# Incident Response Playbook Template

## Incident Type

Credential Leakage/Compromise

## 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 Credential Leakage/Compromise incidents. 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

Credential compromise occurs when credentials related to one or more of your IAM principals (such as an IAM user or role) have been obtained by an actor not authorized to use them. This means that the actor is either *not* the IAM user whom the credentials identify, or is not authorized to assume the IAM role that a set of temporary credentials are associated with. Once a malicious actor has obtained a set of credentials, they can use those credentials to perform any action that is allowed by IAM policies associated with those credentials. This may include the ability to elevate privilege if those allowed actions include various IAM API actions, such as PassRole, CreateUser or Attach*Policy. As a preventative measure, you can use AWS IAM Permission Boundaries and AWS Organizations Service Control Policies (SCPs) to scope down user permissions and prevent potential privilege escalations.

# Incident Response Process

## Part 1: Acquire, Preserve, Document Evidence

1. You become aware of potential indicators of compromise (IoCs). These could come in various forms:

    - An internal ticketing system (the sources of the ticket are varied and could include any of the means below)
    - A message from a contractor or third-party service provider
    - From an alert in one of your monitoring systems either inside or external to AWS (for example, in AWS, in this particular situation, it could be via an Amazon GuardDuty finding, either directly in GuardDuty, or via AWS Security Hub, or it could be via a CloudWatch metric indicating a change in IAM (https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html#cloudwatch-alarms-for-cloudtrail-iam-policy-changes) or from your own systems ingesting AWS data)
    - Alarms or observations that resources have been created 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)
    - Via an anonymous tip
    - Via independent or external security researchers
    - Where credentials have been compromised, you are less likely to be contacted via the threat actor, as in this situation typically they want to maximize their use of resources and minimize the chance of being discovered, unless the compromise is being used to exfiltrate data or cause some other kind of public incident such as through the modification of your publicly facing resources

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. From a user's perspective, for this type of scenario, there may be no direct user impact. Findings should be documented in the ticket/case related to the incident

4. In the case of automatically created tickets/cases, determine what internal alarms/metrics are currently indicating an issue (what caused the ticket to be created?) This may be an Amazon CloudWatch metric indicating an API was called that included a reference to an AWS policy, or an AWS Config rule that indicates some aspect of your AWS Identity and Access Management (IAM) configuration has fallen out of compliance, or an Amazon GuardDuty alert that indicates possible credential compromise. It may also be a billing alert where your billing costs have passed a predetermined threshold, triggering an alarm and notification. Determine exactly which of these has caused the ticket/case to be raised

5. Determine the set of credentials that have been compromised. Note that there will be additional steps in Part 3 to deal with scenarios where threat actors have planted resources to gain an account foothold.

    1. If an internal ticket/case has been created, review the case to determine if the user/role name or ARN, user or role ID, or Access Key ID are provided in that ticket. If it is not, reference the alert specified in the ticket (for example, a GuardDuty finding) and go to point 2., below
    2. If the alert has come from GuardDuty or AWS Security Hub, locate the specific event in the service's AWS console and then locate the Access Key ID for the impacted credentials. In GuardDuty, this will be located under the "Resource" section of the finding. "resourceType" should be Access Key and there will be a field named accessKeyDetails that will contain the Access Key ID, Principal ID, User Type and additional information

6. Determine the likely time at which the credentials were compromised (any API actions taken post that time should be considered as malicious and any resources created post that time should be considered compromised). As an example, for a GuardDuty finding, note the "eventFirstSeen" field in the "Service" section of the finding.

