Research on an AWS-Based DFIR Framework

November 2025

Kim Jinkook, Jang Wonhee, Kim Seojun, Ahn Hyesong

※ This report was prepared as an one of the Cyber Security Expert Group Project.

Important Notice and Disclaimer

Translation Information

This research report was originally written in Korean and has been translated into English using AI. Please be advised that there may be subtle differences in meaning or nuances compared to the original text. You can access the original Korean version **[here]**

Copyright and Intellectual Property © 2026 Plainbit. All rights reserved.

All content within this report is the exclusive property of **Plainbit**. No part of this publication may be reproduced, redistributed, or transmitted in any form for commercial purposes without the prior written consent of Plainbit. *However, the contents of this report may be utilized for internal business purposes and individual professional tasks***.** Any unauthorized large-scale distribution or modification of this material remains strictly prohibited and may lead to legal action.

Inquiries

If you have any questions or require further clarification regarding the contents of this report, please feel free to contact us at: beomjin.kim@plainbit.co.kr

Table of Contents

1. Overview

The cybersecurity landscape faces new forms of threats alongside the proliferation of cloud technology. While cloud adoption can significantly enhance corporate IT operational efficiency and flexibility, most organizations still maintain on-premises-based incident response and forensic frameworks. This results in limitations that fail to adequately reflect the complex structure and service interconnectivity unique to cloud infrastructure. Consequently, there is a lack of digital forensics and incident response (DFIR) capabilities specialized for cloud environments.

Recently, attacks exploiting cloud-specific resources such as IAM credentials, APIs, and serverless functions have increased, with actual breach cases also being steadily reported. Amidst these changes, the MITRE ATT&CK framework has separated the cloud environment into a distinct matrix, systematically defining the Tactics and Techniques attackers can employ using cloud resources. This signifies that the cloud has evolved beyond being merely a service infrastructure into an independent Attack Surface that attackers can seize and manipulate.

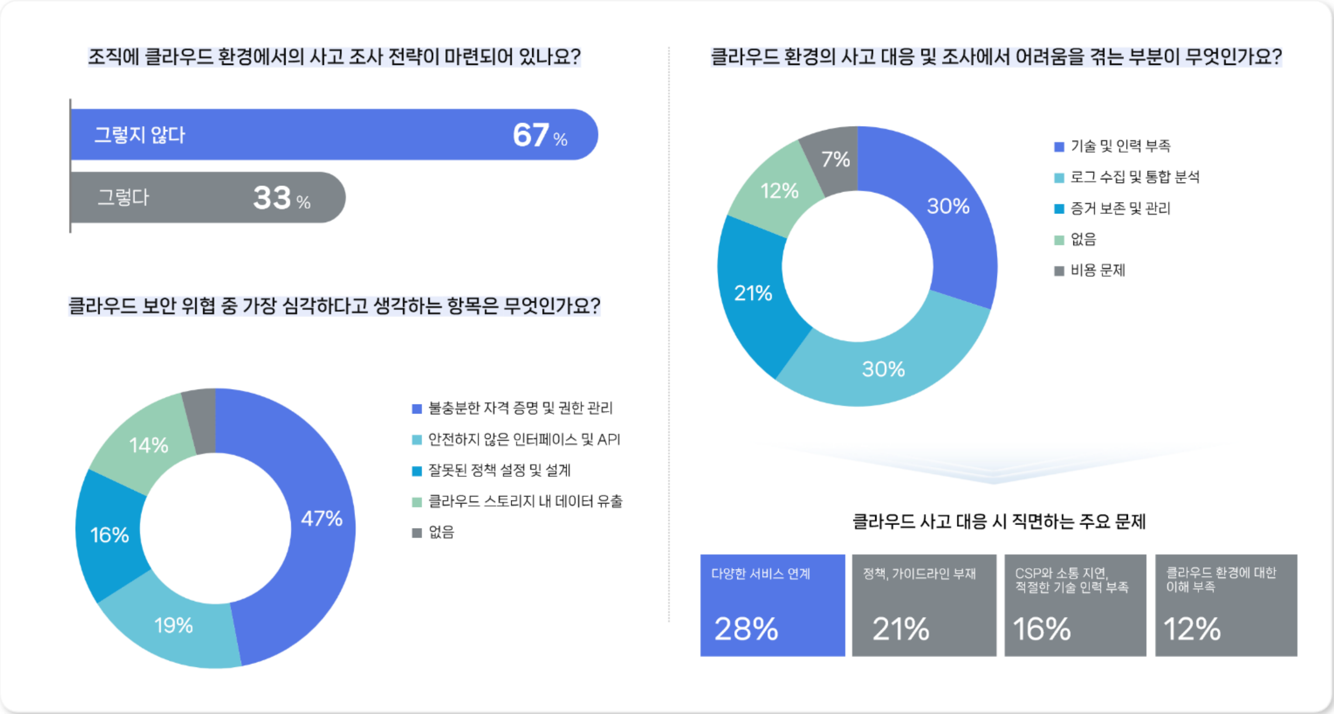
This trend is also clearly evident in global threat intelligence. Mandiant's M-Trends 2025 report classified cloud compromise as an independent section for the first time, placing it as a top-level chapter to emphasize the continuous increase in security incidents within cloud environments.

Given that both the MITRE ATT&CK framework and Mandiant's M-Trends report treat cloud compromise as a distinct threat domain, the importance of establishing incident response systems that reflect the unique structural characteristics and attack surface of the cloud has never been greater.

The domestic situation is no exception. Cloud configuration errors, misunderstandings about security responsibilities, and lack of visibility due to shadow IT resulting from the proliferation of SaaS continue to be pointed out as major security risks. To empirically verify these issues, our research team conducted a survey on the current state of cloud incident response among approximately 100 domestic security and incident response practitioners.

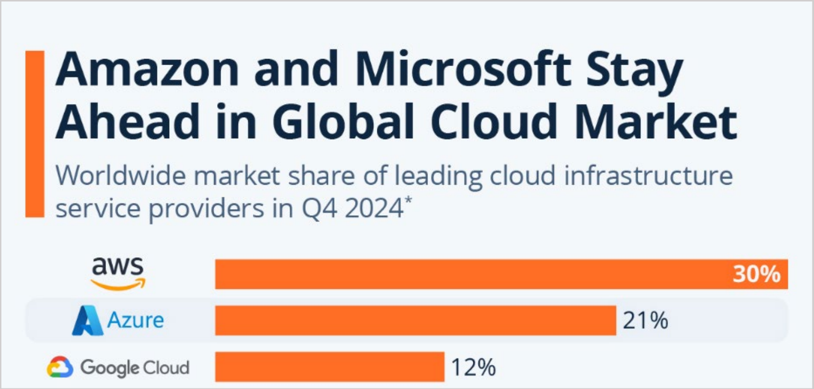
The results revealed that 67% of respondents stated they "do not have a cloud incident investigation strategy in place," while 47% identified "insufficient credential and permission management" as a major threat. Furthermore, difficulties in log collection and integrated analysis (30%) and a shortage of technical personnel (30%) were found to be the most significant response limitations.

These findings clearly demonstrate the need for a standardized DFIR framework that integrates data collection, analysis, and response procedures within cloud environments.



[Figure1] Key Findings from the Cloud Incident Response Survey

This study selected AWS (Amazon Web Services) as the primary empirical analysis target within cloud environments. AWS maintains a consistent leading position in the global cloud market and also showed the highest usage rate as the primary cloud platform in the survey on cloud incident response status conducted by our research team among domestic security practitioners.



[Figure2] Top 3 Global Cloud Market Share (Synergy Research Group)

Therefore, the objective is to operationalize a DFIR framework for effectively collecting and analyzing incidents occurring within the AWS cloud environment. To achieve this, we systematize incident detection and analysis procedures based on correlation analysis of AWS logs (CloudTrail, VPC Flow, S3 Access, etc.). We then validate these procedures through scenario-based empirical analysis, CheatSheet development, and automated tool creation.

By standardizing the proposed procedures through this research, we aim to contribute to strengthening cloud DFIR capabilities and improving accessibility. The specific objectives of the research are as follows.

[Table1] Detailed Research Objectives and Research Methods

Objective Category	Research Method
Deriving Requirements for the Cloud DFIR Framework	Analyze MITRE ATT&CK and AWS Incident Response Playbook to define components necessary for incident response Define data collection and analysis components, identify characteristics of cloud incidents and limitations of existing incident response procedures
Designing an AWS log-based data collection and analysis system	Analyze the structure of key AWS logs such as CloudTrail, VPC Flow Logs, and S3 Access Logs Design log collection and correlation analysis processes suitable for DFIR procedures
Developed an AWS DFIR Cheat Sheet and analysis tools	Standardize key log fields per service and events enabling correlation analysis during incident response Develop a DFIR CheatSheet based on this standardization, and create tools (bitParser for AWS) that support event analysis by attack tactic Performing scenario-based incident analysis verification
Performing scenario-based incident analysis verification	Construct attack scenarios exploiting IAM privilege escalation, misconfigured S3 access permissions, and malicious EC2 activities Validate the detection and analysis effectiveness of the proposed system through log correlation analysis
Evaluate the framework's practical applicability and scalability	Based on scenario validation results, assess the practical applicability and scalability of the proposed framework for automation Evaluate the scalability for practical application and automation of the proposed framework

The research results for each phase are detailed in Chapters 2 through 6 of the report, ultimately presenting the standardization and practical applicability of the AWS-based cloud DFIR framework. The expected benefits are as follows:

First, establishing DFIR procedures optimized for cloud environments to enhance incident response systems

By complementing the limitations of existing on-premises incident response systems and presenting a standardized DFIR framework that reflects the structural characteristics and log generation patterns of cloud environments, organizations can establish and operate systematic and consistent incident response procedures even in cloud environments.

Second, securing visibility through automated log collection and analysis.

To resolve the difficulty of log collection and integrated analysis—a major issue in cloud environments—an automated analysis system based on log flattening and correlation analysis is built. This strengthens the interconnectivity of logs across cloud services and improves the efficiency of incident root cause analysis and anomaly detection.

Third, Improving Practical Accessibility through DFIR CheatSheets and Tooling

To address the shortage of incident response personnel and technical expertise, we provide a DFIR CheatSheet that systematizes service-specific log fields and correlated events, along with analysis support tools. This enables security practitioners to quickly identify key events in cloud incidents, allowing even small and medium-sized organizations to perform efficient cloud DFIR procedures without significant financial burden.

1.1. Overall Summary

This research aimed to strengthen incident response capabilities in AWS environments due to the absence of dedicated cloud incident response guides. By analyzing evidence collection constraints and procedural limitations in cloud environments through prior research, we established a DFIR data collection framework tailored to AWS architecture and defined analysis procedures for specific incident types.

Additionally, we developed an automated analysis tool (bitParser) and a DFIR CheatSheet, validated their effectiveness through ransomware scenarios, and presented a practice-oriented DFIR framework enabling reliable evidence acquisition and rapid behavioral analysis.

[Table2] Summary of research content and results according to the report table of contents

Number	Main Title	Key Content
2	Overview of Cloud Infrastructure and Security	Organizing the theoretical foundation centered on the scope of responsibility and structural constraints in the cloud - Shared Responsibility Model, Security Responsibility Sharing Models by Service Model, AWS Cloud Service Security Architecture, Incident Response Approaches for On-Premises and Cloud Environments, Common constraints in cloud environments
3	Prior Research	Deriving the applicability and limitations of DFIR in AWS cloud environments and identifying improvements to existing response methods - Key cloud security threats, case studies of cloud breach trends, Research on cloud attack tactics and techniques based on MITRE ATT&CK, Investigation of AWS incident response frameworks, AWS security services/logs, AWS Incident Response Playbook investigation
4	Incident Data Collection	Proposing DFIR collection systems and methods specialized for cloud environments - Command-based data collection, log-based data collection, forensic image collection
5	Incident Analysis Techniques	Systematization of DFIR analysis procedures in AWS environments and presentation of incident analysis approaches - Key analysis fields and event analysis per log type, Development of a DFIR CheatSheet mapping log events to attack tactics, Development of AWS DFIR analysis tools
6	Scenario-Based Empirical Analysis	Verification of key log analysis approaches and the effectiveness of the bitParser tool based on a ransomware scenario - Overview of attack scenarios, scenario analysis results

[Table3] Research Deliverables and Outcomes

Category	Content
Cloud Incident Response Status Survey Results	Surveyed approximately 100 domestic security and incident response practitioners on the current state of cloud incident response to establish research direction (seebit.ly/cloud-dfir-survey)
AWS DFIR CheatSheet (CloudTrail, S3 Access Log)	Defined key DFIR events and analysis points based on CloudTrail Log and S3 Access Log (Seebit.ly/dfir-cheatsheet-s3 and bit.ly/dfir-cheatsheet-cloudtrail)
bitParser for AWS Log	Development of an automated tool for integrated parsing and analysis of CloudTrail Log, S3 Access Log, and VPC Flow Logs (See github.com/Plainbit/bitParser)

2. Cloud Infrastructure and Security Overview

Incident response in cloud environments possesses entirely different structural characteristics compared to traditional on-premises environments. These characteristics directly impact security management and incident response procedures. To understand cloud-based incident response frameworks, it is essential to clearly grasp the structural constraints and scope of responsibility within the environment. Therefore, Chapter 2 examines the theoretical foundation of cloud incident response, focusing on the shared responsibility model for security, cloud service security architecture, differences in incident response approaches between on-premises and cloud environments, and common constraints in cloud environments.

The content covered in Chapter 2 is as follows.

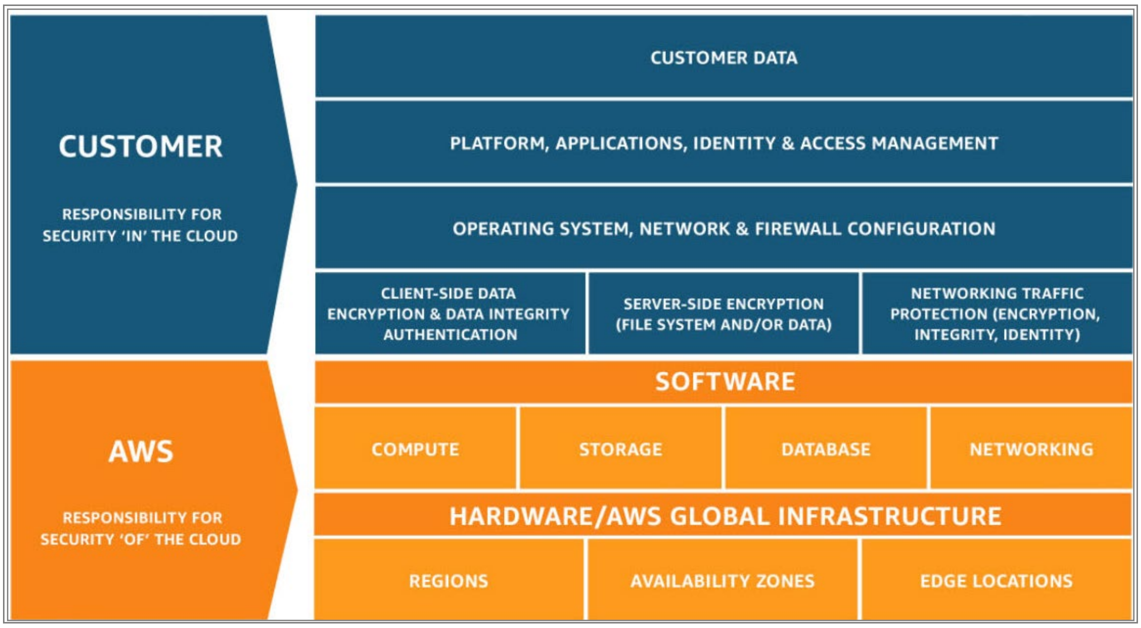
[Table4] Key Research Content – Cloud Infrastructure and Security Overview

Number	Subtitle	Key Content
1	What is the Shared Responsibility Model?	A model that distinguishes cloud security responsibilities between the CSP and the customer, clearly defining each party's scope of protection. Explanation of the Shared Responsibility Model Concept
2	Security Responsibility Sharing Model by Service Model	Classifying cloud services into IaaS, PaaS, and SaaS, and defining customer security responsibilities and Scope of ISMS-P certification audit
3	AWS Cloud Service Security Architecture	Structure a multi-account environment based on the AWS Security Reference Architecture (SRA), and systematically manage security services by integrating them according to roles per organizational unit (OU).
4	Incident Response Approaches for On-Premises and Cloud Environments	Comparing incident response approaches and key differences between on-premises and cloud environments
5	Common constraints in cloud environments	Common Constraints from a Security and Incident Response Perspective

2.1. What is the Shared Responsibility Model?

The Shared Responsibility Model defines how security responsibilities are divided between the Cloud Service Provider (CSP) and the customer in a cloud environment. Its purpose is to prevent security gaps by providing clear guidance on 'Who is responsible for securing what?'. The Cloud Service Provider is responsible for the security of the cloud infrastructure itself, while the customer is responsible for the security of everything operating on that infrastructure.

Therefore, each cloud service provider defines and operates its own Shared Responsibility Model. AWS's Shared Responsibility Model is as follows.


CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

[Figure3] AWS Shared Responsibility Model

2.2. Security Responsibility Sharing Model by Service Model

Cloud service models are broadly categorized into IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). The scope of the customer's security responsibilities varies depending on each model.

[Table5] Security Responsibility Scope by Service Model

Model Classification	CSP Security	Customer Responsibility	Representative Example
IaaS	• Core Infrastructure Protection (Physical data centers, servers, storage, network, etc.) • Includes virtualization layers such as Includes virtualization layers such as hypervisors	• Bears the greatest security responsibility • Directly manages a wide range of areas (operating systems, middleware, data, applications, identity and access management (IAM), network configuration, etc.)	• Amazon EC2 • Microsoft Azure VM • Google Compute Engine
PaaS	• Includes all responsibilities of IaaS • Manages operating systems, middleware, and runtimes • A platform where developers can focus on with a secure platform is the core	• Development and deployment data • Application • User access rights management	• AWS Elastic Beanstalk • Microsoft Azure App Service • Google App Engine
SaaS	• Includes all responsibilities for IaaS and PaaS • Direct management of the application • Customers simply subscribe to and use the software as a service	• The model with the least responsibility • Data within the service • User account and access rights management	• Microsoft 365 • Google Workspace • Salesforce

Furthermore, the scope of assessment for Information Security Management System-Personal Information Protection (ISMS-P) certification also varies depending on the cloud service model.

[Table6] Scope of Services and Assets Subject to ISMS-P Certification Based on Cloud Service Models

Category	Target Services and Assets
IaaS	OS (Guest OS), middleware (WAS, etc.), applications, and DBMS directly managed by the applicant organization
PaaS	Applications directly managed by the applicant organization (However, areas using accounts and permissions assigned by the cloud service provider are included in the authentication scope – e.g., middleware accounts/permissions and passwords)
SaaS	Review performed only on areas related to the application that the applicant organization can manage (Application account/permission management and passwords, etc.)

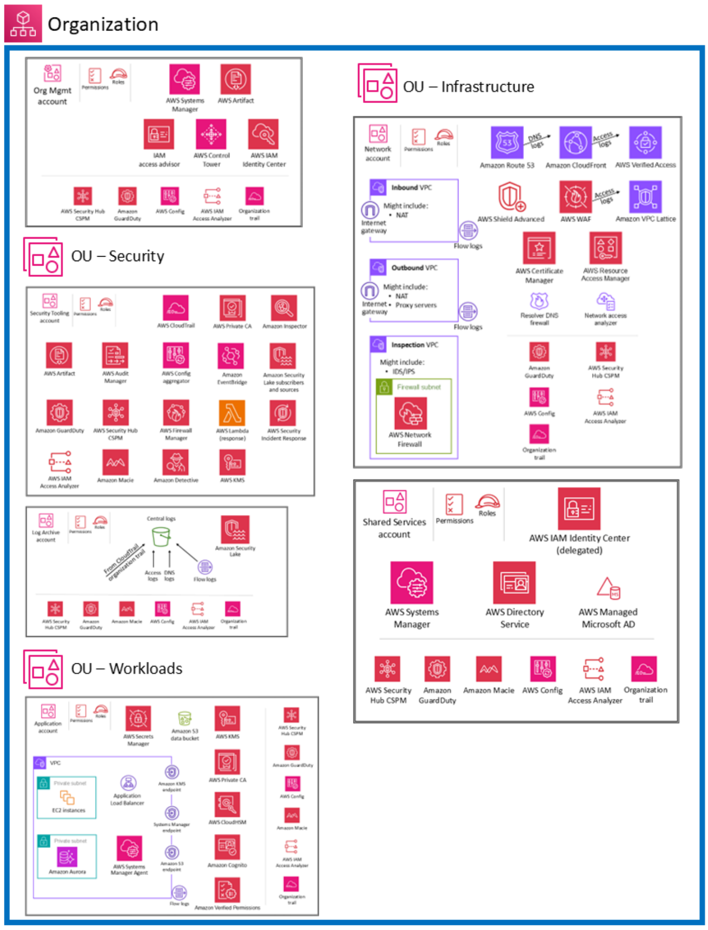
2.3. AWS Cloud Service Security Architecture

CSP provides various security services and architectures to enable customers to securely build and operate their cloud environments.

AWS Security Reference Architecture (SRA) is a guideline demonstrating how AWS security services integrate to deliver comprehensive security capabilities. This architecture centers on systematically managing security in multi-account environments based on AWS Organizations.

AWS separates accounts by purpose through account structuring and Organizational Units (OUs). For example, it applies appropriate security policies based on each account's role and responsibility by structuring accounts into a Security OU (for central management of security services, such as security tool accounts and log archive accounts), an Infrastructure OU (for managing network and common services), and a Workload OU (where actual applications run). The AWS Security Architecture Diagram and key security service principles are as follows.

