete any credentials for those resources, as per the steps outlined in Part 2, Step 1
    2. Use your preferred logging tool to analyze CloudTrail again once any such credentials have been deleted or disabled, this time specifying the additionally discovered credentials as a query parameter, as outlined in Part 3, steps 1 through to 6 (this step). Continue to iterate through this process until you discover that there were no further resources created capable of obtaining valid credentials for the AWS account.
    3. For each set of newly discovered compromised credentials, repeat the above process

# **Part 4**: Recover from the Incident

1. For resources that were modified during the compromise:
    1. If the resource can be destroyed and replaced, do so. For example, EC2 instances in an EC2 Auto Scaling group, with a Launch Configuration referencing a non-compromised AMI, or Launch Templates used to create EC2 instances, terminate the instance, allow EC2 Auto Scaling to adjust capacity as required
    2. If the resource cannot be replaced, either:
        1. Restore the resource from a known good backup, or;
        2. Prepare a new resource and configure it into the application's infrastructure, while isolating the compromised resource and removing it from the application's infrastructure. Update the CMDB accordingly
        3. Either destroy the compromised resource, or continue to leave it isolated for post-incident forensics
2. For resources that were deleted during the compromise:
    1. Determine what (if any) application the resource belonged to, by checking in the CMDB, or confirming the resource's tag(s) (Check AWS Config if the tags aren't listed in the CloudTrail entry and the resource is supported by AWS Config (https://docs.aws.amazon.com/config/latest/developerguide/resource-config-reference.html))
    2. If the deleted resource can be restored from backup, commence the restore procedure
    3. If you are using immutable architectures or similar deployment methods, redeploy the relevant resources
    4. If neither of the above options are possible, consult the CMDB to obtain the resource's configuration, and recreate the resource and configure it into the application's infrastructure.

5. If the CMDB information is inadequate or non-existent, assign resource creation to the relevant personnel in your organization

3. For resources that were created during the compromise:
    1. Confirm these have been deleted or isolated for further analysis, per the steps in **Part 3**.
4. Validate that the affected resources were remediated and the expected state of any impacted systems were restored

# **Part 5**: Post-Incident Activity

This activity contains two parts. Firstly, some compromised resources may require forensic analysis, either to fulfil regulatory obligations or improve incident handling, both taking input from the root cause analysis that will result from forensic investigation. The second part is a “sharpen the saw” activity which helps teams to assess their response to the actual incident, determine what worked and what didn’t, update the process based on that information and record these findings.

Firstly, perform any required forensic investigation to determine (for compromised resources) what methods the actors may have used and to determine if additional risks and risk mitigations are required for the resources and/or applications in question.

1. For any compromised resources that have been isolated for further analysis, perform the forensic activity on those resources and incorporate the findings into the post-incident report.
2. Ensure that the CMDB is correctly updated to reflect the current status of all resources and applications impacted

Secondly, review the incident itself and the response to it, to determine if anything needs to be changed for handling any similar incidents in the future.

1. Review the incident handling and the incident handling process with key stakeholders identified in Part 1, Step 8.
2. Document lessons learned, including attack vector(s) mitigation(s), misconfiguration, etc.
3. Store the artifacts from this process with the application information in the CMDB entry for the application and also in the CMDB entry for the credential compromise response process.

# Additional Resources

## Incident Response

https://aws.amazon.com/blogs/security/how-get-started-security-response-automation-aws/
(https://aws.amazon.com/blogs/security/how-get-started-security-response-automation-aws/)
https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/welcome.html
(https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/welcome.html)
https://aws.amazon.com/blogs/security/how-to-automate-incident-response-in-aws-cloud-for-ec2-instances/
(https://aws.amazon.com/blogs/security/how-to-automate-incident-response-in-aws-cloud-for-ec2-instances/)
https://aws.amazon.com/blogs/security/forensic-investigation-environment-strategies-in-the-aws-cloud/

(https://aws.amazon.com/blogs/security/forensic-investigation-environment-strategies-in-the-aws-cloud/)

https://aws.amazon.com/blogs/security/how-to-automate-forensic-disk-collection-in-aws/

(https://aws.amazon.com/blogs/security/how-to-automate-forensic-disk-collection-in-aws/)

# GuardDuty

https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings-summary.html

(https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings-summary.html)

# Other

https://docs.aws.amazon.com/quicksight/latest/APIReference/API_Reference.html

(https://docs.aws.amazon.com/quicksight/latest/APIReference/API_Reference.html)

https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

(https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html)