7. If there is service disruption for your application occurring (see point 2, above), determine if there are any known events that could be causing that disruption that may not be related to the credential leakage (deployed CRs or other application modifications, for example). Check the deployment pipeline to determine if any changes have been made leading up to the event (the time the event was first recorded, or the time the automated system cut the ticket/case)

8. Incident Communications:

    1. Identify stakeholder roles from the application entry in the Configuration Management Database (CMDB) entry for that application, or via the application's risk register
    2. Open a conference bridge war room for the incident
    3. Notify identified stakeholders including (if required) legal personnel, technical teams and developers and add them to the ticket and the war room, so they are updated as the ticket is updated

9. External Communications:

    1. Ensure your organizations legal counsel is informed and is included in status updates to internal stakeholders and especially in regards to external communications.
    2. For colleagues in the organization that are responsible for providing public/external communication statements, ensure these internal stakeholders are added to the ticket so they receive regular status updates regarding the incident and can complete their own requirements for communications within and external to the business.
    3. If there are regulations in your jurisdiction requiring reporting of such incidents, ensure the people in your organization responsible for notifying local or federal law enforcement agencies are also notified of the event/added to the ticket. Consult your legal advisor and/or law enforcement for guidance on collecting and preserving the evidence and chain of custody.
    4. There may not be regulations, but either open databases, government agencies or NGOs may track this type of activity. Your reporting may assist others

# Part 2: Contain the Incident

The immediate task will be to disable compromised credentials or revoke permissions associated with those credentials, thereby preventing any further API activity using the compromised credentials.

1. For the compromised credential identified from Part 1, disable those credentials
    1. If they are long term IAM user credentials, disable them using the IAM console or API (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html)
    2. If they are short term credentials obtained via the AWS Security Token Service (AWS STS) they will be associated with an IAM role. There are a couple of options available to disable these:
        1. Revoke all current role sessions (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_revoke-sessions.html) (note that if the threat actor has the ability to obtain new credentials, this won't solve the problem)
        2. If the threat actor is able to obtain another set of credentials and the activity continues, it will then be necessary to remove all IAM policies attached to the role, modify the attached policies to block all access, or modify the role's trust policy to prevent the actor from assuming the role. As the credentials will remain valid for the specified time duration once issued, it is important to note that modifying the trust policy will allow any current valid credentials to continue to be used whilst still valid **Note that the above actions (step 2, substeps 1 and 2) will stop all users from using credentials obtained by assuming the role, including any legitimate users or applications**
2. The compromised credentials should now be disabled. Verify this by checking the AWS CloudTrail console for the next 30 minutes or so for ongoing credential use, whether by access key, IAM user, or Role.

# Part 3: Eradicate the Incident

Now it is time to further investigate what API actions the compromised credentials have performed since the compromise occurred, and then determine if additional resources need to be deleted, terminated, or investigated further. From Part 1, you have identified the compromised credentials, and the likely time of compromise. It will now be necessary to determine what API actions have been performed and what resources have been created, deleted or modified after that time.

1. Using your preferred monitoring tool, access CloudTrail and search for all API actions performed by the compromised credentials, from the date and time of compromise through to the current time:
    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
4. From the resulting list, determine which API calls:
    - Accessed sensitive data (such as S3 GetObject)
    - 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 ….], these include Amazon EC2 Auto Scaling Groups, Amazon EC2 Spot Fleets, Amazon Elastic Kubernetes Service (Amazon EKS)/Amazon Container Service (Amazon ECS) clusters, 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.
    4. Deleted existing AWS resources

    5. Modified existing AWS resources
5. 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
6. 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, 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, go through the following steps from 7 in Part 3 (below) and all relevant steps in Part 4, also below
7. For new resources created during the compromise: Delete all resources created by the compromised credentials that were created post the date/time of the compromise occurring
8. For handling resources that were modified or deleted during the compromise, see Part 4

# 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 add a new one
    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 the deleted resource cannot be restored from backup, consult the CMDB to obtain the resource's configuration, and recreate the resource and configure it into the application's infrastructure

# Part 5: Post-Incident Activity

This activity contains two parts. Firstly, some compromised resources may require forensic analysis, either to fulfil regulatory obligations or improved 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.