[Table7] AWS Security Architecture Diagram and Key Security Service Principles

AWS Security Architecture Diagram	Principles of Key AWS Security Services
 Organization Org Mgmt account OU – Security OU – Infrastructure Network account Inbound VPC Outbound VPC Inspection VPC Shared Services account AWS IAM Identity Center (delegated) AWS Systems Manager AWS Directory Service AWS Managed Microsoft AD AWS Security Hub CSPM Amazon GuardDuty Amazon Macie AWS Config AWS IAM Access Analyzer Organization trail OU – Workloads	AWS Control Tower Provides the foundation for securely setting up and managing multi-account environments
	Identity and Access Management (IAM) Using IAM roles, enforce the principle of least privilege Control access to resources
	Virtual Private Cloud (VPC) Logically isolate network traffic and control inbound/outbound traffic via security groups and NACLs to securely configure network routes
	Data Protection Manage encryption keys using AWS KMS (Key Management Service) Manage encryption keys using AWS KMS (Key Management Service) and enable server-side encryption on services like Amazon S3 to protect stored data
	Threat detection and logging Utilize services like Amazon GuardDuty and Amazon CloudWatch to detect threats, record all activity, and centralize monitoring Centralize logs in a separate log archive account to ensure integrity

2.4. Incident Response Approaches for On-Premise and Cloud Environments

Incident response in on-premise and cloud environments shows clear differences in approach and considerations. The differences between the two environments are as follows.

[Table8] DFIR Approaches Between On-Premise and Cloud

Category	On-Premise	Cloud
Infrastructure Characteristics	Static and controlled infrastructure enabling enables easy monitoring and forensic analysis Clearly defined network structure enables relatively easy security policy and breach detection are relatively easy	Dynamic and distributed virtualized environment, Co-managed with CSP, making physical asset access impossible Complex structures like microservices, containers, and serverless make incident response procedures complex
Shared Responsibility Model	Security responsibility for all infrastructure and data rests entirely with the customer, who retains full control to investigate and respond across all domains in the event of an incident	The CSP manages the infrastructure, while the customer is responsible for application, data, and access management. Close collaboration between both parties is essential for incident response
Visibility and Monitoring	Static infrastructure enables visibility, traffic analysis, and anomaly detection. Physical access enables enables forensic analysis of disks, memory, etc.	However, the ephemeral nature and complex structure of resources require real-time log collection and analysis for visibility requires real-time log collection and analysis, but but deep visibility is limited due to reliance on CSP tools
Data Accessibility and Collection	Customers build and operate infrastructure directly in their own data centers, making physical evidence collection difficult during incidents possible	Virtualized environment allows customers only logical access Evidence collection relies on CSP's snapshot and log APIs, Scope is limited
Tools and automation	Traditional security tools tailored for static infrastructure (firewalls, IDS/IPS, EDR, SIEM, etc.) The tool provides real-time monitoring and threat detection and response capabilities within the data center.	There are limitations to utilizing existing on-premises tools, and cloud-native security tools are required. It is crucial to enhance the efficiency of detection and response for large-scale, high-speed cloud operations through automation. is crucial.
Attack surface and threats	The attack surface is limited and centered on physical infrastructure, with networks, endpoints, and internal applications as primary targets Asset damage via malicious files, phishing, and ransomware, and information leakage pose major threats	The attack surface expands due to the responsibility for securing data and services across multiple environments Cloud-specific attack techniques such as misconfigurations, vulnerable APIs, and credential theft Cloud-specific attack techniques pose major threats
Response and Recovery	Clearly defined response and recovery procedures are in place Enables device isolation, backup restoration, and direct patch application	Playbooks and scripts enable rapid isolation and recovery, Complexity exists in multi-region coordination Leveraging redundancy and scalability can significantly improve recovery speed
Technology and Expertise	On-premises response experts specialize in physical infrastructure-centric technologies such as networking, endpoint security, and storage. security, and other physical infrastructure-centric technologies Their capabilities are primarily focused on leveraging existing security tools	Cloud architecture, CSP security tools, automation technologies, etc. Broad expertise is required To respond to rapidly changing environments Continuous learning and staying abreast of the latest threat trends are essential

2.5. Common constraints in cloud environments

While cloud environments offer many benefits, common constraints exist from a security and incident response perspective, as follows.

[Table9] Common Constraints in Cloud Environments

Category	Target Services and Assets
Log Retention Period	Log services like CloudTrail have limited default retention periods, Configure to export logs to separate storage like S3 or Blob for permanent retention to ensure compliance or support long-term analysis Otherwise, there is a high risk of losing evidence during incident investigations.
Limited Access to Evidence	Customers can only access their own virtual resources (such as VMs and storage) and cannot access the physical infrastructure or hypervisor. Therefore, direct investigation is difficult when a breach occurs on CSP infrastructure, necessitating reliance on the CSP's investigation results a Therefore, direct investigation is difficult when a breach occurs involving CSP infrastructure, necessitating reliance on the CSP's investigation results and provided information.
Data Rights and Location	You can choose the physical location (Region) where data is stored, government requests for data access may occur based on country-specific laws or regulations
Multi-tenancy environment	Clouds operate on a multi-tenant architecture (where multiple customers share physical resources). While CSPs provide logical isolation, However, vulnerabilities in the hypervisor or isolation failures could potentially allow resources from one customer to impact another.

3. Prior Research

Chapter 3 analyzes prior research to establish an incident response framework for cloud environments. It analyzes the characteristics of cloud-based threats and the limitations of existing responses through trends and representative cases of cloud incidents. It also conducts an in-depth analysis of the MITRE ATT&CK framework and the AWS Incident Response Playbook to derive the necessary data collection and analysis components for incident response and directions for improving procedures. These research results serve as the basis for the cloud DFIR framework design and implementation plan presented in Chapters 4 and 5.

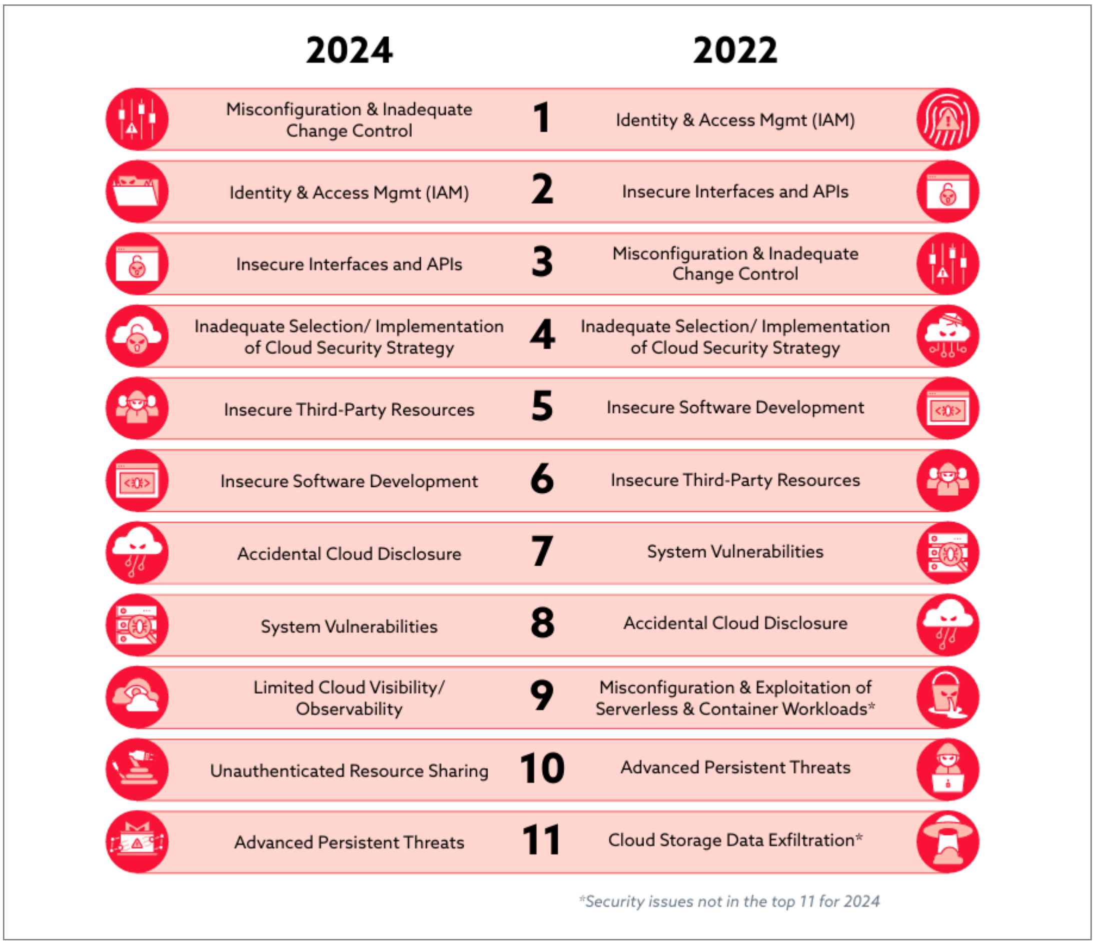
The content covered in Chapter 3 is as follows.

[Table10] Key Research Content – Prior Research

Number	Subtitle	Key Content
1	Key Cloud Security Threats	The Cloud Security Alliance (CSA) has identified the following 11 major security threats in cloud environments as of 2024: 2. Data Breaches
2	Cloud Incident Trends and Case Studies	Investigation of cloud security threats such as misconfigurations and credential theft, along with major incident cases
3	Research on Cloud Attack Tactics and Techniques Based on MITRE ATT&CK Research on Cloud Attack Tactics and Techniques	Summary of Key Attack Techniques by Tactics in MITRE ATT&CK v17.1 (10 Tactics, 85 Techniques Total)
4	AWS Incident Response Framework Investigation	Investigation of the 5-step incident response framework proposed by AWS based on NIST SP 800-61 (Organization of Security Services and Logs Available for Each Phase)
5	AWS Security Service Investigation	Investigation of Utilization Methods for 12 Key AWS Security Services Focused on Log-Based Visibility
6	AWS Log Investigation	Investigation of AWS-provided log types by layer and incident response utilization strategies (Account and Management Activity Logs, Network and Traffic Logs, Service-Specific Access/Activity Logs, Security Service Logs, System and Application Logs)
7	AWS Incident Response Playbook Investigation	Analysis of 16 Incident Types Useful from a DFIR Perspective in AWS Incident Response Playbooks (DFIR perspective analysis points, key logs and data, incident response procedure summary)

3.1. Key Cloud Security Threats

The global security organization 'Cloud Security Alliance (CSA)' publishes an annual cloud threat report. It surveys over 500 industry experts on security issues in the cloud industry and identifies 11 major security problems occurring in cloud environments. The major cloud security threats announced as of 2024 are as follows.



2024		2022
Misconfiguration & Inadequate Change Control	1	Identity & Access Mgmt (IAM)
Identity & Access Mgmt (IAM)	2	Insecure Interfaces and APIs
Insecure Interfaces and APIs	3	Misconfiguration & Inadequate Change Control
Inadequate Selection/ Implementation of Cloud Security Strategy	4	Inadequate Selection/ Implementation of Cloud Security Strategy
Insecure Third-Party Resources	5	Insecure Software Development
Insecure Software Development	6	Insecure Third-Party Resources
Accidental Cloud Disclosure	7	System Vulnerabilities
System Vulnerabilities	8	Accidental Cloud Disclosure
Limited Cloud Visibility/ Observability	9	Misconfiguration & Exploitation of Serverless & Container Workloads*
Unauthenticated Resource Sharing	10	Advanced Persistent Threats
Advanced Persistent Threats	11	Cloud Storage Data Exfiltration*

*Security issues not in the top 11 for 2024

[Figure4] Key Cloud Security Threats - Comparison of 2024 and 2022

Detailed information on each threat is as follows.

1) Misconfiguration & Inadequate Change Control

Misconfiguration refers to configuration errors that make cloud assets vulnerable to unintended damage or attacks. These errors can stem from a lack of understanding of security settings or malicious actions. Key examples include failed password management, disabled logging, excessive access permissions, insufficient validation, subdomain hijacking, and CSP-specific configuration errors (e.g., S3 buckets).

2) Identity & Access Management (IAM)

Identity & Access Management (IAM) is a core security framework that verifies user identities and manages roles, permissions, and access conditions to allow only authorized resource access. Key components include authentication, authorization, SSO, MFA, and activity monitoring. Misconfiguration or inadequate management can create vulnerabilities for unauthorized access.

3) Insecure Interfaces & APIs

In cloud environments, APIs and UIs provided by CSPs, customers, and developers are key control points. They can become vulnerable due to inadequate authentication, insufficient encryption, poor session management, lack of input validation, inadequate logging and monitoring, insufficient patching, excessive access permissions, and lack of rate limiting. These vulnerabilities can lead to unauthorized access, sensitive data leaks, and service disruptions.

4) Inadequate Cloud Security Strategy

A cloud security strategy is the process of establishing principles for cloud architecture, service models, CSP selection, service regions, and billing models, considering external factors, existing implementation status, technology choices, and priorities. It contributes to achieving the organization's security goals and ensuring business continuity. This strategy ensures secure operation across services and supports risk response and decision-making. An inadequate cloud security strategy can lead to actual incidents.

5) Insecure Third-Party Resources

As cloud adoption surges, security risks stemming from Third-Party Resources (external code, open-source, SaaS, etc.) are escalating. These are considered supply chain vulnerabilities and are a primary focus of Cybersecurity Supply Chain Risk Management (C-CSRM).

6) Insecure Software Development

The complexity of cloud technology creates unintended vulnerabilities, and vulnerable software becomes an attack vector. Therefore, in CI/CD and automated environments, understanding the shared responsibility model, applying the SDLC, implementing the principle of least privilege, and providing continuous education for developers are essential.

7) Accidental Data Disclosure

The risk of data leaks due to misconfigured cloud services increases annually. Public search tools can easily locate exposed storage (e.g., S3, Azure Blob, GCP Storage, Docker Hub, Elasticsearch, Redis, GitHub, etc.). These leaks primarily stem from negligence and inadequate access controls (such as accidental public settings).

8) System Vulnerabilities

Defects in cloud services (system vulnerabilities) can compromise confidentiality, integrity, and availability, disrupting service operations. Vulnerability types are primarily categorized as misconfiguration, zero-day (unknown/0-day), unpatched software, and weak or default credentials.

9) Limited Cloud Visibility/Observability

Limited cloud visibility makes it difficult to distinguish between legitimate and malicious service usage, encompassing Shadow IT (unauthorized application use) and misuse of approved applications. Insufficient cloud visibility leads to security blind spots, failure to detect breaches, and permission management issues by failing to properly detect insider or attacker activity.

10) Unauthenticated Resource Sharing

Unauthenticated resource sharing in the cloud exposes sensitive assets like virtual machines, storage, and databases to unauthorized access risks. Default passwords remain commonly unconfigured, making such resources easily discoverable via public search tools like Shodan.

11) Advanced Persistent Threats (APTs)

APTs (Advanced Persistent Threats) remain a major threat to cloud security. Nation-state hackers and organized crime groups target sensitive data within the cloud through long-term, sophisticated attacks.

3.2. Cloud Incident Trends and Cases

Cloud security threats have evolved from simple configuration errors to credential theft, supply chain breaches, and API abuse, with attackers exploiting the scalability and accessibility of cloud environments. Recent incident cases occurring in domestic and international cloud environments include the following.

1) [APT41] Information Leakage Incident (May 2023)

The Chinese APT group APT41 exploited the 'Follina' zero-day vulnerability in Microsoft software to gain unauthorized access to the cloud systems of multiple government agencies and potentially extract sensitive information.

2) [Toyota] Vehicle Data Leak Incident (May 2023)

Toyota Motor Corporation experienced an incident where user data for approximately 2.15 million users in Japan was exposed publicly for 10 years. The cause was identified as a cloud configuration error. Affected users were those of the T-Connect and G-Link services. The exposed information included vehicle location and identification numbers, but no cases of misuse were reported.

3) [JumpCloud] Personal Information Leak Incident (June 2023)

Identity and access management company JumpCloud suffered a data breach by sophisticated nation-state attackers. The cause was traced to a spear-phishing campaign and unexpired credentials. For the breach, attackers injected malicious data into JumpCloud's command framework, targeting specific customer accounts.

4) [DarkBeam] Personal Information Leak Incident (September 2023)

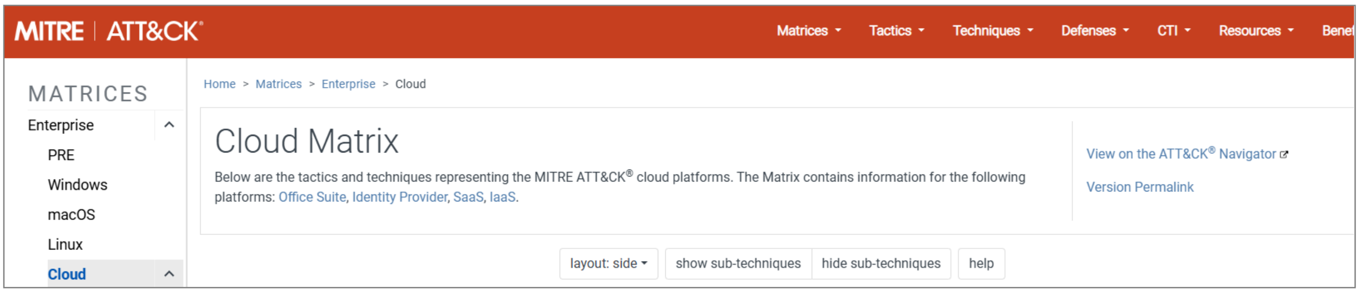
Cloud security firm DarkBeam suffered a breach where over 3.8 billion email and password records were exposed due to unprotected Elasticsearch and Kibana interfaces. The cause was identified as an administrator's configuration error: passwords were not set after maintenance. The exposed data included 16 collections such as "email 0–9" and "email A–F".

5) [Mercedes Benz] Data Breach Incident (January 2024)

A Mercedes Benz API leak incident allowed attackers to access the company's GitHub Enterprise, resulting in the exposure of source code, cloud keys, and internal documents. The breach was traced to an employee's GitHub token discovered in a public repository the previous year, which was exploited as an entry point.

3.3. Research on Cloud Attack Tactics and Techniques Based on MITRE ATT&CK

Based on MITRE ATT&CK v17.1, the Cloud Matrix comprises a total of 10 tactics and 85 techniques. The attack techniques for each tactic are as follows.



Cloud Matrix

Below are the tactics and techniques representing the MITRE ATT&CK® cloud platforms. The Matrix contains information for the following platforms: Office Suite, Identity Provider, SaaS, IaaS.

View on the ATT&CK® Navigator

Version Permalink

[Figure5] MITRE ATT&CK Framework - Cloud Matrix

1) Initial Access

Attackers exploit vulnerabilities in assets exposed to the internet or security configuration errors, or use techniques like phishing and credential theft to infiltrate cloud environments. With the proliferation of cloud services, the number of applications potentially exposed externally has increased, and the number of user and service accounts requiring management has grown. Consequently, these infiltration paths and the risk of account compromise have escalated, heightening the likelihood of incidents such as data breaches and privilege abuse.

[Table11] List of techniques used in the Initial Access tactics of the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1180	Drive-by Compromise	• An attacker can gain access to a system through a user visiting a website during normal browsing. to gain access to the system • by modifying a normal script file provided to the website Modifying legitimate script files served to the website
T1190	Exploit Public-Facing Application	• Initial penetration attempt exploiting vulnerabilities in hosts exposed to the internet • When an application is hosted on cloud-based infrastructure or containerized, exploiting this may compromise the underlying container compromised
T1566.002	Phishing: Spearphishing Link	• Exploiting spearphishing emails containing malicious links
T1199	Trusted Relationship	• Granting high-level access privileges to external vendors and managing not only internal systems but also cloud environments, Accounts assigned to external vendors may be compromised
T1078.004	Valid Accounts: Cloud Accounts	• In a cloud environment, initial access can be achieved using valid accounts. to gain initial access • Attackers can gain access to accounts through brute-force attacks, phishing, or other various means gain access to accounts

2) Execution

Attackers who successfully gain access to the cloud environment execute malicious code to achieve their objectives. Unlike on-premises environments, cloud environments allow direct command execution via APIs. Attackers exploit centralized management tools like AWS Systems Manager or Microsoft Intune to remotely deploy and execute code. These centralized management tools possess high privileges enabling extensive control over hosts within the network. If administrative privileges are compromised, there is a risk of rapid and widespread compromise across numerous connected systems.

[Table12] List of techniques used in the Execution tactics of the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1651	Cloud Administration Command	• Abuse cloud management services to execute commands within virtual machines AWS Systems Manager, Azure RunCommand, • Using resources such as AWS Systems Manager, Azure RunCommand, Runbooks, it is possible to execute commands within virtual machines agents to execute scripts remotely on virtual machines
T1059.009	Command and Scripting Interpreter: Cloud API	• Malicious commands can be executed by exploiting the cloud API • Using Cloud API functionality enables control over computing, storage, identity and Access Management (IAM), networking, security policies, and other tenant all major services
T1648	Serverless Execution	• CSP provides serverless resources that enable applications to be built without managing servers. Attackers can exploit these resources to execute arbitrary commands to execute arbitrary commands • Malicious code can be executed by exploiting Lambda, a serverless function
T1072	Software Deployment Tools	• Attackers can use centralized configuration management and software deployment tools to to execute commands on other systems within the network • This service supports cloud management commands and enables arbitrary command execution on on-premises hosts
T1204.003	User Execution: Malicious Image	• Attackers can upload images containing malicious code (backdoors, cryptocurrency mining, etc.) to public repositories (e.g., GitHub), Users can download the malicious image and deploy it in a cloud environment • Not only AWS AMIs and images, but widely used container runtimes like Docker container runtimes like Docker. • Instances deployed via malicious images are already infected with malware allowing attackers to access the system without an initial penetration process.

3) Persistence

The goal is to maintain access to the environment even after system reboots, credential changes, or other disruptive actions. While persistence in on-premises environments primarily relies on leaving malicious files in the file system or registry, persistence in cloud environments relies on manipulating state and configuration. The core of cloud persistence is manipulating IAM objects. Both Account Manipulation and Create Account: Cloud Account techniques directly attack the cloud IAM structure to secure persistent access paths. These techniques are difficult to detect because they are hard to distinguish from legitimate cloud management activities.

[Table13] List of techniques used in the Persistence tactics of the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1098.001	Account Manipulation: Additional Cloud Credentials	• To gain persistent access to the victim's cloud accounts and instances, Adding attacker-controlled credentials to the cloud account • In Azure/Entra ID environments, attackers can add attackers-controlled credentials to cloud accounts add credentials for service principals and applications (which may take the form of x.509 certificate keys and passwords)) • If they have the appropriate permissions, these credentials can be used to access resources through the Azure Portal, Azure CLI, or the Azure/Azure PowerShell module. in various ways. • In an IaaS (Infrastructure as a Service) environment, after gaining access through a cloud account, and then generate or import their own SSH key. and add access keys to the account using AWS APIs or GCP commands.
T1098.003	Account Manipulation: Additional Cloud Roles	• Attackers can add roles or permissions to cloud accounts they control to maintain persistent access. • Examples exist where IAM was manipulated to gain persistence and elevate privileges, or where a global administrator role was added to an account created on a cloud instance within the target organization
T1098.004	Account Manipulation: SSH Authorized Keys	• Attackers can modify the SSH file (authorized_keys) on the victim host to maintain persistence. Modify the SSH file (authorized_keys) • In cloud environments, attackers can use the command-line interface or REST API to modify the SSH authorized_keys file file of a specific virtual machine
T1098.005	Account Manipulation: Device Registration	• An attacker can register a device with the multi-factor authentication (MFA) system of an account they manage. T1098.005
T1136.003	Create Account: Cloud Account	• Attackers can create new cloud user accounts or service account (Azure service account, GCP service account, AWS IAM user) within the victim's environment
T1546	Event Triggered Execution	• Attackers can configure their malicious code to execute automatically when specific events occur, thereby achieving persistence • Pacu malware can trigger a malicious Lambda function when a CloudFormation template is uploaded to a bucket to trigger a malicious Lambda function

TID	Technique Category	Technique Description
T1525	Implant Internal Image	• After gaining access to the environment, attackers can implant malicious code into cloud or container images to achieve persistence. by implanting malware into cloud or container images • AWS AMI, Google Cloud Platform (GCP) images, Azure images, as well as widely used container runtimes like Docker container runtimes like Docker
T1556.006	Modify Authentication Process: Multi-Factor Authentication	• Attackers can disable or modify multi-factor authentication (MFA) systems to gain persistent access to compromised accounts • by excluding accounts from Azure AD Conditional Access policies or or register new MFA methods controlled by the attacker. to bypass it
T1556.007	Modify Authentication Process: Hybrid Identity	• Attackers can apply patches, modify, or otherwise install backdoors into the cloud authentication process linked to on-premises user IDs to bypass standard authentication mechanisms, gain access to credentials, and enable persistent access to accounts. permanent access to accounts. • Attackers can run malicious DLLs on on-premises servers running PTA agents in the Entra ID authentication process • In environments using AD FS, they can modify the configuration file to load the malicious DLL, by modifying configuration files to bypass AD FS policies
T1556.009	Modify Authentication Process: Conditional Access Policies	• Attackers can disable or modify conditional access policies to allow persistent access to compromised accounts

4) Privilege Escalation

This is the stage where an attacker gains higher-level control over systems or data beyond the limited privileges obtained through the initial breach. In cloud environments, privilege escalation primarily involves techniques that exploit configuration errors or intentional features within complex, granular cloud IAM systems.

[Table14] List of techniques used in the Privilege Escalation tactics within the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1548.005	Abuse Elevation Control Mechanism: Temporary Elevated Cloud Access	• An attacker can exploit a configuration that allows temporary elevated access to cloud resources by exploiting poorly configured permissions • Just-in-time access is a mechanism that assigns additional roles to cloud accounts additional roles in a granular and temporary manner Accounts operate daily with only the necessary permissions and request additional permissions as needed
T1098.001	Account Manipulation: Additional Cloud Credentials	• Attackers can use AWS APIs or GCP commands to add access keys to accounts and use the AWS API to add passwords to add to the request account. • If the target account's permissions differ from the requesting account's, Attackers can also escalate privileges within the cloud environment • In an Entra ID environment, an attacker with an application administrator role can add new credentials to the application's service principal Add new credentials
T1098.003	Account Manipulation: Additional Cloud Roles	• Attackers can add roles to compromised existing accounts to gain elevated privileges • In AWS environments, attackers can use the CreatePolicyVersion API to define a new version of an IAM policy, or use the or use the AttachUserPolicy API to attach an IAM policy to a compromised user account with additional or unique permissions
T1484.002	Domain or Tenant Policy Modification: Trust Modification	• An attacker can modify the attributes of a new domain or an existing domain trust or alter trust relationship configurations to achieve privilege escalation • and share trust details such as whether a user ID is federated to access shared resources. Applying authentication and authorization attributes between domains or tenants • By manipulating these trusts, attackers can modify settings to add objects they control to elevate privileges
T1078.004	Valid Accounts: Cloud Accounts	• Cloud accounts can be used to gain temporary elevated or other privileges • In Azure environments, attackers can exploit Azure managed identities to request tokens that access connected Azure resources to request tokens

5) Defense Evasion

Comprises techniques attackers use to evade detection throughout the penetration process. In cloud environments, beyond traditional malware concealment techniques, attackers can manipulate the configuration and functionality of the cloud infrastructure itself to neutralize defense systems. A prime example is attackers deleting event logs stored within the system; in cloud environments, they can disable or delete cloud logging services like AWS CloudTrail, Azure Monitor, or Google Cloud Audit Logs.

[Table15] List of techniques used in the Defense Evasion tactics of the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1548.005	Abuse Elevation Control Mechanism: Temporary Elevated Cloud Access	• An attacker can exploit a permission setting that allows temporary elevated access to cloud resources. by exploiting permission settings that allow such access. • In many cloud environments, administrators use 'Just-in-Time access permission requests' for user or service accounts. 'Impersonation of another account', 'Delegation of roles to resources and services', 'Obtaining short-term high-privilege access', and similar permission-granting capabilities • These capabilities must be assigned to specific roles to be used, but yet configuration errors by cloud administrators can inadvertently create elevated access paths to unintended resources.
T1484.002	Domain or Tenant Policy Modification: Trust Modification	• An attacker can add a new domain trust or modify the attributes of an existing trust, alter the trust relationship configuration between domains/tenants, to bypass defenses or elevate privileges. • By manipulating trust relationships, attackers can add objects under their control or modify settings to achieve privilege escalation or bypass defenses.
T1672	Email Spoofing	• Attackers can manipulate email header values to spoof the sender's identity and spoof the sender's identity. This can include not only the email body but also the From header containing the sender's email address, can be manipulated.
T1211	Exploitation for Defense Evasion	• Attackers can exploit vulnerabilities in systems or applications to bypass security features • Through vulnerabilities in exposed infrastructure such as SaaS applications, enabling the establishment of concealed infrastructure and evasion of security log detection
T1564.008	Hide Artifacts: Email Hiding Rules	• Attackers can hide incoming emails in compromised users' mailboxes by exploiting email rules • Attackers can set email rules within the mailbox of a compromised account to delete responses to security alerts, C2 communications, and internal spear-phishing emails or move responses to these emails to inconspicuous folders. They can also configure rules to automatically modify or delete all emails related to security incident notifications.

TID	Technique Category	Technique Description
T1562.001	Impair Defenses: Disable or Modify Tools	• Attackers can modify or disable security tools to avoid detection of malware, tools, or activities. to evade detection of their malware, tools, or activities • In cloud environments, attackers may modify or disable security tools to evade detection of malware, tools, or activities. Google Cloud Monitor to disable log collection and notification functions
T1562.007	Impair Defenses: Disable or Modify Cloud Firewall	• Disable or modify firewalls within the cloud environment to bypass security controls restricting access to cloud resources • If an attacker gains appropriate permissions, they can add a new ingress rule to the default security group Add New Ingress Rule to Default Security Group', 'Create New Security Group to run scripts or tools that allow TCP/IP access', or 'configure policies to allow malicious traffic such as cryptocurrency mining'. Firewall settings can be modified. • Or by modifying or disabling the cloud firewall 'Enable C2 communication', 'Internal movement from the cloud control plane to the data plane', internal movement from the Control Plane to the Data Plane', brute-force attacks and exposing resources for endpoint denial-of-service attacks.
T1562.008	Impair Defenses: Disable or Modify Cloud Logs	• Attackers may disable or modify logging capabilities and integration settings in cloud environments disable or manipulate logging capabilities and integration settings in the cloud environment manipulate them.
T1656	Impersonation	• Attackers can impersonate trusted individuals or organizations to deceive and persuade targets to perform specific actions on their behalf
T1070.008	Indicator Removal: Clear Mailbox Data	• Ability to manipulate mail data to erase traces of attack activity
T1556.006	Modify Authentication Process: Multi-Factor Authentication	• Attackers may compromise accounts without MFA or use bypass techniques like creating fake MFA requests to gain network access, then modify or disable the MFA defenses.
T1556.007	Modify Authentication Process: Hybrid Identity	• Attackers can patch or modify cloud authentication processes linked to on-premises user IDs or plant a backdoor to bypass standard authentication mechanisms, steal credentials, and gain persistent access to accounts
T1556.009	Modify Authentication Process: Conditional Access Policies	• Can disable or modify conditional access policies to gain persistent access to compromised accounts
T1578.001	Modify Cloud Compute Infrastructure: Create Snapshot	• Attackers can create snapshots or data backups within cloud accounts to bypass defense systems • After creating a cloud instance, attackers mount the created snapshot to the instance and apply policies granting attackers access, such as firewall rules permitting SSH connections • Unlike the Revert Cloud Instance technique, this approach bypasses defenses by creating a separate instance • The Pacu tool can be used to create snapshots of EBS volumes and RDS instances

TID	Technique Category	Technique Description
T1578.002	Modify Cloud Compute Infrastructure: Create Cloud Instance	• Attackers can bypass defenses by creating new instances or virtual machine (VM) within the cloud account to bypass defense systems • Creating a new instance may bypass firewall rules or permission controls applied to existing instances or permission controls applied to existing instances • Creating new instances does not affect currently running instances and enables them to perform malicious activities covertly within the same cloud environment perform malicious activities within the same cloud environment
T1578.003	Modify Cloud Compute Infrastructure: Delete Cloud Instance	• After performing malicious activities, attackers can delete cloud instances to conceal their traces and and delete the cloud instance to evade detection
T1578.004	Modify Cloud Compute Infrastructure: Revert Cloud Instance	• After performing malicious actions, attackers may evade detection and erase their traces by reverting changes to cloud instances to evade detection and erase their traces. • Another technique involves utilizing temporary storage (Temporary Storage)
T1578.005	Modify Cloud Compute Infrastructure: Modify Cloud Compute Configurations	• Attackers modify settings that directly affect the size, deployment location, and available to bypass defense systems. • Even if attackers gain control of the cloud environment, they can request quota adjustments to achieve their objectives (e.g., resource hijacking) without depleting the victim's entire quota, enabling them to perform malicious operations covertly. • Additionally, you can increase the allowed resource usage by modifying tenant-level policies such as virtual machine (VM) size limits, and activate unsupported or unused cloud regions to enable resource deployment in specific regions.
T1666	Modify Cloud Resource Hierarchy	• Attackers may attempt to modify the hierarchy of the IaaS (Infrastructure as a Service) environment to evade defense systems
T1535	Unused/Unsupported Cloud Regions	• Attackers may create cloud instances in unused geographic service regions to evade detection • They exploit the fact that users typically utilize only some of the available regions and may not actively monitor the remaining regions to create resources in unused regions • A similar variant involves exploiting differences in security features between cloud regions and select regions that do not provide advanced detection capabilities to conceal their attack activities

TID	Technique Category	Technique Description
T1550.001	Use Alternate Authentication Material: Application Access Token	• Attackers use stolen application access tokens to to bypass standard authentication procedures and access restricted accounts, information, or services on remote systems. • These tokens are typically stolen from users or services and and can be used without login credentials. • Stolen access tokens can be leveraged in the initial stages as an initial step for penetration into other services • Direct access via APIs can bypass MFA (multi-factor authentication) and is difficult to defend against even with difficult to defend against even with intuitive countermeasures like password changes
T1550.004	Use Alternate Authentication Material: Web Session Cookie	• Attackers can use stolen session cookies to authenticate to web applications and services. Since they reuse an already authenticated session, by reusing an already authenticated session, potentially bypassing some multi-factor authentication (MFA) protocols.
T1078.001	Valid Accounts: Default Accounts	• Attackers can obtain and exploit credentials for default accounts to gain initial access, achieve persistence, elevate privileges, or evade defenses. and exploit them • Default accounts also include AWS root user accounts, ESXi root user accounts, Kubernetes default service accounts and other systems, software, or equipment. by the provider.
T1078.004	Valid Accounts: Cloud Accounts	• Valid accounts in cloud environments are those that attackers can use to gain initial access, persistence, privilege escalation, or defense evasion. • Service accounts or user accounts can be targeted by attackers to gain access to the environment. as targets to gain access to the environment. • Attackers can maintain persistence within the environment and bypass security controls such as multi-factor authentication (MFA) by generating additional cloud credentials for compromised cloud accounts that can be used over extended periods. .

6) Credential Access

Attackers aim to gain elevated privileges within systems and networks, stealing credentials like account information, tokens, and keys for lateral movement. Credential acquisition in cloud environments combines methods used in traditional on-premises environments with cloud-specific approaches. Brute-force attacks target externally accessible cloud services and may also directly attack centralized credential management systems like AWS Secrets Manager, Azure Key Vault, and GCP Secret Manager.

[Table16] List of techniques used in the Credential Access tactics of the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1110.001	Brute Force: Password Guessing	• An attacker with no prior knowledge of legitimate credentials within a system or environment attempts to access accounts by guessing passwords • Attacks can be performed not only on commonly targeted services (SSH, Telnet, FTP, etc.) but also on cloud-based applications and external email T1110.002
T1110.002	Brute Force: Password Cracking	• Once an attacker obtains credential data such as password hashes, attempt password cracking to recover usable credentials • by guessing passwords used in the hash calculation or or use precomputed rainbow tables to decrypt the hash.
T1110.003	Brute Force: Password Spraying	• Attempting to obtain valid account credentials by a single commonly used password or a small list of passwords T1110.004
T1110.004	Brute Force: Credential Stuffing	• Attackers may use credentials obtained from accounts unrelated to the target environment to access the target account • Exploiting the tendency for users to reuse the same password to hijack accounts
T1555.006	Credentials from Password Stores: Cloud Secrets Management Stores	• Attackers can exploit vulnerabilities in AWS Secrets Manager, GCP Secret Manager, Azure Key Vault, Terraform Vault, and other cloud-based secret management solutions. • If attackers gain sufficient privileges within the cloud environment, they can request credentials using commands like AWS's `get-secret-value`, GCP's `gcloud secrets describe`, or Azure's `az key vault secret show`.
T1212	Exploitation for Credential Access	• Attackers can exploit software vulnerabilities to collect credentials to collect credentials. • Credential and authentication mechanisms are processes that allow attackers to access useful credentials or bypass the process of gaining authenticated access to a system. as a means to bypass the process. Attackers can exploit vulnerabilities in public cloud vulnerabilities in public cloud infrastructure to generate and renew unintended authentication tokens.
T1606.001	Forge Web Credentials: Web Cookies	• Attackers can forge web cookies that can be used to access web applications or internet services. to access web applications or internet services • Web applications and services (hosted in cloud SaaS environments or on-premises servers) often use session cookies to to authenticate and authorize user access • Attackers can use web cookies to bypass multi-factor authentication and other authentication protection mechanisms

TID	Technique Category	Technique Description
T1606.002	Forge Web Credentials: SAML Tokens	- If an attacker possesses a valid SAML token signing certificate, they can forge SAML tokens using any permission claims and validity period - Using forged SAML tokens, attackers can authenticate to services using SAML 2.0 in services using SAML 2.0 as a single sign-on (SSO) mechanism - If a SAML token representing a high-privilege account is forged, the attacker could gain Entra ID administrative privileges
T1556.007	Modify Authentication Process: Hybrid Identity	- An attacker could patch, modify, or otherwise install a backdoor in the cloud authentication process linked to on-premises user IDs to bypass standard authentication mechanisms and gain access to credentials - By modifying the authentication process linked to hybrid identities, attackers can gain persistent privileged access to cloud resources
T1621	Multi-Factor Authentication Request Generation	- Attackers can bypass the multi-factor authentication (MFA) mechanism and and attempt to access accounts by sending MFA requests to users - If the attacker lacks credentials for the victim's account, they can exploit this option even when configured for self-service password reset (SSPR) they can exploit the automatic push notification generation feature
T1528	Steal Application Access Token	- Attackers can steal application access tokens as a means to obtain credentials as a means to obtain credentials for accessing remote systems and resources - Application access tokens are used to make authorized API requests on behalf of a user or service and are commonly used to access container-based applications and SaaS resources. container-based applications and accessing SaaS resources. - In cloud and containerized environments, attackers who have stolen account API tokens can access data and perform actions with the account's privileges
T1649	Steal or Forge Authentication Certificates	- Attackers can steal or forge certificates used for authentication to access remote systems or resources - Configuration errors related to certificates allow users to obtain accounts or privileges associated with the certificate's Subject Alternative Name (SAN) to gain access to privileged accounts or privileges
T1539	Steal Web Session Cookie	- Steal session cookies from web applications or services to access web applications or internet services as an authenticated user without credentials - Web applications and services often use session cookies as authentication tokens after users authenticate on the website often use session cookies as authentication tokens - Other applications on the target system (apps authenticating to cloud services) may also store sensitive authentication cookies in memory, and session cookies can be used to bypass some multi-factor authentication protocols
T1552.001	Unsecured Credentials: Credentials In Files	- Attackers can search for files containing stored credentials in the local file system and remote file shares stored in files - In cloud and/or containerized environments, service account credentials for authenticated users and are often stored in local configuration and credential files

TID	Technique Category	Technique Description
T1552.005	Unsecured Credentials: Cloud Instance Metadata API	• Attackers may attempt to access the Cloud Instance Metadata API to collect credentials and other sensitive data. Cloud Instance Metadata API • Cloud service providers offer a service that provides Cloud Instance Metadata API support • Through this API, applications can access information about running virtual instances. information about running virtual instances • If an attacker exists on a running virtual instance, they can query the instance metadata API directly to identify credentials to gain access to additional resources
T1552.008	Unsecured Credentials: Chat Messages	• Attackers can obtain credentials stored or transmitted through user chat messages Users can obtain credentials stored or transmitted through user chat messages • Users may share various forms of credentials (account names, passwords, API keys, authentication tokens, etc.) various forms of credentials (account names, passwords, API keys, authentication tokens, etc.) through internal corporate communication channels (private or public). These credentials can be exploited to perform subsequent activities such as internal movement or privilege escalation.

7) Discovery

Comprises attack methods used by attackers to gather information about a system or network to understand the environment and set the next attack target. In cloud environments, information gathering activities are often performed via APIs, and attackers can collect all information about the environment, including infrastructure, storage, services, accounts, and permissions.

[Table17] List of techniques used in the Discovery tactics of the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1087.004	Account Discovery: Cloud Account	• Attackers may attempt to collect a list of cloud accounts • In Azure CLI (AZ CLI), use 'az ad user list'; in AWS, use 'aws iam list-users', 'aws iam list-roles', GCP: 'gcloud iam service-accounts list', ‘gcloud projects get-iam-policy’ commands
T1580	Cloud Infrastructure Discovery	• In an Infrastructure as a Service (IaaS) environment, you may attempt to collect available infrastructure and and resources available in an Infrastructure as a Service (IaaS) environment. • AWS can be obtained via the DescribeInstances API, and GCP's Cloud SDK CLI can be obtained via the 'gcloud compute instances list' command
T1538	Cloud Service Dashboard	• Using stolen credentials, attackers can gain useful information about the operational cloud environment, such as specific services, resources, and features, through the Cloud Service Dashboard GUI
T1526	Cloud Service Discovery	• After gaining access to a system, an attacker may attempt to enumerate cloud services running on the system.
T1619	Cloud Storage Object Discovery	• They can enumerate objects within the cloud storage infrastructure, and an attacker could use this information to request all objects in the cloud storage or a specific object • Cloud service providers offer APIs that allow users to enumerate objects stored in cloud storage
T1654	Log Enumeration	• System and service logs can be analyzed to obtain useful information • In a cloud environment, attackers can utilize utilities such as the Azure VM agent (CollectGuestLogs.exe) to collect security logs from cloud infrastructure T1046
T1046	Network Service Discovery	• Attempt to obtain a list of services running on remote hosts and local network infrastructure devices Attackers may attempt to obtain a list of services running on remote hosts and local network infrastructure devices • Within a cloud environment, an attacker may attempt to discover services running on other cloud hosts • If the cloud environment is connected to an on-premises environment, attackers may also identify services running on non-cloud systems
T1040	Network Sniffing	• Passively sniffing network traffic to steal environment information transmitted over the network including credentials transmitted over the network • In cloud-based environments, attackers can use traffic mirroring services to sniff the network traffic of virtual machines

TID	Technique Category	Technique Description
T1201	Password Policy Discovery	• Attackers may attempt to access detailed information about password policies used in corporate networks or cloud environments to access detailed information about password policies • AWS allows you to use the GetAccountPasswordPolicy API to to obtain password policies
T1069.003	Permission Groups Discovery: Cloud Groups	• You can attempt to collect cloud groups and permission settings • Azure CLI (AZ CLI) and Google Cloud Identity Provider API provide interfaces to obtain permission groups • Attackers can use this information to target accounts with permissions to specific objects or leverage already compromised accounts to access objects
T1518.001	Software Discovery: Security Software Discovery	• Attackers may attempt to collect lists of defense tools and sensors, configurations, defensive tools, and sensors installed on systems or in cloud environments • Attackers may attempt to collect a list of security software, configurations, Google Cloud Monitor agents, and other agents installed on the computing infrastructure cloud-native security software installed on the computing infrastructure
T1082	System Information Discovery	• Attackers may attempt to obtain detailed information about operating systems and and hardware, such as operating system and hardware details
T1614	System Location Discovery	• Information gathering may be performed to determine the geographic location of the victim host • Attackers may attempt to infer the system's location using various system checks, such as time zone, keyboard layout, and/or language settings • In cloud environments, the availability zone of an instance can be accessed from the instance by accessing the instance metadata service
T1049	System Network Connections Discovery	• Queries information over the network to obtain a list of network connections from the compromised system currently being accessed or remote systems • Attackers accessing systems that are part of a cloud-based environment can map virtual private clouds or virtual networks to identify connected systems and services • The retrieved information may contain network cloud details relevant to the attacker's objectives environment

8) Lateral Movement

This consists of techniques attackers use to move from an compromised system to other systems within the network, expanding their access scope. In cloud environments, lateral movement exploits trust relationships and shared resources between cloud services to execute malicious files on other internal systems. With the widespread adoption of SaaS platforms, malicious files can be distributed via cloud-based shared drives or code repositories.

[Table18] List of techniques used in the Lateral Movement tactics within the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1534	Internal Spearphishing	• After gaining access to accounts or systems within the environment, uses internal spearphishing to access additional information or compromise other accounts within the same environment. compromise other accounts within the same environment. • As part of internal spearphishing, attackers may use attachments or links to deliver payloads or redirect victims to external sites to steal credentials from phishing sites.
T1021.007	Remote Services: Cloud Services	• Synchronized with on-premises user accounts or using shared accounts Accessible within compromised environments • Attackers can use the Cloud API, Azure PowerShell, or Google Cloud CLI commands Cloud API, Azure PowerShell, or Google Cloud CLI commands to connect to available cloud services
T1021.008	Remote Services: Direct Cloud VM Connections	• Directly log in to cloud-hosted computing infrastructure accessible via cloud-native methods using valid accounts • Cloud providers include Azure Serial Console, AWS EC2 Instance Connect, AWS System Manager, and provide interactive connections to virtual infrastructure accessible through cloud APIs. Interactive connections are provided for virtual infrastructure accessible via cloud APIs such as Azure Serial Console, • Attackers can use cloud-based methods to directly access virtual infrastructure and switch environments.
T1072	Software Deployment Tools	• Access centralized software suites installed within the enterprise to and execute commands and perform internal movement • SaaS-based configuration management services support a wide range of cloud management commands on cloud-hosted instances and execute arbitrary commands on on-premises endpoints • Microsoft Configuration Manager enables global administrators or Intune administrators to run scripts with SYSTEM privileges on on-premises devices connected to Entra ID
T1080	Taint Shared Content	• Attackers can add content to shared storage locations such as network drives or internal code repositories to deliver payloads to remote systems • Content stored on network drives or other shared locations can be be added to legitimate files as malicious programs, scripts, or malicious code. and can use infected shared content to to perform lateral movement.

TID	Technique Category	Technique Description
T1550.001	Use Alternate Authentication Material: Application Access Token	• Attackers can steal application access tokens to bypass standard authentication procedures and gain access to restricted accounts, information, or services on remote systems • Application access tokens are used to make authorized API requests on behalf of a user or service to make authorized API requests on behalf of a user or service. They are commonly used to access resources in cloud, container-based applications, and SaaS . • In AWS and GCP environments, attackers can trigger short-lived access token requests with privileges belonging to other user accounts. They can then use these tokens to request data or perform actions not possible with the original account.
T1550.004	Use Alternate Authentication Material: Web Session Cookie	• An attacker can use a stolen session cookie to authenticate to web applications and services • After acquiring the cookie, attackers can access sensitive information, read emails, or perform actions authorized for the victim's account.

9) Collection

This is the stage where attackers identify and collect data to achieve their objectives. Cloud environments are critical collection targets for attackers due to their centralized data storage characteristics. Attackers can exploit automation features and APIs within cloud environments to gather data.

[Table19] List of techniques used in the Collection tactics of the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1119	Automated Collection	• Attackers who have infiltrated a system or network can use automated techniques to collect internal data • In cloud-based environments, attackers can use cloud APIs, data pipelines, command-line interfaces, or ETL (Extract, Transform, Load) services to automatically collect data
T1530	Data from Cloud Storage	• Sensitive data can be collected from cloud storage solutions • Unintentionally granting public access to unauthorized users, granting overly broad access permissions to all users, or even allowing anonymous users outside the control of the identity access management system to anonymous users outside the control of the identity access management system
T1213	Data from Information Repositories	• Attackers can exploit information repositories to obtain information. • Information repositories are tools that store information to facilitate collaboration or information sharing among users. and can be used to store information. They can store various data that may help achieve attackers' additional objectives or provide direct access to target information
T1074.002	Data Staged: Remote Data Staging	• The attacker can store data collected from multiple systems prior to the leak in a central or a single system directory • In cloud environments, attackers may store data within specific instances or or create a cloud instance and store data within that instance
T1114.002	Email Collection: Remote Email Collection	• Attackers can target Office 365 or Google Workspace as targets to collect sensitive information.
T1114.003	Email Collection: Email Forwarding Rule	• Attackers can set up email forwarding rules to collect sensitive information • Attackers can exploit email forwarding rules to monitor victims' activities, steal information, and obtain additional information about the victim or the victim's organization about the victim or the victim's organization.

10) Exfiltration

This is the stage where attackers exfiltrate collected data to external systems. After gathering data, attackers may package it using compression and encryption to evade detection during removal. Data exfiltration in cloud environments is challenging to distinguish from legitimate cloud service traffic due to near-unlimited bandwidth and the widespread use of encrypted channels (HTTPS).

[Table20] List of techniques used in the Exfiltration tactics of the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1048	Exfiltration Over Alternative Protocol	• Data exfiltration via a protocol different from the existing C2 channel • IaaS and SaaS platforms (AWS S3, Microsoft Exchange, Microsoft SharePoint, etc.) support downloading for files, source code, and other critical information
T1567.004	Exfiltration Over Web Service: Exfiltration Over Webhook	• Data exfiltration via Webhook endpoints instead of traditional C2 channels • Services like Discord and Slack offer webhook endpoints create Webhook endpoints that can be used by other services like GitHub, Jira, and Trello
T1537	Transfer Data to Cloud Account	• Attackers can transfer data to other cloud accounts through sharing/synchronization and backup creation in cloud environments to leak data by transferring it to other cloud accounts • Attackers can exploit cloud-based file sharing services, such as generating anonymous file sharing links or Azure SAS (Shared Access Signature) URIs, to share data to their attacker cloud accounts

11) Impact

This is the stage where attackers manipulate, interrupt, or destroy the availability or integrity of systems and data to achieve their objectives. In cloud environments, attacks at the impact stage can cause significant financial loss, service disruption, and a decline in trust. Attackers can mine cryptocurrency by unauthorized use of cloud resources. Following a data breach, they can permanently delete or irrecoverably overwrite critical operational data such as cloud storage objects, machine images, and database instances.

[Table21] List of techniques used in the Impact tactics of the MITRE ATT&CK Cloud Matrix

TID	Technique Category	Technique Description
T1531	Account Access Removal	• Block access to the user's account to prevent access to system and network resources • Account deletion, locking, or manipulation (e.g., changing credentials).
T1485.001	Data Destruction: Lifecycle-Triggered Deletion	• Modify the lifecycle policy of a cloud storage bucket to delete all stored objects • Using cloud storage buckets, users can automate the migration, archiving, or deletion of objects If an attacker has permission to modify the policy, • If an attacker has permission to modify these policies, they could delete all objects at once
T1486	Data Encrypted for Impact	• An attacker can encrypt data on a target system or within a network to prevent access to system and network resources • In cloud environments, storage objects within compromised accounts can be encrypted. • In an AWS environment, an attacker can utilize services such as server-side encryption with customer-provided keys (SSE-C) to encrypt data. encryption (SSE-C) to encrypt data.
T1491.002	Defacement: External Defacement	• Attackers can deliver messages to users or pose threats through an organization's systems. threaten them • and cause users to lose trust in the system, potentially to spread political messages or propaganda
T1667	Email Bombing	• Attackers can send large volumes of messages to specific email addresses • Normal emails may not be received accurately, potentially disrupting business operations may be disrupted
T1499	Endpoint Denial of Service	• Attackers can perform by performing an Endpoint Denial of Service (DoS) attack
T1657	Financial Theft	• Attackers target financial gain through various attack types, including ransomware, business email compromise (BEC) and fraud, cryptocurrency network exploitation.
T1490	Inhibit System Recovery	• Terminates services for recovering compromised systems and can delete embedded data from the system. • On ESXi servers, attackers can delete or encrypt virtual machine snapshots encrypt them to prevent their use as backups, and delete folders synchronized with cloud services to erase online backups

TID	Technique Category	Technique Description
T1498	Network Denial of Service	• Attackers can exhaust the network bandwidth used by services T1498.001
T1496.001	Resource Hijacking: Compute Hijacking	• Attackers can utilize computer resources to mine cryptocurrency
T1496.002	Resource Hijacking: Bandwidth Hijacking	• Attackers can utilize a system's network bandwidth to launch network denial-of-service attacks or distribute malicious torrents distribute malicious torrents
T1496.003	Resource Hijacking: SMS Pumping	• After obtaining a phone number from a telecommunications provider, an attacker can use the victim's messaging infrastructure to send massive volumes of SMS messages to that phone number
T1496.004	Resource Hijacking: Cloud Service Hijacking	• Attackers can use compromised SaaS applications to perform resource-intensive tasks • Attackers can leverage email and messaging services like AWS Simple Email Service (SES), AWS Simple Notification Service (SNS), SendGrid, and Twilio to send large volumes of spam and phishing emails

3.4. AWS Incident Response Framework Investigation

AWS has presented a five-step security incident response framework based on the NIST SP 800-61 incident response standard. This framework reflects the dynamic nature of cloud environments and log-centric investigation methods, enabling cloud-using organizations to perform integrated responses across all AWS resources. The framework specifically proposes AWS security services and logs available for each step. Key details regarding the procedures presented in the framework are as follows.

[Table22] AWS Security Incident Response Framework Procedures

Stage	Description
Preparation (Preparation)	Establishes response systems before an incident occurs and completes organizational and technical preparations # Key Tasks • Apply the IAM principle of least privilege, protect root accounts • Enable CloudTrail, AWS Config, VPC Flow Logs, and Route 53 Resolver Logs • Configure S3-based log storage (Evidence Bucket) • Prepare automation tools such as Systems Manager, Lambda, and EventBridge • Conduct response drills based on Gamedays and Runbooks # Key AWS Services • IAM, CloudTrail, AWS Config, S3, Systems Manager, Lambda
Detection and Analysis (Detection and Analysis)	The stage of detecting anomalies and analyzing logs and events to determine the cause and impact of an incident # Key Tasks • Detect abnormal behavior using GuardDuty (IAM abuse, malicious IP communications, etc.) • Aggregating results from multiple security services via Security Hub • Event Correlation Analysis Using Detective • CloudTrail, VPC Flow Logs, DNS Logs, and other log analysis • Incident classification and prioritization based on alerts and detection results # Key AWS Services • GuardDuty, Security Hub, Detective, CloudTrail, CloudWatch
Isolation (Containment)	The step of quickly isolating resources where attacks are detected and blocking access to prevent propagation # Key Actions • Network isolation of infected instances (modify security groups, NACLs) • Disable IAM keys and session tokens • EventBridge + Lambda-based automated isolation trigger • Remote remediation via System Manager Session Manager # Key AWS Services • System Manager, Lambda, EventBridge, Network Firewall, IAM
Eradication and Recovery (Eradication and Recovery)	Steps to remove traces of compromise and restore the system to a normal state # Key Tasks • Remove malware, unauthorized accounts, and backdoors • System restoration via snapshots (EBS, RDS) or backups • Redistribution of CloudFormation Stack and verification of Config rules • Reactivation of security monitoring after recovery # Key AWS Services • Backup, CloudFormation, AWS Config, EBS, Systems Manager

Stage	Description
Post-Improvement (Post-Incident Response)	Perform continuous learning through root cause analysis, process improvement, and detection rule refinement after incident closure # Key Tasks • Prepare incident reports and Lessons Learned documentation • Analysis of bottlenecks and failure causes in the response process • Update GuardDuty and Security Hub detection rules • Improve Lambda and EventBridge automation logic • Conducting regular retraining and repeating GameDay exercises # Key AWS Services • Security Hub, GuardDuty, SSM Runbook

3.5. AWS Security Service Investigation

Incident response in cloud environments relies on achieving log-centric data visibility. AWS provides various native security services for this purpose, with each service interconnected and operated across the 'detection-analysis-response' stages. The primary security services utilized in AWS-based incident response are as follows.

[Table23] List of Key Security Services Used in AWS-Based Incident Response

Number	Service Name	Primary Role	Incident Response Phase
1	Amazon GuardDuty	Anomaly Detection (Behavior-Based)	Detection
2	Amazon CloudWatch	Integrated monitoring of metrics, logs, and events; alarms and anomaly detection; Event-Based Automation	Detection, Analysis, Containment, Post-Incident Response
3	Amazon Detective	Log correlation analysis, root cause investigation	Analysis
4	Amazon Athena	Forensic/Correlation Analysis on S3 Logs Using SQL	Analysis, Post-Incident Response
5	Amazon Security Hub	Security Results Integration, Compliance Management	Detection, Post-Incident Response
6	AWS Systems Manager	Response and Recovery Automation (Runbooks/Remote Commands/Sessions)	Containment, Eradication & Recovery
7	Amazon Macie	S3 Sensitive Data Identification & Risky Bucket Detection	Preparation, Detection, Post-Incident Response
8	AWS Config	Track resource configuration changes, assess policy violations, Automate remediation	Preparation, Detection, Post-Incident Response
9	Amazon Inspector	Vulnerability Scanning (CVE·Network Exposure)	Preparation, Detection
10	Prowler	Security Configuration Audit, Vulnerable Settings Discovery, Compliance Assessment	Preparation, Post-Incident Response
11	Self-Service Security Assessment (SSSA)	Self-Service Security Assessment/Readiness/Compliance Evaluation	Preparation, Post-Incident Response

Detailed information for each major security service (service description, key features, key characteristics, incident response integration plans) is as follows.

1) AWS GuardDuty

AWS GuardDuty is a managed, intelligent threat detection service that analyzes logs across your entire AWS environment to automatically detect abnormal behavior, account takeovers, network intrusions, and data leaks. Key features include analysis of CloudTrail/VPC Flow Logs/DNS Log/EKS Audit Log, IAM Anomaly Detection, S3 Data Access Monitoring, generation and classification of Findings, and automated follow-up actions.

[Table24] AWS GuardDuty Key Features

Category	Description
Managed Threat Detection Service	Automatically analyzes AWS internal logs without requiring separate agent installation or infrastructure operation
Log-based anomaly detection	CloudTrail, VPC Flow Logs, DNS Query Log, EKS Audit Log, etc. Analyzes key logs to identify authentication abuse, data leaks, abnormal API calls, and external communication attempts
Machine learning and threat intelligence-based detection	Utilizes AWS proprietary ML models + AWS Threat Intelligence + Partner Feeds (MISP, Abuse.ch, etc.) to detect C2, phishing, malicious IPs, etc.
Real-time continuous monitoring	Performs 24/7 monitoring across all regions, generating detection results near real-time
Automated Response Integration	Integrated with Security Hub, EventBridge, Lambda, and Systems Manager for automated isolation and alerts
Multi-account and organizational unit unified management	Integrated with AWS Organizations to manage detection across multi-account environments
Cost efficiency and uninterrupted operation	Provides low-cost/non-disruptive monitoring based on log sampling and metadata analysis

[Table25] AWS GuardDuty Incident Response Integration Approach

Step	Role of AWS GuardDuty
Detection	Log-based anomaly detection and generation of Findings
Analysis	Integrate findings into Security Hub, perform correlation analysis with Detective
Containment	Execute Lambda containment workflows via EventBridge triggers
Post-Incident Response (Post-IR)	Enhance detection rules, update Custom Threat List

2) Amazon CloudWatch

Amazon CloudWatch is a managed service that enables centralized collection, monitoring, storage, and analysis of log files from AWS resources, applications, and on-premises servers. It detects performance anomalies or signs of failure early, sends alerts, and enhances operational efficiency and security visibility through log-based analysis. By integrating with other security services like CloudTrail, GuardDuty, and Config, it leverages security events as triggers, serving as the foundation for incident detection and automated remediation within AWS environments.

Key features include collecting key metrics for AWS resources, centralized log management, generating alerts and dashboards based on metrics and logs, event-based automation triggers, log pattern analysis, and automatic detection of abnormal metric changes.

[Table26] Amazon CloudWatch Key Features

Category	Description
Integrated Monitoring Hub	Enables unified management of metrics and logs for all AWS resources within a single service
Automation and Scalability	Build automated response workflows by integrating with EventBridge, Lambda, and SNS
Real-time Anomaly Detection	Identify abnormal behavior in real time with metrics-based anomaly detection
Support for long-term storage and analysis	Export log data to S3 and OpenSearch for long-term storage, search, and visualization
Security service integration	Achieve complete visibility and automated response when combined with GuardDuty, Config, CloudTrail, and more
Enables integrated analysis of operations and security	Correlation analysis of performance issues and security events within the same environment (e.g., detect CPU spikes + suspicious API calls in conjunction)

[Table27] Amazon CloudWatch Incident Response Integration Approach

Step	Amazon CloudWatch Role
Detection	Send CloudTrail logs to CloudWatch Logs, then configure real-time alerts for specific API events (e.g., CreateUser, DeleteBucket) Detect abnormal traffic like unauthorized IP access and port scans via VPC Flow Logs collection Immediately notify security personnel via CloudWatch Alarms when GuardDuty/Config Findings are detected
Analysis	Analyze logs by timeframe and track attacker activity sequences using CloudWatch Logs Insights Analyze root causes using Contributor Insights, including attacker IPs, user IDs, and called APIs
Containment	Automatically isolate incidents upon security event detection using CloudWatch Events → Lambda automated execution framework
Post-Incident Response (Post-IR)	Long-term storage of CloudWatch Logs and Metrics data (S3 Export) for use as forensic evidence Reproduce the system state at the time of the incident and conduct reviews to validate response effectiveness

3) AWS Detective

AWS Detective is a tool that automatically models relationships between AWS accounts, resources, IAM activities, and network logs to support root cause analysis and correlation investigations for security incidents. Detective is a service capable of tracing past activities based on up to one year of event history. It integrates with other AWS services (such as GuardDuty, Security Hub, Security Lake, etc.) to enable one-click investigations.

Key features include Findings investigations, behavior profiling, graph-based automated relationship modeling, timeline analysis, summaries of recent activity/connection events for specific resources, and integration with IR Playbooks.

[Table28] AWS Detective Key Features

Category	Description
Managed Security Investigation Service	No need for separate infrastructure operation, log storage, or indexing Automatically collects and analyzes findings from GuardDuty and Security Hub
Relationship-Based Data Modeling	Visualizes relationships between AWS resources (accounts, EC2, IAM, IP, S3, etc.) as a graph to enable correlation analysis between incidents and explore correlations
Integrated log correlation analysis	Integrate and analyze CloudTrail, VPC Flow Logs, GuardDuty Findings, EKS Audit Logs, etc., to track "who did what, when, and where"
Automated Data Collection and Retention	Automatically collects and summarizes selected log sources, storing them for up to one year for efficient forensic analysis
Visual root cause analysis	Visually display the user, resource, and action timeline for specific events (e.g., malicious IP, IAM permission misuse) in the console and action timelines for specific events (e.g., malicious IP, IAM permission misuse)
GuardDuty, Security Hub, IAM Access Analyzer	Investigate GuardDuty detection events in detail with a single click in the Detective console
Cost efficiency and zero-downtime operation	Automatically summarizes and indexes logs for analysis within AWS, offering lower costs compared to SIEM Enables low-cost, high-speed analysis

[Table29] AWS Detective Incident Response Integration Approach

Step	Role of AWS Detective
Detection	Receive GuardDuty Findings and secure targets for analysis
Analysis	Perform correlation analysis based on CloudTrail and VPC Flow Logs to identify root causes
Containment	Establishing grounds for blocking attacker access paths, supporting IAM reconfiguration
Post-Incident Response (Post-IR)	Providing insights for improving detection rules to prevent recurrence

4) Amazon Athena

Amazon Athena is a serverless interactive query service that enables you to query and analyze data in Amazon S3 directly using standard SQL (based on Presto/Trino).

Key features include integration with data catalogs, support for defining path-based partitions to minimize scan volume, support for Materialize using columnar formats, and the ability to utilize UDFs (User Defined Functions) and JSON functions.

[Table30] Amazon Athena Key Features

Category	Description
Serverless·Instant Query	No cluster management required; execute directly from console/CLI/JDBC
S3 Native	Supports CSV/JSON/Parquet/ORC/Avro, compressed formats (Gzip, Snappy, etc.)
Schema on Read	Define tables in Glue Data Catalog and analyze immediately without moving files
Distributed SQL engine	Parallel processing of large-scale data based on Presto/Trino
Integrated ecosystem	Directly query S3 logs such as CloudTrail, ALB/ELB logs, VPC Flow Logs, WAF/CloudFront logs with SQL
Security/Governance	IAM, Lake Formation, data masking/column-level permissions, S3/KMS encryption, encrypted query results
Connectivity	JDBC/ODBC, QuickSight visualization, Pandas/BI tool integration

[Table31] Amazon Athena Incident Response Integration Plan

Phase	Amazon Athena Role
Detection, Analysis	Correlation analysis of CloudTrail/VPC Flow/WAF/ALB/CloudFront logs logged in S3 using SQL Reconstruct attack timelines, track suspicious IAM activities/source IPs/resources
Post-Incident Response (Post-IR)	Identify recurrence prevention rules (based on anomaly patterns), derive candidate detection rules

5) AWS Security Hub

AWS Security Hub is a service that helps integrate, analyze, and manage security findings from various AWS security services and third-party solutions within a single console.

Key features include: consolidating Findings results, automatically checking for compliance with standards, filtering and prioritizing Findings, configuring automated isolation and notification workflows, managing aggregated account identity Findings in AWS Organization environments, and automatically updating and reflecting security events in real-time dashboards.

[Table32] AWS Security Hub Key Features

Category	Description
Security Findings Integration	Integrated management of detection results (Findings) from AWS security services like GuardDuty, Inspector, Macie, Detective, and third-party tools Integrated management of detection results (Findings)
Standardized Result Format	Standardize all security findings into a common JSON format (ASFF) for automated analysis and easy integration
Security posture assessment	Automatically assess your account's security compliance status based on standard regulations such as AWS CIS Benchmark, PCI DSS, and NIST 800-53 Security Compliance Status
Automated Action Integration	Trigger automated actions for findings using EventBridge rules and Lambda automation functions
Multi-Account Management and Organization Integration	Centrally aggregate and monitor security findings across multiple AWS accounts at the organizational level
Dashboard visualization	Provides graphical representations of security findings, CIS compliance status, and detection trends
Integration with other security tools	API integration with security solutions like Splunk, CrowdStrike, and Palo Alto Prisma Cloud

[Table33] AWS Security Hub Incident Response Integration Approach

Step	Role of AWS Security Hub
Detection	Aggregates and collects results from GuardDuty, Inspector, etc., in a standardized format
Analysis	Prioritize findings, eliminate duplicates, and verify compliance status
Containment	Trigger automated response scenarios using EventBridge and Lambda
Post-Incident Response (Post-IR)	Policy Enhancement Based on Secure Score and Lessons Learned

6) AWS Systems Manager (SSM)

AWS Systems Manager is an automation service that centrally controls AWS instances, applications, accounts, and components, automating security and operational tasks to enhance operational efficiency and incident response speed.

Key features include remote command execution on EC2 and other services, automation of repetitive tasks using SSM Documents, automated OS and application patch management, continuous monitoring of instance configuration status, terminal access without SSH keys (Session Manager), secure storage and retrieval of passwords, API keys, and configuration values through encryption, and assessment of compliance with patch/configuration status requirements.

[Table34] AWS Systems Manager Key Features

Category	Description
Centralized Management	Control EC2, on-premises servers, and hybrid environments from a single console
Automated Operations and Security Actions	Automated execution of patching, isolation, and recovery procedures via Lambda, Run Command, and Automation Documents (SSM Doc) Automated execution of patching, isolation, and recovery procedures
Agent-Based Management	Agents installed on each EC2 and on-premises instance securely execute commands
Granular permission control	Manage permissions for each command and Runbook execution in detail via IAM policies
Security event response integration	Automated actions triggered by GuardDuty Findings or EventBridge events
Patch, configuration, and compliance management	Maintain security patches and configuration standards with Patch Manager, State Manager, and Compliance features
Support for hybrid environments	Register on-premises systems to AWS Systems Manager Fleet to apply the same policies and patches

[Table35] AWS Systems Manager Incident Response Integration Approach

Stage	Role of AWS Systems Manager
Detection	Automatically triggered upon GuardDuty Findings occurrence
Containment	Disconnect EC2 network interfaces or modify security groups via Run Command or Automation
Eradication	Terminate malicious processes and delete backdoor files using automated scripts
Recovery	Restore snapshots, apply patches, restart services using SSM Documents
Post-Incident Response (Post-IR)	Verify compliance status using compliance features

7) Amazon Macie

Amazon Macie is a serverless data security and privacy service that automatically identifies sensitive information (such as PII) in data stored in Amazon S3 using machine learning and pattern matching, finds risky buckets/objects, and provides alerts (Findings).

Key features include scan target/sampling/identifier set/schedule configuration, bucket assessment, Findings management, and custom adjustments.

[Table36] Amazon Macie Key Features

Category	Description
Sensitive Information Identification	Name/Social Security Number/Passport/Credit Card/Account, etc. Supports Managed Data Identifiers + Custom Identifiers (regular expressions/keywords)
S3 Full/Selective Scan	Set scan cycles for data within S3 and selectively scan only desired data ranges (buckets, prefixes)
Risk Visibility	Bucket-level risks such as Public/Shared buckets, lack of encryption, excessive policies, and Separate reporting of object-level sensitivity
Multi-Account Consolidation	Manage all accounts collectively from the management account using AWS Organizations
Integration and Automation	Integrate findings into Security Hub, link automated actions via EventBridge → Lambda/SSM
Serverless·Managed	No agents/clusters required, pay-as-you-go pricing (based on scan bytes and assessment counts)

[Table37] Amazon Macie Incident Response Integration Plan

Step	Amazon Macie Role
Preparation	Identify sensitive data holdings, determine high-risk bucket scope
Detection, Analysis	Identify potential data leakage points, analyze in conjunction with bucket policies/access paths
Containment	Automate Public Access Block, policy modification, and object encryption via EventBridge triggers
Post-Incident Response (Post-IR)	Quantify the scope of the leak (objects/fields), supplement recurrence prevention criteria/identifiers

8) AWS Config

AWS Config is a service that continuously records, evaluates, and corrects the configuration state of cloud resources. It monitors security policy compliance before incidents occur and supports forensic analysis and root cause investigation after incidents by providing change history and point-in-time configuration restoration. Specifically, it automatically records and analyzes "how resources were configured," "when, who, and what changes were made," and "whether they currently comply with policy standards" within an AWS account. Unlike CloudTrail, it handles state-centric data rather than actions, tracking resource configurations as point-in-time snapshots.

Key features include configuration history tracking, configuration snapshots, rule-based assessments, automatic actions via SSM Runbook invocation upon rule violations, centralized querying and assessment of Config data across multiple accounts and regions, and dependency tracking through resource mapping.

[Table38] AWS Config Key Features

Category	Description
State-centric logging	While CloudTrail records 'what was done', Config records 'what was configured and how'.
Continuous Monitoring	Automatically evaluates every resource change, detecting policy drift in real time
Policy Compliance and Compliance Support	Implement policies for major security standards like CIS, NIST, and PCI-DSS as Config Rules
Integrated Management	Integrated with Security Hub, CloudTrail, SNS, S3, SSM, etc., enabling automated 'Detect–Analyze–Act'
Forensic support	Restore resource states at specific points in time as JSON Snapshots to recreate the environment at the time of an incident

[Table39] AWS Config Incident Response Integration Approach

Step	Role of AWS Config
Preparation	Enable compliance monitoring based on Config Rules for key resources (S3, IAM, Security Groups, etc.)
Detection	Detect abnormal configurations when policy violations occur during configuration changes
Analysis	Verify the resource configuration state and the actor responsible for changes at the time of the incident
Containment, Recovery	Automatically correct violations or restore to previous state
Post-Incident Improvement (Post-IR)	Re-verify whether drift occurred after the incident, redistribute improved rules

9) Amazon Inspector

Amazon Inspector is a service that automatically scans AWS instances, containers, and software packages for vulnerabilities and assesses security risks based on CVSS scores, impact, patch status, and more.

Key features include EC2 instance vulnerability assessment, container image scanning, Lambda function scanning, analysis of publicly accessible paths to assess attack exposure, findings management and prioritization, and automated notifications and action integration.

[Table40] Amazon Inspector Key Features

Category	Description
Automated Vulnerability Scanning	Automatically analyzes code, packages, and components of EC2, ECR (container images), and Lambda functions Detects CVE vulnerabilities
Agent-based real-time assessment	Evaluates EC2 OS, packages, Network configuration on EC2 in real time
Risk assessment based on CVE/CVSS	Calculate vulnerability risk levels based on the National Vulnerability Database (NVD) and AWS proprietary metrics Calculate vulnerability risk levels
Security standard integration	Security standards mapped to CIS Benchmark, NIST 800-53, etc.
Security Allowance Policy-Based Exception management	Ability to temporarily ignore or handle specific vulnerabilities as exceptions
Automatic transmission of findings and Integrated management	Automatically send detection results to Security Hub for centralized management
Continuous Monitoring	Automatically triggers scans when new instances or new container images are created

[Table41] Amazon Inspector Incident Response Integration Plan

Step	Amazon Inspector's Role
Preparation	Pre-identification of Vulnerabilities and Misconfigurations
Detection	CVE-based system, container, and Lambda vulnerability detection
Containment, Recovery	Support for patching and security configuration remediation actions
Post-Incident Response (Post-IR)	Providing vulnerability improvement guides to prevent recurrence

10) Self-Service Security Assessment (SSSA)

Self-Service Security Assessment is a self-service security assessment toolkit that evaluates compliance status by checking AWS security configurations and operational procedures against checklists/guides. This toolkit is deployed as an AWS CloudFormation template and assesses current AWS settings against security best practices.

Key features include domain-specific checks, evidence management, and report generation.

[Table42] Key Features of Self-Service Security Assessment

Category	Description
Self-check (self-assessment) focused	Review policy/procedure/configuration status item by item (record evidence links, responsible personnel, deadlines)
Security Standard and Best Practice Mapping	Mapping CIS, NIST 800-53, ISO 27001, etc., to AWS best practices
Automation/Manual Hybrid Approach	Automatically verify some items using Config and Security Hub data; verify procedures and organizational items through interviews and document review
Score and priority calculation	Derive risk/maturity scores and improvement backlog

[Table43] Self-Service Security Assessment Incident Response Integration Plan

Stage	Role of Self-Service Security Assessment
Preparation	Check IR Readiness (CloudTrail/GuardDuty/Backup/Access Control/Contact Network/Training)
Detection Support	Preventing Detection Failures
Post-Incident Improvement (Post-IR)	Elevate Security Levels Through Reassessment

11) Prowler

Prowler is a CLI tool that supports AWS security checks, audits, hardening, and incident response.

Key features include providing a security assessment engine based on security standards (CIS AWS Foundation, NIST SP 800-53, ISO/IEC 27001, PCI DSS, etc.), batch assessment of organizational units, coverage of core services and domains, result formatting and integration, customization aligned with organizational policies, and CI/CD and scheduling operations.

[Table44] AWS Prowler Key Features

Category	Description
Comprehensive Security Audit for AWS Environments	IAM, CloudTrail, Config, S3, Security Group, KMS, GuardDuty, etc. Automatically inspects settings across the entire AWS environment
Standard-Based Assessment	Provides assessments mapped to key regulations like CIS AWS Foundations Benchmark, NIST SP 800-53, ISO 27001, PCI DSS, GDPR, etc.
Automation and Scheduling Support	Enables regular checks via CI/CD pipelines (GitHub Actions, GitLab CI, Jenkins, etc.) or Lambda/CloudWatch Events
Multi-account and multi-region support	Simultaneously inspect multiple accounts and regions via AWS Organizations or AssumeRole
Reporting and integrated management features	Output reports in CSV, JSON, or HTML formats; send results to Security Hub
Customizable Support for configuring check items	Configure inspection items based on specific regulations (e.g., CIS Level 1/2) or your own policies as needed
Cloud-Native Integration	Operates via AWS CLI/API, eliminating the need for external agent installation

[Table45] AWS Prowler Incident Response Integration Plan

Stage	Prowler's Role
Preparation	Check the security environment before an incident occurs to proactively identify and improve vulnerable configurations
Detection	Unlike GuardDuty, it does not provide real-time detection, but it identifies potential incident factors such as incorrect log and permission settings
Analysis	Post-incident re-evaluation of the environment to identify configuration vulnerabilities that attackers may have exploited
Containment, Recovery	Enables automated reset and patch procedures based on report findings, integrated with SSM, Config, etc.
Post-Incident Improvement (Post-IR)	Post-incident security reassessment and recurrence prevention checks based on CIS standards

12) AWS Shield

AWS Shield is a managed service that detects and mitigates DDoS attacks occurring at the network, transport, and application layers.

Key features include real-time DDoS detection and mitigation, multi-layered protection, dedicated response team coordination, cost protection, policy and group-based protection operations, and automated response integration.

[Table46] AWS Shield Key Features

Category	Description
Continuous Monitoring & Automatic Mitigation	Learns traffic patterns globally at edge/backbone to mitigate attacks in real-time (Inline)
Multi-Layer Protection	L3/L4 (UDP/TCP bandwidth exhaustion, SYN/ACK flooding, etc.) + L7 (HTTP/S flooding)
Resource-Level Protection	CloudFront, Elastic Load Balancing (ALB/NLB/CLB), Elastic IP (EC2), Amazon Route 53, AWS Global Accelerator, etc.
DRT (Dedicated Response Team) 24/7 Support	Real-time expert support and rule tuning during large-scale/persistent attacks
Cost Protection	Offset surging scaling/data transfer costs from large-scale DDoS attacks with credits
WAF·Firewall Manager Integration	Automatic rule recommendation/application, organization-wide policy deployment, Protection Group configuration
Visibility & Alerts	Alerts via attack vector/scale/duration metrics, CloudWatch/SNS, AWS Health events

[Table47] AWS Shield Incident Response Integration Plan

Stage	Role of AWS Shield
Preparation	Enable Shield Advanced on critical endpoints (CloudFront/ALB/EIP/Route 53/Global Accelerator) Pre-configure Shield Advanced, threshold alarms, contact networks, and DRT access permissions
Detection	Receive Shield events + AWS Health notifications during attacks; assess attack scale via metrics
Containment	Automatic mitigation (L3/L4) + L7 blocking via WAF rules, immediate DRT engagement when necessary
Eradication, Recovery	Gradually return to normal policies after confirming attack termination, traffic engineering/redirection
Post-Incident Response (Post-IR)	Review attack reports/metrics, tune WAF rules, request cost protection (claims), update playbooks

3.6. AWS Log Investigation

AWS provides logs across all layers—account, network, data, application, and security events—which are used complementarily throughout the incident response process. The log types provided by AWS are as follows.

1) Account and Management Activity Logs

These logs track overall management activities and resource configuration changes within an AWS account, primarily used for security audits, governance, and compliance purposes.

[Table48] List of Account and Management Activity Logs

Log Category	Description
AWS CloudTrail	• Records all API calls and management events occurring in the AWS account • Provides detailed records of who performed what actions, when, and where, offering complete visibility into account activity Complete visibility into account activity • Free access to the last 90 days of management event logs per account • Storing logs beyond 90 days or recording data events incurs charges
AWS Config	• Record and monitor the history of configuration changes to AWS resources • Evaluate whether resources comply with specific rules and track change history to maintain compliance status. • Provides configuration history for resources to help track activities performed during security events • This log is a paid service that records resource configuration items and charges for evaluating rules Charges apply
AWS Identity and IAM Access Analyzer	• Analyzes external access to resources to identify unintended public access • When resources like S3 buckets or IAM roles are shared with external accounts or users, It alerts you to potential security risks • This service is free and incurs no additional charges

2) Network and Traffic Logs

Used to record and analyze traffic flows within and outside the AWS network. In the AWS cloud, network activity can be recorded by creating a proxy that logs network traffic or by using traffic mirroring to send a copy of network traffic to a logging server. Essential for troubleshooting network performance issues, security audits, and analyzing traffic patterns.

[Table49] Network and Traffic Logs

Log Type	Description
VPC Flow Logs	• Detailed records of IP traffic passing through network interfaces in a Virtual Private Cloud (VPC) • Enables identification of traffic source, destination, port, protocol, and whether it was allowed or denied Useful for validating network security group and NACL (Network Access Control List) rules • These logs are a paid service that generates flow logs and incurs costs for storing them in S3 or CloudWatch Logs incurs charges
ELB Access Log	• Records information about all HTTP/HTTPS requests processed by the Elastic Load Balancer (ELB). • Provides detailed information such as client IP address, request time, ELB response time, HTTP status code, etc. Provides detailed information for analyzing application performance and debugging • These logs can be enabled for free, but storing the generated logs in an S3 bucket incurs S3 storage costs
AWS Global Accelerator Flow Log	• Records network traffic passing through Global Accelerator • Provides traffic-related information such as the user's geographic location, source and destination IP addresses, and protocol Helps analyze global network performance • This log is a paid service, incurring costs for generating logs and storing them in S3

3) Service-specific access/activity logs

Each AWS service generates its own activity logs, detailing usage patterns and access records for that service.

[Table50] Service-specific access/activity logs

Log Type	Description
Amazon S3 Server Access Log	• Records of all requests to Amazon S3 buckets • Identify who accessed objects in the bucket, when, and how to analyze bucket usage and troubleshoot security issues • This log can be enabled at no cost, but storage costs apply for the S3 bucket where logs are stored
Amazon RDS Logs	• Includes various database logs generated by RDS (Relational Database Service) instances including • Contains information necessary for analyzing and troubleshooting issues, such as MySQL's general query logs, error logs, and PostgreSQL logs Contains information necessary for analysis and troubleshooting • These logs can be enabled at no cost, but storing and analyzing them in CloudWatch Logs incur storage and analysis costs in CloudWatch Logs
Amazon CloudFront Access Log	• Records all user requests to CloudFront edge locations • Contains information such as request time, client IP, requested file, and HTTP status code and helps optimize cache hit rates • These logs can be enabled at no cost, but storage costs apply for the S3 bucket where logs are stored
Amazon API Gateway Access Log	• Records detailed information about API requests to API Gateway • Provides API call-related data such as requestor, response code, and latency to and diagnose issues • This log is a paid service, and logs are sent to CloudWatch Logs, incurring storage and analysis costs for CloudWatch Logs.

4) Security Service Logs

Generates logs that identify and record potential threats and security breaches. These logs are essential for continuously monitoring security posture and responding to threats.

[Table51] Security Service Logs

Log Category	Description
AWS GuardDuty Findings	• Detects and reports potential threats in your AWS environment • Identifies malicious activities such as abnormal access to EC2 instances and disabled port scans and records them as alerts (findings) • This service charges based on the volume of processed data and the number of generated detection results
AWS Security Hub	• Provides a unified view of security alerts and compliance status across your entire AWS account • Centralizes detection results from multiple security services like GuardDuty and Inspector Enables centralized management • This service incurs charges for collecting and analyzing detection results
AWS WAF Log	• The Web Application Firewall (WAF) monitors and logs web requests to web applications or APIs and logs them • and blocks web attacks such as SQL injection and cross-site scripting (XSS), logging detailed information • This log is a paid service, and costs are incurred for transmitting WAF logs to CloudWatch Logs or an S3 bucket incurs charges
Amazon Inspector Findings	• Scans resources like EC2 instances for vulnerabilities, identifying security weaknesses and best practice violations • Records these analysis results as 'findings' to help implement security enhancements • This service is paid; charges apply when Inspector scans resources and analyzes vulnerabilities

5) System and application logs

Refers to logs generated directly from computing resources such as EC2 instances, containers, and Lambda functions. Used to verify application behavior and system status.

[Table52] System and Application Logs

Log Classification	Description
Amazon CloudWatch Logs	• Collects, monitors, and stores logs generated by various AWS services and custom applications • Systematically manage logs through log streams and log groups, and use them to set up alarms or build dashboards • While collecting, storing, and analyzing logs incurs costs, a free tier is provided, allowing you to use up to a certain capacity (5GB) for free
Amazon EC2 System Log	• Includes operating system (OS) logs from EC2 instances, recording events and system messages during boot • Useful for diagnosing instance boot issues or system errors • The system logs of EC2 instances themselves are provided free of charge and can be viewed directly in the console
EKS/ECS Container Log	• Container logs generated by containers running on EKS (Elastic Kubernetes Service) and ECS (Elastic Container Service) Contains application logs and system logs • Includes application logs and system logs, essential for monitoring the operational status of containerized applications essential for monitoring the operational status of containerized applications • These logs incur storage and analysis costs when container logs are sent to CloudWatch Logs

3.7. AWS Incident Response Playbook Investigation

AWS addresses common incident scenarios customers may face, based on procedures from the NIST Computer Security Incident Response Guide (SP 800-61 Rev. 2). Playbooks are structured for reference at each stage: evidence collection, incident containment and removal, recovery from the incident, and post-incident activities.

This study selected 16 playbooks deemed particularly useful for incident response reference. Each playbook can be summarized by its DFIR perspective analysis points, key logs and data, and a summary of incident response procedures. The DFIR perspective analysis points were reconstructed and summarized by the research team after reviewing the playbooks. The playbooks for each incident type are as follows.

1) Unintended access to an Amazon Simple Storage Service (Amazon S3) bucket

This incident involved an Amazon Simple Storage Service (S3) bucket with misconfigured access policies, allowing unauthorized external users to access data.

S3 buckets are private by default, but during changes to bucket policies or ACLs for operational convenience or to meet service requirements, excessive permissions may be granted to IAM principals or anonymous users ("Principal": "*").

[Table53] DFIR Perspective Analysis Points - Unintended Access Incident in S3 Buckets

Category	Key Analysis Items
Root Cause Analysis	Public exposure due to incorrect S3 bucket policy changes (PutBucketPolicy)
Attack Traces	GetObject, ListBucket, and external IP calls within CloudTrail events
Impact Scope	Identification of exposed object lists, data types, and access log IPs
Enhanced Response	Continuous monitoring via IAM Access Analyzer, automated detection based on Config Rules, S3 policy change notification setup

[Table54] Key Logs and Data - Unintended access incidents to S3 buckets

Category	Data Source	Purpose of Acquisition
CloudTrail	Event Type: PutBucketPolicy, PutBucketAcl, PutObjectAcl, etc.	Identify the point in time when bucket policies and access permissions are changed
AWS Config	Resource state change history	Tracking policy modification history and access control changes
GuardDuty, Security Hub	Detection Alerts	Public Access or Abnormal Access Detection
IAM Access Analyzer	Identify IAM Policy Exposure	Detection of Misconfigured Trust Policies and Unauthorized Access

[Table55] Incident Response Procedure Summary - Unintended Access Incident in S3 Bucket

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Verify bucket policy, ACL, and IAM role change events in CloudTrail logs (PutBucketPolicy, PutBucketAcl) • Analyze the timing of changes and related principals using AWS Config and Detective • Secure abnormal access and API call history based on GuardDuty results • Preserve logs and configuration snapshots (S3 policies, IAM roles, etc.) in a separate evidence bucket

Incident Containment
(Containment)
	• Immediately block public access via the S3 console or CLI
	- Enable Block All Public Access
	- Remove Everyone and AuthenticateUsers permissions from ACL
	- Remove Principal: "*" and Get*/Put* actions from bucket policy
	• Block and reissue all potentially compromised credentials (Keys, Roles, STS sessions) in the IAM console
	• Apply IMDSv2 to EC2 instances to prevent attack propagation
	`aws ec2 modify-instance-metadata-options \`
	`--instance-id <ID> --http-tokens required --http-endpoint enabled`

Incident Eradication (Eradicate)	• Invalidate IAM role session: Modify Trust Policy and Detach Role • Restore S3 Bucket Policy to Least Privilege Principle • Enable S3 Versioning and MFA Delete • Apply S3 Server-Side Encryption (SSE-KMS) and Set Object Lock to Prevent Deletion/Modification
Recovery and Follow-Up Actions (Post-Incident Response)	• Restore damaged or deleted objects from backups/versions • Review backup policies and lifecycle settings • If public access is required, replace with pre-signed URLs • Review all IAM users and keys and reset to least privilege

2) Personal Data Breach

AWS Config rules, CloudTrail, GuardDuty, or external reports may detect such incidents. This is an incident where personally identifiable information (PII) in an AWS environment was exposed externally due to improper access controls, credential theft, or inappropriate actions by insiders.

PII can be exposed through various paths such as S3, DynamoDB, Lambda logs, and CloudWatch. Human error or configuration oversights, such as security perimeter breaches, misconfigured bucket policies, or application logging mistakes, are the primary causes.

[Table56] DFIR Perspective Analysis Points – Personal Information Leak Incident

Category	Key Analysis Items
Cause Analysis	External access permitted due to excessive IAM permissions and S3/DynamoDB configuration errors
Attack Traces	Detection of GetSecretValue, GetObject, and external IP addresses in CloudTrail events
Scope of Impact	Leaked data items (PII fields), accessing entities, API call timestamps
Enhanced Response	Re-establish IAM least privilege, enable automated detection via DLP/Macie, activate Secrets Manager auto-renewal

[Table57] Key Logs and Data – Personal Information Leak Incident

Category	Data Source	Purpose of Acquisition
CloudTrail	Event Type: GetObject, PutObject, ListBucket, GetParameter, GetSecretValue, etc.	Verify PII access and unauthorized download activities
VPC Flow Logs	Detect abnormal external traffic	Identify PII leakage paths
AWS Config	Resource Policy and Permission Change History	Track when IAM and resource settings change
GuardDuty, Security Hub	Detection Alerts	Detect Suspected Unauthorized Data Access and Leaks
DynamoDB, CloudWatch Logs	Detection of PII in Data	Verify PII records or log leaks
EBS Snapshot, Memory Dump	Original evidence for forensic analysis	Extraction of residual PII data from memory
IAM Access Analyzer	Detection of improper permission delegation and external entity access	Incident cause investigation and damage scope analysis

[Table58] Incident Response Procedure Summary – Personal Information Leak Incident

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Identify PII-related API call events in CloudTrail logs (`GetObject`, `GetParameter`, `GetSecretValue`) • Use IAM Access Analyzer to check external entity access permissions and bucket policy exposure • Cross-analyze AWS Config, VPC Flow Logs, and GuardDuty results together Identify PII access entities, timing, and paths • Identify PII contained in DynamoDB, CloudWatch logs, etc., then preserve log group snapshots

Incident Containment
(Containment)
	• Immediately block access to compromised accounts and roles via the IAM console or CLI
	`aws iam detach-user-policy --user-name CompromisedUser --policy-arn`
	`arn:aws:iam::aws:policy/AmazonS3FullAccess`
	• Block public access via the S3 console
	- Enable Block All Public Access
	- Remove Principal: "*" and excessive permissions from bucket policies
	• Disable vendor/external account access (DynamoDB/SQS)
	`aws iam detach-role-policy --role-name VendorRole --policy-arn`
	`arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess`
	`aws sqs remove-permission --queue-url <QueueURL> --label VendorAccess`
	• Manually delete logs containing PII within CloudWatch log groups or
	configure a shorter log expiration period

incident removal
(Eradicate)
	• Block STS sessions by invalidating IAM role sessions and modifying trust policies
	• Replace roles and key pairs associated with EC2 instances,
	Prevent metadata exposure by applying IMDSv2 (Instance Metadata Service v2)
	`aws ec2 modify-instance-metadata-options \`
	`--instance-id i-0123456789abcdef0 \`
	`--http-tokens required --http-endpoint enabled`
	• Encrypt or delete sensitive information fields in data processing paths such as DynamoDB and Lambda
	`aws dynamodb update-item \`
	`--table-name MyPIITable \`
	`--key '{"UserId": {"S": "123"}}' \`
	`--update-expression "SET pii_field = :val" \`
	`--expression-attribute-values '{":val":{"S":"[REDACTED]"}}'`
	• Re-examine Secrets Manager and Parameter Store and reissue keys and passwords

Recovery and Follow-Up Actions (Post-Incident Response)	• Perform recovery based on backups or version history after identifying the restore point for affected data • Notify data subjects and report to regulators regarding the scope of leaked PII (GDPR/Personal Information Protection Act Compliance) • Redesign IAM Role least privilege policies, implement PII encryption, and enhance automatic detection (Macie/DLP) to prevent recurrence and enhance automated detection (Macie/DLP)

3) Credential Leakage / Compromise

This incident involved the leakage and theft of AWS credentials such as Access Keys, STS tokens, and console credentials, which were exploited to create, modify, and access resources within the account. This incident can be detected through GuardDuty or Security Hub alerts, CloudTrail anomalies, resource creation outside operational regions, discovery of unregistered resources in the CMDB, or external reports. Due to the nature of credentials, they tend to be used covertly over extended periods. Therefore, the key is to block them immediately upon incident detection and reconstruct the timeline to take action.

[Table59] DFIR Perspective Analysis Points – Credential Leakage and Theft Incident

Category	Key Analysis Items
Cause Analysis	Credential exposure due to leaked Access Keys or STS tokens, excessive IAM permission settings, inadequate secret management, etc.
Attack Traces	Detection of AssumeRole, AttachPolicy, PassRole, privilege escalation, and new resource creation events in CloudTrail
Impact Scope	Identify resources and data accessible via compromised credentials (S3, EC2, IAM, RDS, etc.)
Enhanced Response	Apply access key rotation policies, enforce IMDSv2, re-establish IAM least privilege, enable Secrets Manager auto-rotation, and enhance detection monitoring

[Table60] Key Logs and Data – Credential Leak and Theft Incidents

Category	Data Source	Purpose of Acquisition
CloudTrail	Event Type: AssumeRole*, CreateUser/Role, Attach*Policy, Get*Token, RunInstances(PassRole), GetObject, etc.	All API timelines performed with leaked credentials Restore all API timelines
GuardDuty, Security Hub	CredentialAccess, UnauthorizedAccess, etc.	Initial incident index for leakage and misuse
AWS Config	Resource/policy change history	Tracking timing and subject of permission/configuration changes
VPC Flow Logs, WAF, ELB Log	External Communication/Abnormal Traffic	Identify C2, bulk communication, and data exfiltration paths
Detective, Athena	Relationship Graphs·Large-Scale Queries	Behavioral Correlation Analysis, Massive Log Exploration
EBS Snapshot, Memory (if needed)	Forensic Source	Volatile/Disk Evidence Preservation

[Table61] Incident Response Procedure Summary – Credential Leak and Theft Incident

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Perform a comprehensive API search (using Athena/search tools) for the entire CloudTrail period (covering at least the time around the suspected incident) based on the relevant Access Key and Principal ID • Cross-verify the timing and responsible party for changes to policies, roles, security groups, S3 policies, etc., using AWS Config • Use GuardDuty and Security Hub Findings to determine the incident detection time and event indexing time Use as a timeline key • Preserve logs and configuration snapshots in a separate evidence bucket (including hashes)
Incident Containment (Containment)	• Immediately block credentials - Disable/delete long-term keys (IAM User Access Keys), reset console passwords, enforce MFA - STS sessions remain valid until TTL expires → Detach role permissions/completely block policies, Block actual usage by modifying Trust Policy • Block high-risk paths - Remove S3 public/broad permission policies, close excessive inbound security group rules, enforce IMDSv2 • Immediately tag and clean up unauthorized regions and resources outside CMDB (Prioritize evidence preservation before immediate deletion)
Accident Removal (Eradicate)	• Identify all newly created users, roles, access keys, and profiles through CloudTrail reanalysis, then deactivate and delete them • Remove traces of privilege escalation (e.g., AttachPolicy, PassRole) and redefine least privilege • Eliminate vulnerabilities and error causes, and update pipeline secrets - Example causes: Key exposure, CI/CD secret leakage, public repositories, incorrect policies, etc. • Remove persistence (footholds) in logs/configuration/network aspects - Check whether schedulers, Lambda triggers, EventBridge rules, and access keys need to be regenerated
Recovery and follow-up actions (Post-Incident Response)	• Restore only essential permissions for legitimate users and workloads (Apply PBAC/ABAC or Permission Boundary) • Renew all secrets (Secrets Manager/Parameter Store/KMS key policies) and Redistribute referenced applications • Enforce key issuance/renewal/expiration policies, MFA/Conditional Access, private repository/secret scanners

4) Web Application DoS/DDoS Attack

This incident involves service availability degradation or disruption caused by attacks targeting web applications, such as massive requests (HTTP Flood) or network traffic generation. In AWS environments, built-in defense mechanisms like CloudFront, ALB, WAF, and Shield can be utilized for response. Attack patterns can be analyzed based on logs, and recovery is possible via Auto Scaling.

[Table62] DFIR Perspective Analysis Points – Web Application Denial of Service Attack

Category	Key Analysis Items
Cause Analysis	Abnormal surge in HTTP requests, external bot/script traffic, vulnerable WAF configuration, or lack of Auto Scaling setup
Attack Traces	Excessive requests in CloudFront/ALB logs, repeated identical IPs and User-Agents, requests concentrated in specific regions
Impact Scope	Reduced web application availability, service response delays, and outages
Enhanced Response	Strengthening AWS WAF/Web ACL policies, implementing Shield Advanced, and enhancing Auto Scaling and traffic distribution architecture

[Table63] Key Logs and Data – Web Application Denial-of-Service Attack

Category	Data Source	Purpose of Acquisition
CloudTrail, ALB Log	Request patterns, 4xx/5xx error rates, sudden spikes in request volume, etc.	Identify attack traffic types and concentrated intervals
Web Server Access Log	Analysis of abnormal requests, attacker IPs, and request parameters	Verify application-level attacks
CloudWatch Metrics	RequestCount, TargetResponseTime, ActiveConnection, etc.	Load metric tracking and auto-scaling decisions
AWS WAF, Shield Log	Blocked Requests, Rule Application History	Filter Policy Verification and Bypass Detection Analysis

[Table64] Incident Response Procedure Summary – Web Application Denial-of-Service Attack

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Collect CloudFront, ALB, and web server logs to identify abnormal request patterns (Multiple IPs, request surges, sudden spikes in 4xx/5xx error rates, etc.) • Analyze CloudWatch metrics to determine performance degradation causes and attack timing
Incident Containment (Containment)	• Distribute traffic via Auto Scaling, load balancers, and CloudFront • Security group reconfiguration (restrict public access, allow only load balancers) • Connecting WAF Web ACLs to CloudFront and ALB
Incident Eradication (Eradicate)	• Apply AWS WAF rules (AWS-Managed Rules) or custom rules • Strengthen filtering policies based on attack IPs, User-Agent, and URL patterns • Remove unnecessary resources and temporary configurations; configure CloudFront IP auto-update
Recovery and Follow-Up Actions (Post-Incident Response)	• Monitoring for return to normal traffic • Review activation of Shield Advanced and automation of DDoS response rules • Reorganize CloudWatch alarms and AWS Config rules

5) Dos/DDoS Attack

This is an attack scenario where web applications, load balancers, or network infrastructure in an AWS environment become unable to provide normal service due to excessive requests or traffic. It can manifest as traffic concentrated on a single instance (DoS) or simultaneous attacks from multiple distributed sources (DDoS). While AWS provides basic mitigation capabilities through Shield, WAF, CloudFront, Auto Scaling, etc., the core focus during incident response is analyzing the cause of the traffic, isolating it, and validating defense policies.

[Table65] DFIR Perspective Analysis Points – Denial of Service Attacks

Category	Key Analysis Items
Cause Analysis	Attacker generates massive requests/packets to exposed service endpoints to induce resource exhaustion
Attack Traces	GuardDuty DoS Findings, Flow Log showing flood from same IP, CloudWatch traffic spike
Impact Scope	Service delays, response failures (5xx errors), excessive CPU/network utilization, increased auto-scaling costs
Enhanced Response	WAF Rate-Based Rule, Shield Advanced subscription, Enhanced CloudFront caching, CloudWatch-based real-time alert configuration

[Table66] Key Logs and Data – Denial of Service Attacks

Category	Data Source	Purpose of Acquisition
AWS WAF, Shield Log	Attack Summary, Event Details	Identify attack timing, traffic type, and primary target resources
CloudWatch Metrics	Service-specific metrics (ALB, NLB, CloudFront, etc.)	Detect request volume, connection count, error rate, and abnormal traffic patterns
CloudTrail Log	API call logs	Identify resource changes, policy modifications, security setting deactivations, and other manipulations
VPC Flow Logs	Network traffic logs	Analyze external attack IPs, ports, and protocols to establish grounds for blocking
GuardDuty Findings	Automatic Detection Results	Detection of network-based anomalies such as DoS, spam bots, and abnormal port usage
AWS Config	Resource change tracking	Verify changes to security group, WAF rules, and load balancer settings

[Table67] Incident Response Procedure Summary – Denial of Service Attack

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Identify attack indicators - Verify spikes in CloudWatch metrics (RequestCount, ActiveConnectionCount, TargetResponseTime, HTTPCode_ELB_4XX_Count, RejectedConnectionCount, etc.) - Detect L7 (application layer) attacks using CloudFront metrics (Requests, TotalErrorRate) - Review GuardDuty Findings (e.g., Backdoor:EC2/DenialOfService.*, Behavior:EC2/TrafficVolumeUnusual) • Traffic source analysis - Identify repeated requests from identical IPs/ASNs and overseas IP ranges in VPC Flow Logs (Flow Log → Athena query to identify large-scale traffic sources) • Review WAF & Shield events - AWS Management Console → WAF & Shield → Global Threat Dashboard - Verify attack types, targeted resources, packet/request volumes • Preserve evidence - Download WAF & Shield logs - CloudWatch Metrics Snapshot - Flow Log backup (S3 Evidence Bucket)
Incident Containment (Containment)	• Block Attack Traffic - Create AWS WAF rules (based on attacker IP, User-Agent, URI) - Initially set to Count mode, then monitor via CloudWatch; switch to Block mode if normal requests are unaffected Switch to Block mode • Adjust Security Groups and NACLs - Block attack source IPs or port ranges - Exercise caution when adjusting NACLs to avoid affecting entire subnets • Auto Scaling group expansion - Manually adjust Auto Scaling policies to temporarily secure capacity - Guide traffic distribution to prevent service interruptions • Enable Shield Advanced (if applicable) - Monitor attack metrics via Console → WAF & Shield → Events (DDoSDetected, DDoSAttackRequestsPerSecond) - Request support from the AWS DDoS Response Team (DRT)
Accident Removal (Eradicate)	• Service Recovery - Verify CloudWatch metrics return to normal after attack traffic ceases - Clear CloudFront cache (Invalidations) and configure API Gateway Rate Limiting • Release unnecessary blocks - Modify excessive WAF blocking rules or incorrect security groups • Test normal traffic - Verify Route53 Health Check status (HealthCheckStatus) and Application Load Balancer Response (5xx Rate) Verification • Policy Review - Re-verify Shield and WAF detection settings, CloudFront origin access policies

Procedure Category	Action
Recovery and Follow-up Actions (Post-Incident Response)	• Restore normal infrastructure state - Redistribute standard security policies based on CloudFormation or Terraform - Reduce attack surface using CloudFront + WAF + API Gateway combination • Continuous Defense Enhancement - Configure rate-based rules, implement region-based filtering, utilize IP reputation lists - Set up CloudWatch alarm triggers (based on DDoSDetected and TargetResponseTime) • Improve operational response - Document the path from attack detection → mitigation → reporting - Conduct DDoS response drills and review emergency contact protocols

6) Public Resources Exposure - RDS

This incident involved a database exposed to the internet due to an RDS instance or snapshot with public access enabled.

[Table68] DFIR Perspective Analysis Points – Database Exposure

Category	Key Analysis Items
Root Cause Analysis	Public access setting changes, excessive Security Groups rules (0.0.0.0/0), snapshot exposure, IaC/script misconfiguration
Attack Traces	DB port scans/connection attempts from external IPs, increased abnormal login failures, mass Select/Export attempts
Impact Scope	Risk of data exposure, potential credential theft, operational disruption/performance degradation, compliance violation risk
Enhanced Response	Guardrails prohibiting public access, least privilege IAM, periodic password/token rotation, Config/WAF·GuardDuty integrated alerts, periodic Prowler checks

[Table69] Key Logs and Data – Database Exposure

Category	Data Source	Purpose of Acquisition
CloudTrail	AWS API call logs (ModifyDBInstance, ModifyDBSnapshotAttribute, ModifyDBClusterSnapshotAttribute)	When an RDS instance or snapshot becomes public, Identify the performer and credentials used (IAM user/role)
VPC Flow Logs	Record ENI traffic within the VPC	Identifying Attempts to Access Database Ports on the Internet (3306/5432/1433, etc.), External connections to public endpoints, unauthorized IP access, port scanning, and repeated access attempts
RDS Engine Log	Error/General/Audit Logs	Login failures, permission denials, mass query/export attempts, etc. Post-intrusion trace verification
CloudWatch Metrics	DB connection count, CPU·network I/O	Detecting service anomalies like traffic spikes and increased load
AWS Config	rds-instance-public-access-check, rds-snapshots-public-prohibited, etc.	Track configuration change history and public access violations, compare pre- and post-change states

[Table70] Incident Response Procedure Summary – Database Exposure

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Identify Scope of Impact - List public RDS instances/clusters and public snapshots • CloudTrail Query - Obtain the timing and subject of public configuration changes, along with relevant IAM permission usage history • Analyze VPC Flow Logs - Statistical analysis of external IPs, ASNs, countries, access ports, and frequencies • Collect RDS engine logs - Verify failed authentication, permission errors, bulk dump/scan patterns • Evidence preservation - Snapshots original logs and reports to evidence S3 bucket (version control and SSE-KMS applied)
Incident Containment (Containment)	• Immediately block public access - Disable RDS Public access, restrict endpoints to private subnets • Reduce network perimeter - Minimize security group inbound rules (VPN/Bastion dedicated CIDR), review NACLs • De-publish snapshots - Remove public/cross-account sharing attributes • Mitigate credential risks - Force password changes for DB users, rotate Secrets Manager/DB authentication tokens
Eradicate (Eradicate)	• Root cause elimination - Clean up misused IAM policies and roles (privilege reduction, inline policy checks), Correct change paths (console·CLI·IaC) • Clean up unnecessary resources - Delete unauthorized public snapshots, terminate public test instances • Close access paths - Review public routing/IGW paths, reorganize endpoint policies and route tables
Recovery and Follow-Up Measures (Post-Incident Response)	• Architecture Maintenance - Fixed RDS private subnet; use RDS Proxy and Private Link when necessary • Enhanced Monitoring - CloudTrail → Continuous analysis in S3/Athena, Config rule alerts, GuardDuty integration • Apply Baseline - Security group standard templates, Terraform/CloudFormation guardrails, change approval (Segregation of Duties) • Regular inspections - Prowler extra78 (public RDS), extra723 (public snapshots), group13 (RDS security) Periodic Execution

7) Public Resources Exposure – S3

This incident involves internal data exposure to the outside due to S3 buckets or objects with public access enabled. The primary causes are incorrect bucket policies, ACL settings, Public Access Block (PAB) disengagement, and configuration errors due to user mistakes.

[Table71] DFIR Perspective Analysis Points – S3 Bucket Exposure

Category	Key Analysis Items
Root Cause Analysis	Misconfiguration due to S3 bucket policy errors, ACL setting errors, deactivation of Public Access Block, or incorrect IaC deployment
Attack Traces	PutBucketAcl, PutObjectAcl, DeletePublicAccessBlock in CloudTrail, GET requests from external IPs, GuardDuty S3/ObjectRead.Unusual detection
Impact Scope	Sensitive data exposure, external download/replication possible, potential for privilege escalation and subsequent penetration
Enhanced Response	Enforce Block Public Access by default, automate Config rules, CloudWatch/EventBridge-based real-time alerts, Prowler regular scans, IAM least privilege design

[Table72] Key Logs and Data – S3 Bucket Exposure

Category	Data Source	Purpose of Retention
CloudTrail	PutBucketAcl, PutBucketPolicy, DeletePublicAccessBlock, PutObjectAcl, GetObjectAcl events	Identify when an S3 bucket or object became public, Identify the actor and reason for the change (IAM user, console/CLI, etc.)
S3 Server Access Log	Access Log, Server Log	Object access records from external IPs, HTTP request method (GET, PUT, DELETE) and response code analysis
VPC Flow Logs	ENI traffic records	Detection of abnormal external requests to S3 endpoints (In environments using S3 Gateway endpoints)
CloudWatch Metrics	GetRequests, 4xxErrors, 5xxErrors, FirstByteLatency	Identify excessive object requests, increased error rates, and abnormal traffic patterns
AWS Config	s3-bucket-level-public-access-prohibited, s3-bucket-public-read-prohibited, etc.	Detect bucket/object-level public access violations, tracking change history and status

[Table73] Incident Response Procedure Summary – S3 Bucket Exposure

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Extract public setting change events from CloudTrail - Performer, Access Path (IP, UserAgent), Console/CLI Distinction - DeletePublicAccessBlock, PutBucketAcl, PutBucketPolicy, PutObjectAcl • Analyze S3 server access logs - Identify GET request history from external IPs and successful access response records during the period of public exposure • Verify VPC Flow Logs and GuardDuty Findings - Detection of access from external C2 IPs or Tor nodes - Example: S3/MaliciousIPCaller, S3/ObjectRead.Unusual • Preserve all logs and analysis results in an S3 Evidence Bucket (with versioning and SSE-KMS applied)
Incident Containment (Containment)	• Immediately apply Block Public Access at account and bucket level aws s3api put-public-access-block --bucket <Bucket_Name> --public-access-block-configuration "BlockPublicAcls=true,IgnorePublicAcls=true, BlockPublicPolicy=true,RestrictPublicBuckets=true" • Remove public policies/ACLs - Console → S3 → Bucket → Permissions → Review and modify Bucket Policy / Access Control List • Review IAM policies - Immediately modify policies granting s3:* full permissions or containing Principal: * • If necessary, restrict bucket access to VPC endpoints in private subnets to enhance control
Eradicate (Eradicate)	• Clean up unauthorized objects and modifications - Identify and delete objects uploaded by attackers, scripts, and unauthorized files - Console → S3 → Bucket → Object List → Review versions and delete unnecessary ones • Disable Public Snapshots and Sharing - Check AccessControlList via AWS CLI or Console → Remove PublicRead or PublicReadWrite settings • Replace IAM Access Key·Role - If exposed keys or roles exist, immediately disable and reissue them
Recovery and Follow-Up Actions (Post-Incident Response)	• Restore deleted or corrupted objects - Restore the latest valid data from versioned and backup objects • Re-establish access controls - Apply least privilege policy, VPC Endpoint, and S3 prefix-based permission restrictions • Automated rule enforcement - Enable AWS Config rules (s3-bucket-level-public-access-prohibited, s3-public-read-prohibited) • Continuous Monitoring - Real-time detection of public access change events via CloudWatch EventBridge - Regular checks using Prowler (extra73 – detect public S3 buckets, extra769 – detect externally shared resources, group17 – Internet-exposed resource group scanning)

8) Code Exposure

This is a source code leak incident where code or configuration files from internal storage were replicated externally or made public. Such incidents are typically detected through DLP alerts, code posted on external platforms (e.g., GitHub, Pastebin), or abnormal bulk download requests in CloudTrail.

[Table74] DFIR Perspective Analysis Points – Source Code Leak

Category	Key Analysis Items
Root Cause Analysis	Credential exposure, public policy configuration errors, excessive permissions, misuse of external sharing links
Attack Traces	Massive `GetObject`, `BatchGetCommits` in CloudTrail, access from abnormal IPs or time zones
Impact Scope	Potential for follow-up attacks due to exposure of code, configuration information, credentials, environment variables, etc.
Enhanced Response	Full MFA implementation, S3 object event logging, secrets scanning, DLP rule refinement, Permission Boundary reinforcement

[Table75] Key Logs and Data – Source Code Leak

Category	Data Source	Purpose of Acquisition
CloudTrail	CodeCommit, S3, ECR, CodeBuild, etc. API calls	View code, download code, change policies, verify credential usage history
VPC Flow Logs	Traffic between resources and external IPs Traffic information	Identify large data transfers, abnormal ports, or countries
DNS Log	Internal DNS query records	Detection of data exfiltration channels like Pastebin, GitHub, etc.
CloudWatch	Service-specific logs and metrics	Identify abnormal traffic, errors, and increased throttling
CodeCommit or S3 Access Log	Repository access logs	Individual commit and object-level access timeline analysis

[Table76] Incident Response Procedure Summary – Source Code Leak

Procedure Category	Action Taken
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Review CloudTrail logs - Identify bulk code lookup or download events such as GetObject, BatchGetCommits, Download*, etc. - Verify API call originator, time, IP, and location information • Review CodeCommit or S3 access logs - Review bulk access logs within the last 24 to 72 hours - Verify S3 public access and whether GetObject requests originated from external IPs • Analyze VPC Flow Logs - Identify large-scale transfer patterns between code repositories or S3 and external IPs • Review DNS Logs - Verify requests to external domains such as GitHub, Pastebin, Discord • Review CloudWatch Metrics - Detect spikes in network outbound traffic, increased error rates, and abnormal request patterns • Identify relevant IAM users and credentials - Verify use of Access Keys, Role Sessions, and Federation Tokens
Incident Containment (Containment)	• Block access by unauthorized IAM users or roles • Disable compromised Access Keys and issue new ones • Revoke access permissions to CodeCommit repositories • Apply S3 public access blocking policy • Tag affected resources (repositories, buckets, code build environments) and mark them as "Quarantine" • Identify whether externally published code matches actual internal assets and secure evidence
Incident Eradication (Eradicate)	• Remove unauthorized commits, branches, or objects • Renew all exposed credentials and secret keys • Review and clean up related IAM roles, policies, access keys, and Secrets Manager entries • Remove exposure paths within automation pipelines or deployment scripts • Review API call history immediately before/after the leak using CloudTrail
Recovery and Follow-Up Actions (Post-Incident Response)	• Redistribute code based on the normal repository • Modify affected keys and environment variables • Re-examine security settings (Verify activation status of S3, CodeCommit, CloudTrail, VPC Flow Logs) • Enable MFA and reset passwords for developer accounts • Archive relevant logs and evidence (e.g., S3 Evidence Bucket)

9) Ransom Response for EC2

This incident involves an AWS EC2 instance infected with ransomware, resulting in detected data encryption, service disruption, or a monetary demand message. Attackers primarily infiltrate through stolen credentials (IAM Keys) or vulnerable remote access (SSH/RDP), potentially leading to EBS volume encryption, S3 data deletion, or further internal movement. Rapid isolation, evidence preservation, and identifying the recovery point are key response points.

[Table77] DFIR Perspective Analysis Points – EC2 Ransomware Infection

Category	Key Analysis Items
Root Cause Analysis	Initial infection via compromised IAM Key, vulnerable SSH access, or unapplied patches
Attack Traces	Deletion of EBS encryption and snapshots in CloudTrail, internal EC2 encryption processes, external C2 communication
Impact Scope	EC2 service disruption, EBS data encryption/loss, IAM credential exposure
Enhanced Response	Automated backup and offline storage, periodic IAM key rotation, mandatory IMDSv2, Continuous collection of VPC Flow Logs, EDR-based real-time infection detection

[Table78] Key Logs and Data – EC2 Ransomware Infection

Category	Data Source	Purpose of Acquisition
CloudTrail	EC2, EBS, IAM, KMS related API calls (RunInstances, CreateVolume, EncryptVolume, DeleteSnapshot, PutBucketLifecycle)	Instance creation by attackers, snapshot deletion, volume encryption, and other actions
CloudWatch Metrics	CPUUtilization, NetworkPacketsOut, DiskWriteOps	Data leakage, encryption process execution, Identifying abnormal resource load
VPC Flow Logs	Network traffic flow within the VPC	Verify communication with external C2 servers or attacker IPs Verify communication
AWS Config	ec2-instance-no-public-ip, ec2-volume-inuse-check, ebs-snapshot-public-restorable-check, etc.	Instance and volume security configuration status, Change history tracking
EBS Snapshot	Forensic evidence image	Infected files, encryption processes, ransom notes, etc. Internal evidence collection

[Table79] Incident Response Procedure Summary – EC2 Ransomware Infection

Procedure Category	Action

Acquire, Preserve, Document
(Acquire, Preserve, Document)
	• Identify event timing, actor, and API path via CloudTrail
	- Verify calls to `RunInstances`, `CreateTags`, `EncryptVolume`, `DeleteSnapshot`, etc.
	• Detach EBS volume to create snapshot
	`aws ec2 create-snapshot --volume-id <EBS_ID>`
	`--description "Forensic Snapshot before isolation"`
	• Review history of abnormal configuration changes (e.g., public IP assignment, IMDSv1 usage) in AWS Config
	• Preserve logs and snapshots in the S3 Evidence Bucket (with versioning + KMS encryption applied)

Incident Containment
(Containment)
	• Block infected instance from network
	- Create security group blocking all traffic
	- Remove default egress rules before connecting to instance
	`aws ec2 modify-instance-attribute --instance-id <INSTANCE_ID>`
	`--groups <ISOLATION_SG_ID>`
	• Detach if connected to an Auto Scaling group or ELB
	`aws autoscaling detach-instances --instance-ids <INSTANCE_ID>`
	`--auto-scaling-group-name <ASG_NAME>`
	`aws elb deregister-instances-from-load-balancer --instances <INSTANCE_ID>`
	`--load-balancer-name <ELB_NAME>`
	• Using EC2 Systems Manager (SSM) or EDR, capture volatile evidence such as memory/process lists After capturing volatile evidence, terminate

Incident Eradication
(Eradicate)
	• Clean infected instances and network
	- Apply NACL block rules based on attacker C2 IPs; remove malicious scripts, accounts, and schedulers
	• NACL Creation and Application Procedure
	- Amazon VPC Console → Network ACLs → Create → Inbound/Outbound Rules Edit
	- Enter IoC-based IP CIDR → Set to "DENY" and associate with Subnet
	• Identify and Delete Malicious IAM Users, Roles, and Access Keys
	`aws iam list-access-keys --user-name <User>`
	`aws iam update-access-key --user-name <User> --access-key-id <KeyID> --status Inactive`

Recovery and Follow-up Actions (Post-Incident Response)	• Data Recovery - Utilize CloudEndure Disaster Recovery → Select a recovery point prior to ransomware infection and restore workloads - Verify backup data is not encrypted before performing restoration • Instance Redeployment - Create new instances using trusted AMIs or backup EBS volumes - Renew IAM and KMS keys before terminating existing instances • Review and clean up IAM policies - Minimize root and operator privileges; discard temporary credentials • Long-term monitoring and prevention - Configure alerts based on CloudWatch anomalies (sudden network output spikes, CPU spikes) - Automate security compliance checks using Prowler and Security Hub

10) Ransom Response for RDS

This is an incident case where AWS RDS suffered data encryption, deletion, and unauthorized snapshot leakage attempts due to a ransomware attack. Attackers exploited stolen IAM keys, exposed RDS endpoints, and weak access controls (Security Groups, Public Access) to access the DB instance. They then encrypted data by issuing API calls to create, encrypt, and delete snapshots, demanding ransom. While RDS offers more automated backup and restore capabilities than EC2, early detection during the initial infection stage and verification of backup integrity are critical.

[Table80] DFIR Perspective Analysis Points – RDS Ransomware Infection

Category	Key Analysis Items
Root Cause Analysis	Exposed RDS endpoints, weak authentication, IAM key compromise, or vulnerable Security Group configuration
Attack Traces	`CreateDBSnapshot`, `StartExportTask`, `DeleteDBSnapshot` events in CloudTrail, Massive query/delete commands in RDS logs, External IP access
Impact Scope	Data encryption/deletion, unauthorized snapshot leakage, service unavailability
Enhanced Response	Enable Deletion Protection, implement RDS encryption and multi-AZ backups, enforce IAM least privilege policies, Continuous monitoring of AWS Config and Security Hub rules, Periodic verification of backup integrity

[Table81] Key Logs and Data – RDS Ransomware Infection

Category	Data Source	Purpose of Retention
CloudTrail	`CreateDBSnapshot`, `ModifyDBInstance`, `DeleteDBSnapshot`, `StartExportTask`, `ModifyDBClusterSnapshotAttribute`	Abnormal snapshot creation, deletion, or export activities, Attacker IAM identification
CloudWatch Metrics	CPUUtilization, FreeStorageSpace, NetworkPacketsOut	Execution of encryption processes, signs of data leakage, Detecting storage depletion
VPC Flow Logs	Network traffic flow within the VPC	Attempts to access RDS endpoints from external IPs, Identifying abnormal traffic such as repeated login failures
RDS Database Log	Error / General / Audit Log	Unauthorized user logins, bulk queries, Verify if Export commands were executed
AWS Config	`rds-logging-enabled`, `rds-storage-encrypted`, `rds-snapshots-public-prohibited`, etc.	Verify backup, encryption, public access settings, and other security configuration statuses and Change history verification

[Table82] Incident Response Procedure Summary – RDS Ransomware Infection

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Extract abnormal snapshot creation/deletion events from CloudTrail - Verify calls to CreateDBSnapshot, DeleteDBSnapshot, StartExportTask - Record the performer, IP, access channel (Console/API/CLI), and call time • Check compliance status in AWS Config resource timeline - Check for violations like rds-snapshots-public-prohibited, rds-storage-encrypted, etc. • Analyze unauthorized login attempts and SQL command execution via RDS Error/Audit Logs (ALTER/DROP/EXPORT) • Detect external access IPs and potential C2 servers using VPC Flow Logs • Back up and encrypt all evidence in an S3 Evidence Bucket (using KMS)
Incident Containment (Containment)	• Block RDS public access - Console → RDS → Select DB instance → Connectivity → Set Public access to "No" • Isolate security groups - Remove all inbound/outbound rules except for allowed IPs • Review IAM Policies - Remove excessive permissions (rds:*, *:*) held by roles/users related to RDS • Snapshot Export Stopped - Stop the StartExportTask task in progress aws rds cancel-export-task --export-task-identifier <task_id> • Verify AWS Config Rules are Enabled - rds-snapshots-public-prohibited, rds-instance-public-access-check
Eradicate (Eradicate)	• Remove unauthorized snapshots and unrecognized DB resources - Console → RDS → Snapshots → Manual snapshots → Select items to delete → Delete Snapshot • Blocking Based on Network IoCs - Modify VPC NACL → Add Inbound/Outbound DENY rules for attacker IP CIDR • Clean up IAM credentials - Renew potentially exposed IAM keys - Delete unauthorized IAM users, roles, and policies aws iam update-access-key --user-name <user> --access-key-id <key> --status Inactive • • Reactivate database deletion protection - Set deletion protection attributes for critical DB instances aws rds modify-db-instance --db-instance-identifier <db_id> --deletion-protection
Recovery and Follow-Up Actions (Post-IR)	• Verify backup data and restore - Using CloudEndure Disaster Recovery or automated backups (AWS Backup, RDS Point-in-Time Restore) - Verify snapshot integrity to prevent reintroduction of infected data during recovery • Deploy new RDS instance - Restore from a trusted backup, then deploy after reviewing security groups and IAM roles • - Renew and strengthen management of IAM keys and root keys - Perform full access key rotation • Automated security configuration checks - Enable AWS Config and Security Hub rules - Perform regular checks using Prowler (extra723 – Detect public snapshots, extra735 – Verify RDS encryption, group13 – Full RDS security check

11) Ransom Response for S3

This is an incident case where AWS S3 data was deleted, encrypted, or leaked externally via a ransomware attack. Attackers gain access through stolen IAM keys, exposed S3 buckets, disabled public access restrictions, or abuse of S3 API call permissions, primarily using APIs to encrypt or delete data.

[Table83] DFIR Perspective Analysis Points – S3 Ransomware Infection

Category	Key Analysis Items
Root Cause Analysis	Stolen IAM keys or policies with excessive permissions, MFA not enabled, S3 public access unblocked
Attack traces	CloudTrail DeleteObject, DeleteBucket, PutBucketEncryption, Mass DELETE/COPY requests in S3 Access Log
Impact Scope	Object deletion/encryption, data leakage, backup neutralization, and snapshot corruption
Enhanced Response	Enable S3 Object Lock and Versioning, implement MFA Delete, monitor Config and GuardDuty, Adherence to IAM Least Privilege Policies, Enable CloudTrail Data Event Logging

[Table84] Key Logs and Data – S3 Ransomware Infection

Category	Data Source	Purpose of Acquisition
CloudTrail	DeleteBucket, DeleteObject, PutBucketEncryption, PutObjectAcl, PutBucketPolicy, PutBucketReplication	Data deletion, encryption policy changes, bucket policy manipulation, etc. Identify Perpetrator and Timestamp
S3 Server Access Log	Requester, Remote IP, Request Type (REST.COPY.OBJECT, GET, DELETE)	Attempted data leakage such as bulk object deletion/copy or Identify destructive actions
CloudWatch Metrics	NumberOfObjects, BucketSizeBytes, 4xxErrors, 5xxErrors	Sudden decrease in object count, sharp drop in storage, Detect a sudden surge in error rates
AWS Config	s3-bucket-versioning-enabled, s3-bucket-default-lock-enabled, s3-bucket-level-public-access-prohibited, s3-bucket-server-side-encryption-enabled	Versioning, encryption, public access blocking, etc. Verify compliance status of security settings
IAM Access Analyzer	s3:* or external share policies (Principal: *) Inclusion status	Identify potential privilege abuse and external account sharing

[Table85] Incident Response Procedure Summary – S3 Ransomware Infection

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Identify unauthorized API calls in CloudTrail logs - Verify presence of DeleteBucket, DeleteObject, PutBucketEncryption calls - Extract caller, IP, UserAgent, access path (API/Console) • Identify consecutive REST.COPY.OBJECT and DELETE requests from the same remote IP and requester - Analyze for potential mass deletion or data exfiltration • Review AWS Config Rule Violations - s3-bucket-versioning-enabled, s3-bucket-public-write-prohibited • Retain logs and recovery backups in the S3 Evidence Bucket (Applied SSE-KMS encryption and versioning)

Procedure Category	Action
Incident Containment (Containment)	• Block public access and restrict policies - Immediately configure Block Public Access via the console or CLI aws s3api put-public-access-block --bucket <Bucket_Name> --public-access-block-configuration "BlockPublicAcls=true,IgnorePublicAcls=true, BlockPublicPolicy=true, RestrictPublicBuckets=true" • IAM Permission Restrictions - Identify and disable users/roles with excessive s3:* permissions - Restrict root user access and enforce MFA • Review Bucket Policies - Verify and remove combinations of Principal: * and Effect: Allow • Log Backup - Copy S3 server access logs and CloudTrail logs to a separate bucket
Incident Eradication (Eradicate)	• Remove unauthorized objects and changes - Console → S3 → Bucket → Objects → Verify and remove Delete Markers • Restore S3 encryption policy - Restore encryption settings (PutBucketEncryption) altered by attackers - Reapply normal SSE-KMS or SSE-S3 settings • Delete Unauthorized IAM Credentials and Roles - Clean up attacker-created accounts using `aws iam delete-user` or `delete-access-key` • Network-based Blocking - Identify and close external leakage paths associated with CloudFront, API Gateway, etc.
Recovery and Follow-up Actions (Post-Incident Response)	• Restore versioned objects - Restore normal data from previous versions in a bucket with versioning enabled aws s3api delete-object --bucket <Bucket_Name> --key <Object_Key> --version-id <DeleteMarkerID> • Restore a Deleted Bucket - Recreates with the same name and restores data from a Cross-Region Replication bucket or Restore data from the backup bucket • CloudEndure Disaster Recovery and backup restoration - Select a restore point prior to the ransomware incident - When using external backup solutions (Veritas, CommVault, etc.), verify backup integrity before restoring • • Enhance Security Policies - Enable MFA Delete and Object Lock (Compliance Mode) - Audit externally shared resources using IAM Access Analyzer - Enable automatic logging for new regions (SCP/Control Tower configuration)

12) Unauthorized Network Changes

This incident involves unauthorized or suspicious changes to network assets within an AWS account, such as security groups, NACLs, routing, and gateway ENIs. Incidents can be detected through unknown resource creation (EC2, LB, NAT, etc.), security group openings, routing changes, or sudden cost spikes.

[Table86] DFIR Perspective Analysis Points – Unauthorized Network Asset Modification Incident

Category	Key Analysis Items
Root Cause Analysis	Compromised/misused IAM credentials, overprivileged policies, non-compliance with change approvals, IaC pipeline malfunction
Attack Traces	Consecutive calls to modify Security Groups/NACLs/Routes/IGWs in CloudTrail, creating attack footholds via RunInstances, New external communications in Flow Logs
Impact Scope	Expanded external exposure, traffic bypass/interception, creation of data exfiltration paths, cost surge
Enhanced Response	Minimum privilege/role separation, CloudTrail/Config/FlowLog enabled across all regions at all times, EventBridge real-time detection, SSM auto-rollback, Change Management Board (CAB) review, tagging, CMDB consistency checks

[Table87] Key Logs and Data – Unauthorized Network Asset Modification Incident

Category	Data Source	Purpose of Acquisition
CloudTrail	Authorize/RevokeSecurityGroup*, Create/ModifyRoute*, Create/AttachInternetGateway, CreateNatGateway, RunInstances	Who/When/Where Network Control Plane Track who made changes to the network control plane, when, and where
VPC Flow Logs	ENI/VPC/Subnet level traffic records	Compare allowed/denied traffic flows before and after changes, Verify external communication
AWS Config	Security Group/NACL/RouteTable/IGW/ENI configuration history	Change timeline, previous state snapshots, Identify non-compliant resources
CloudWatch	NAT Gateway·ALB·EC2 Network Metrics	Detect operational impacts like traffic spikes/abnormal port usage
Network Manager, VPC Console	Topology and Route Analysis	Identify route changes, new transit points, and abnormal connection points

[Table88] Incident Response Procedure Summary – Unauthorized Network Asset Modification Incident

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Identifying Unauthorized API Calls in CloudTrail Logs - AuthorizeSecurityGroupIngress/Egress, Revoke*, Create/ModifyRoute*, AttachInternetGateway, RunInstances - Extract caller, sourceIPAddress, userAgent, and region • AWS Config Resource Timeline - Capture pre/post-change states of security group rules, route tables, and NACLs • Preserve VPC Flow Logs snapshots - Retain partitions for ±2 hours before/after change time, copy to S3 evidence bucket • Verify operational impact - Inspect NAT/ALB/EC2 network metrics (BytesOut, ActiveFlow, 4xx/5xx)
Incident Containment (Containment)	• Temporarily replace overly open security groups (bind to containment Security Group), Additional blocking via NACL DENY if necessary • Identify suspicious resource creators (CloudTrail RunInstances, etc.) → Restrict permissions for the relevant IAM user/role (detach policy or block trust policy) • When path pollution is suspected - Disable problematic route entries (temporarily switch to a bypass route)
Incident Eradication (Eradicate)	• Revert unauthorized changes - Restore Security Group/NACL/RouteTable/IGW Restore status • Clean up unauthorized resources - Remove unknown EC2/NAT/ENI/LB instances, inspect remaining Security Groups/Keys/Roles/tags • CloudTrail re-examination - Verify presence of additional privilege escalation (AssumeRole, Attach*Policy) and scaling traces
Recovery and Follow-up Actions (Post-Incident Response)	• Verify service paths and health checks are normalized (ALB/NLB/Route53) • Configure CloudWatch Alarms - NAT BytesOutToDestination, ALB RejectedConnectionCount, EC2 NetworkIn/Out, Flow Log-based Detection • Enhanced automatic detection/blocking - EventBridge → Authorize/RevokeSecurityGroup*, ModifyRoute* events Immediate Notifications - AWS Config rules (overly open security groups, public routes) + Automatic Remediation (SSM Automation) - Enable Security Hub FSBP

13) GuardDuty: PrivilegeEscalation-Kubernetes:PrivilegedContainer

This is an example of an incident detected by Amazon GuardDuty involving privilege escalation within an EKS cluster. The attacker could have executed containers with root privileges (privileged containers) using an IAM role or Kubernetes account with excessive administrative permissions, potentially leading to cluster control, data exfiltration, and attacks expanding into the internal network.

[Table89] DFIR Perspective Analysis Points – Privilege Escalation in EKS Clusters

Category	Key Analysis Items
Root Cause Analysis	Excessive ServiceAccount/IAM Role permissions, misused OIDC credentials, privileged container allowance settings
Attack Traces	Privileged Pod execution, abnormal RoleBinding/ClusterRole changes, GuardDuty detection events
Impact Scope	EKS cluster control takeover, internal network expansion, potential data exfiltration
Enhanced Response	Apply RBAC least privilege principle, block privileged mode, Enhanced Secrets renewal and monitoring, Continuous validation of GuardDuty and Config rules

[Table90] Key Logs and Data – Privilege Escalation within EKS Cluster

Category	Data Source	Purpose of Acquisition
GuardDuty Findings	GuardDuty Detection Event	PrivilegedContainer Detection Basis and Identifying the Executing Entity
CloudTrail Log	API call logs within the AWS account	EKS, IAM, STS related API calls and Tracking Credential Usage History
CloudWatch (EKS Audit Log)	EKS Control Plane audit logs	RoleBinding/ClusterRole changes, API calls, Service account activity identification
AWS Config	Resource change history and rule evaluation results	Verify whether cluster and IAM resource configurations have changed
Security Hub, Detective	Analyze GuardDuty integration detection results	Account and region-level threat correlation analysis
Forensic artifacts (EBS Snapshot, container logs, etc.)	Worker Node Disk, Runtime Logs	Files, processes, ports, and network status of compromised nodes/Pods Network Status, etc. Evidence Preservation

[Table91] Incident Response Procedure Summary – Privilege Escalation Within EKS Cluster

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Identify EKS clusters, pods, users, and nodes based on GuardDuty findings • Analyze API calls and permission change history using CloudTrail and EKS Audit Logs • Collect volatile data such as Pod container logs, process lists, and modified files • Create EBS snapshots for evidence preservation
Incident Containment (Containment)	• Block relevant Kubernetes users, renew IAM Role credentials • Isolate or suspend problematic Pods • Apply worker node cordon (block new scheduling) • Review ConfigMap and ServiceAccount mapping relationships
Incident Eradication (Eradicate)	• Analyze causes of Privileged Container creation (Incorrect RoleBinding, ServiceAccount permissions, etc.) • Delete or Privilege Reduction of Abnormal IAM Roles and ServiceAccounts • Verify and Clean Up New Resource Creation and Change History Based on CloudTrail

Procedure Category	Action
Recovery and Follow-Up Actions (Post-Incident Response)	• Reorganize Kubernetes RBAC policies • OIDC Credentials, Secrets Renewal • GuardDuty, Security Hub, Config Rule Re-evaluation • Kubernetes Audit Log Long-Term Retention Configuration

14) GuardDuty: Discovery – Kubernetes/SuccessfulAnonymousAccess

This is an instance where Amazon GuardDuty detected activity where API requests were successfully performed by an unauthorized user in a Kubernetes cluster. This detection indicates that the Anonymous Access setting on the Kubernetes API server is misconfigured, which could lead to a misconfiguration or credential compromise.

[Table92] DFIR Perspective Analysis Points – API requests by unauthorized users in a Kubernetes cluster

Category	Key Analysis Item
Root Cause Analysis	Kubernetes RBAC misconfiguration, system:anonymous permission maintained, API server authentication policy not applied
Attack Traces	API calls by anonymous user (system:anonymous), ClusterRoleBinding modification logs
Impact Scope	Cluster metadata leakage, exposure of Pod/service structure, potential for further intrusion propagation
Enhanced Response	Block anonymous access (anonymous-auth=false), automate RBAC verification, enhance GuardDuty and Audit Log monitoring

[Table93] Key logs and data – API requests from unauthorized users in Kubernetes clusters

Category	Data Source	Purpose of Acquisition
GuardDuty Findings	GuardDuty Detection Event	API events called by an anonymous user (system:anonymous) Identify API events called by anonymous users and determine the calling entity (IP·ASN·Region, etc.)
CloudTrail Log	API call logs within the AWS account	EKS-related API call history and IAM·STS activity tracking
CloudWatch (EKS Audit Log)	Kubernetes Control Plane Audit Log	Abnormal API calls, RoleBinding/ClusterRoleBinding changes, Verify anonymous user access history
AWS Config	Resource configuration change history	EKS Cluster, IAM Role, Security Group, etc. Detect changes and rule violations
Security Hub, Detective	Correlation analysis of GuardDuty integration results	Correlation analysis of detected anonymous API calls and other suspicious activities forensics artifacts (EBS Snapshot)
Forensic artifacts (EBS Snapshot)	EKS worker node evidence image	Preservation and Change History Verification of System-Level Evidence for Future Analysis Verification of Change History

[Table94] Incident Response Procedure Summary – API Request by Unauthorized User in Kubernetes Cluster

Procedure Category	Action
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Identify EKS cluster, Pod, User, Node based on GuardDuty Finding • Identify timing and resource types of system:anonymous API calls via CloudWatch (EKS Audit Log) • Review EKS and IAM-related API call history in CloudTrail logs • Preservation of relevant evidence data such as EBS snapshots
Containment (Containment)	• Review the necessity of using the system:anonymous user - Disable anonymous access if no operational requirements exist - Verification required before changes if impact on production workloads is anticipated • Review RBAC configuration - Use tools like rbac-lookup to verify permissions granted to the system:unauthenticated or system:anonymous groups - Remove unnecessary bindings (e.g., system:discovery, system:basic-user) • Share access status with cluster administrators and verify approval procedures

Procedure Category	Action
Incident Eradication (Eradicate)	• Analyze EKS-related API call logs from the past 90 days using CloudTrail - CreateUser, CreateRole, AssumeRole*, Attach*Policy, Get*Token - RunInstances (including PassRole) and new resource creation APIs - Traces of modifying or deleting existing resources • Identify root causes where anonymous access is enabled (API Server settings, RoleBinding, etc.) • Remove unnecessary ClusterRoles and RoleBindings • Renew IAM and ServiceAccount credentials
Recovery and follow-up actions (Post-Incident Response)	• Reapply RBAC least privilege principle and restrict system:anonymous usage • Review disabling the --anonymous-auth setting on the API Server • Enable long-term retention of CloudWatch-based EKS audit logs • Continuous validation of detection rules in GuardDuty, Config, and Security Hub

15) GuardDuty: GuardDuty: Impact – IAMUser/AnomalousBehavior

This is an instance where Amazon GuardDuty detected abnormal API calls by an IAM user or role (IAMUser/Role). This indicates that API patterns related to data tampering, deletion, operation, or disruption attempts were detected.

Typically detected when APIs like DeleteSecurityGroup, PutBucketPolicy, or UpdateUser are repeatedly called.

[Table95] DFIR Perspective Analysis Points – Anomalous API Calls by IAM Users or Roles

Category	Key Analysis Items
Root Cause Analysis	Credential compromise of IAM user or role, excessive permission granting, misuse of automation scripts
Attack Traces	Abnormal API calls (e.g., `DeleteSecurityGroup`, `PutBucketPolicy`), IAM policy modifications, traces of Access Key reissuance
Impact Scope	Resource manipulation within the account, service disruption, compromised data integrity
Enhanced Response	Re-evaluate IAM access controls, periodically renew Access Keys, integrate GuardDuty/CloudTrail real-time alerts

[Table96] Key Logs and Data – Abnormal API calls by IAM users or roles

Category	Data Source	Purpose of Acquisition
GuardDuty Findings	GuardDuty Detection Event	Basis for detecting abnormal API calls and Principal Identification
CloudTrail Log	API call logs within the AWS account	Full API call history and Analysis of the timing of abnormal activity
VPC Flow Logs	Network flow records	Analysis of external access IPs, ports, and traffic patterns
S3 Server Access Log	S3 Object Access Log	Verify Data Tampering and Deletion Requests
AWS Config	Resource configuration change history	IAM policies, S3 policies, security groups, etc. Track configuration changes
Security Hub, Detective	GuardDuty integration detection Correlation analysis	Identify Security Event Correlation Analysis

[Table97] Incident Response Procedure Summary – Abnormal API Calls by IAM Users or Roles

Procedure Category	Action Taken
Evidence Collection and Preservation (Acquire, Preserve, Document)	• Identify the type of abnormal API call and calling entities (Principal ID, User Name, ARN) • Obtain all API activity history for the relevant principal over the past 90 days via CloudTrail • Analyze IP, traffic, and data access traces using VPC Flow Logs and S3 Access Logs • Back up logs before and after the GuardDuty Finding occurred to S3 for evidence preservation

Incident Containment (Containment)	• Operational Impact Assessment
	- Verify whether the IAM user or role is in use for production workloads
	- Perform phased containment as immediate deactivation may cause service disruption
	• Privilege Documentation and Backup
	- Back up current permissions
	`aws iam list-attached-user-policies` and `get-user-policy` commands
	- Back up CloudTrail logs locally or to S3
	• Remove and Block Permissions
	- Deactivate access keys
	`aws iam update-access-key --status Inactive`
	- Detach IAM policies
	`aws iam detach-user-policy --user-name <user> --policy-arn <arn>`
	- For IAM Roles, track AssumeRole users via lookup-events and apply conditional blocking

Incident Eradication (Eradicate)	• CloudTrail Log Analysis (Athena-based) - Query all API activities by abnormal IAM entities over the past 90 days - Sensitive data access: S3 GetObject, DescribeInstances, etc. - New resource creation: EC2, Lambda, RDS, CloudFormation, Beanstalk - IAM resource operations: CreateUser, AssumeRole, Attach*Policy, Get*Token - Deletion and modification of existing resources: DeleteBucket, UpdatePolicy, UntagResource • Verify if additional credentials were created - Immediately block upon detecting the following image calls in CloudTrail events CreateAccessKey, CreateRole, GetFederationToken, etc. • Clean up abnormal resources - Disable or delete IAM users, roles, EC2 instances, etc. created by attackers - Restore modified policies and resources to their original state
Recovery and Follow-Up Actions (Post-Incident Response)	• Reapply the principle of least privilege for IAM users and roles • Strengthen access key rotation policies • Automated Long-Term Retention and Correlation Detection for CloudTrail, Config, and GuardDuty Logs • Add SNS notification rules for IAM policy change events

16) GuardDuty: Execution – EC2/MaliciousFile

This is an example of Amazon GuardDuty's Malware Protection scan detecting a malicious file within an EC2 instance. This detection indicates the instance is likely already compromised. The malicious file's path, volume ID, and infection trigger cause can be verified in the GuardDuty Finding details.

[Table98] DFIR Perspective Analysis Points – Malicious File Detection Within EC2 Instance

Category	Key Analysis Items
Cause Analysis	Malicious file uploaded to EC2 instance, exposure of vulnerable service, execution of infected AMI or S3 object
Attack Traces	GuardDuty MalwareFinding, suspicious API calls in CloudTrail (RunInstances, GetObject, PutUserData, etc.)
Impact Scope	Malware execution within instances, system resource abuse (CPU/network), potential for lateral movement
Enhanced Response	Automated EBS Snapshot backups, Enable Malware Protection real-time scanning, Minimize permissions for IAM Instance Profiles, Configure CloudWatch-based alerts for CPU/network anomaly detection

[Table99] Key Logs and Data – Malicious File Detection Within EC2 Instances

Category	Data Source	Purpose of Acquisition
GuardDuty Findings	Malware Protection Results	Identify instances, volume ARNs, and file paths where malicious files were detected
CloudTrail Log	EC2-related API records	Tracking creation/modification/snapshot events of infected instances
VPC Flow Logs	Network traffic logs	Detection of C2 (Command & Control) or external data transfer activities
EBS Snapshot (Forensic image)	EBS Evidence Replica	Malicious files, processes, logs Retention for forensic analysis
Security Group, NACL Settings Log	Security Rule Logging	Assessment of Instance External Exposure and Isolation Possibility
CloudWatch Metrics	CPU, Network, Disk I/O	Detection of abnormal process activity and identification of resource load patterns

[Table100] Incident Response Procedure Summary – Malicious File Detection Within EC2 Instance

Procedure Category	Action

Evidence Collection and Preservation
(Acquire, Preserve, Document)
	• Identify the following items in GuardDuty Findings
	- Instance ID, Volume ARN, malicious file path/name, Trigger Finding ID
	• Collect EC2 metadata and security group configurations
	`aws ec2 describe-instances`
	• Create EBS data volume snapshot
	`aws ec2 create-snapshot --volume-id <Volume_ID> --description "Forensic Snapshot of Infected Instance"`
	• Enable instance termination protection
	`aws ec2 modify-instance-attribute --instance-id <Instance_ID> --attribute disableApiTermination --value true`
	• Disable DeleteOnTermination and change termination action to Stop
	• Apply the "Quarantine" tag to indicate quarantine status

Incident Containment
(Containment)
	• Remove instances from Auto Scaling group and ELB
	`aws autoscaling detach-instances --instance-ids <Instance_ID> --auto-scaling-group-name <ASG_NAME>`
	`aws elb deregister-instances-from-load-balancer --instances <Instance_ID> --load-balancer-name <ELB_NAME>`
	• Disassociate IAM instance profiles
	`aws ec2 disassociate-iam-instance-profile --association-id <Association_ID>`
	• Network Isolation via Security Group Replacement
	`aws ec2 modify-instance-attribute --instance-id <Instance_ID> --groups <Isolation_SG_ID>`
	• Capture volatile data before forced instance shutdown (Shutdown) if necessary
	- Memory dump, network sessions, process list
	- Utilize host-based EDR agents or margaritashotgun

Incident Eradication (Eradicate)	• Malware removal - Cleanup via verified AV/EDR agents within infected instances - AWS Marketplace security solutions (TrendMicro, SentinelOne, etc.) can be utilized • Stop the instance if maintaining it is risky (Replace with a new instance and restore service) `aws ec2 stop-instances --instance-ids <Instance_ID>` • Check if the AMI is infected - If instances deployed from the same AMI exist, perform a full scan and rebuild them
Recovery and follow-up actions (Post-Incident Response)	• Restore services after redeploying healthy instances • Verify CloudTrail and GuardDuty integration notification rules • Improve Malware Protection scan cycles and infection response policies • Concurrent integrity checks of other instances and AMIs within the same region

4. Incident Data Collection

Incident response in cloud environments differs from on-premises environments because direct access to physical equipment or storage is not possible. Therefore, the process of securing digital evidence becomes critical. In other words, the rapid and systematic collection of data generated and held by cloud services is a core element of DFIR execution, representing a distinct difference from traditional collection procedures in on-premises environments.

[Table101] DFIR Data Collection Structure Framework in On-Premises vs. Cloud Environments

Category	On-Premises Environment	Cloud Environment
Active Data	Direct collection possible via physical access (Memory, sessions, processes, network connections, etc.)	Can be collected at the guest OS level (SSM, Agent, console-based remote commands)
Inactive Data	Disk images, event logs, configuration files, etc. Preserved on local storage media	CloudTrail, Config, S3, EBS Snapshot, etc. Automatically preserved in managed storage
Access Method	On-site physical equipment connection or local access	Access inside virtual instances (EC2) or Remote command execution via SSM
Limitations	Requires local administrator or physical device access privileges	No access to hypervisor or host level Access limited to guest OS and service API layers
Collection Tools	Volatile memory dump tools, network capture tools, automated selective collection scripts, etc. Non-volatile Disk imaging tools, automated selective log collection scripts, etc.	AWS CLI, Systems Manager (SSM), SDK commands, CloudTrail, AWS Config, CloudWatch, Athena, S3 Export, etc.
Data retention characteristics	Manual retention (extracted and stored by analysts)	Automatic retention (continuously managed by the service)
Collection limitations	Volatile data loss during equipment damage or power outage Possibility of log deletion, overwriting, or tampering Unable to collect if on-site access is restricted	Inability to access hypervisor memory Data loss when services are inactive or logs are not synchronized Uncollectable in certain managed service domains

The most critical purpose of data collection procedures in DFIR within cloud environments is to ensure incident replayability. Only when replayability is secured can the attack process be reconstructed on a timeline basis—revealing what privileges the attacker used to gain access, what resources were modified, and what logs were generated.

Therefore, this study categorizes collection targets into Command-based Data, Log-based Data, and Forensic Image to achieve reproducible data acquisition during incident response in the AWS cloud environment, and describes the collection procedures for each.

Chapter 4 covers the following content.

[Table102] Key Research Content – Incident Data Collection

Number	Subtitle	Key Content
1	Key logs frequently used in incident analysis	Defined log sources and collection/utilization procedures by component for conducting DFIR in AWS environments, selecting 7 core logs
2	Classification of collected data types and collection procedures	Classified data types in AWS incident response as command-based, log-based, and forensic images and propose step-by-step utilization objectives and reproducibility assurance measures

4.1. Key logs frequently used in incident analysis

To effectively perform DFIR in a cloud environment, you must first determine which logs to collect from the AWS environment. To do this, you must first understand the components of the deployed application (EC2, RDS, S3, etc.) and the layers of the cloud application stack (network, application, data layer). By clearly defining the types of log sources, their intended purposes, and the collection and storage procedures for each component, you can select the core logs to utilize for incident analysis. Key logs for use are as follows.

1) AWS CloudTrail Log

AWS CloudTrail is a service that supports governance, compliance, operational, and risk auditing for AWS accounts. All actions performed by users through the management console, SDKs, command-line tools, and other AWS services are logged as CloudTrail events. These logged events provide essential data for security analysis, tracking resource changes, and compliance audits.

CloudTrail is enabled by default, and only management events from the past 90 days are viewable. Therefore, to archive logs, you must create a Trail to store log files in an Amazon S3 bucket. CloudTrail logs are recorded in JSON format, and each event contains a field composed of multiple key-value pairs.

2) AWS VPC Flow Logs

AWS VPC Flow Logs capture IP traffic information passing through network interfaces within your Amazon Virtual Private Cloud (VPC). These logs provide detailed records of who communicated with whom, where, when, and through which ports, delivering essential data for analyzing network traffic patterns, detecting potential threats, and meeting security and compliance requirements. Flow Logs can be stored in Amazon S3 or CloudWatch Logs after creation, with additional costs incurred for log storage and transmission.

When stored in S3, VPC Flow Logs are recorded in plain text or Parquet format (a column-based data format using Gzip compression). When stored in CloudWatch, they are recorded in the CloudWatch service console.

3) Amazon S3 Server Access Log

Amazon S3 Server Access Log is a feature that records all request information for Amazon S3 buckets. This log provides detailed information about who, when, where, and how objects (files) in the S3 bucket were accessed. This enables analysis of access patterns to S3 buckets, serves as a key resource for security and compliance audits, and aids in incident analysis.

In AWS, object-level access logs for S3 buckets can be viewed in both S3 Server Access Log and CloudTrail S3 Data Event, but they differ in purpose and use cases. S3 Server Access Log is recorded as a text file format consisting of a list of space-delimited fields, with each log record containing information about a single S3 request.

[Table103] S3 Server Access Log vs. CloudTrail S3 Data Event

Category	S3 Server Access Log	CloudTrail S3 Data Event
Default State	• Disabled (requires separate configuration)	• Disabled (Requires enabling Data Event in Trail)
Log Storage Location	• Specified S3 bucket	• S3 bucket, CloudWatch Logs, CloudTrail Lake
Format	• Text (similar to Apache access log)	• JSON
Log Unit	• Request-based	• API call event-based
Included Information	• Requester, Request Timestamp (UTC), Request IP, Request type (GET/PUT, etc.), HTTP status code, Number of bytes transferred, User-Agent, etc.	• Event time, Event source, Event name (GetObject, PutObject, etc.), Requester, Request Parameters, Response Elements, Region, etc.
Logging time	• Possible logging delays	• Real-time logging
Purpose of use	• Tracking which IP accessed the object • Track whether downloads/uploads occurred • Detect DDoS/mass access attempts	• Verify API call activities • Detect misuse of IAM permissions • Tracking access based on specific users/roles
Cost	• S3 bucket storage costs for logs	• Charges apply when setting up Data Events (API calls in units of 100,000 + log storage costs)
Advantages	• Actual request flow and HTTP information (User-Agent, bytes transferred, etc.) can be verified	• CloudTrail standard event format Easy to analyze in conjunction with other AWS logs
Limitations	• Requires additional parsing since it is not in JSON format Additional parsing steps required • Cannot verify IAM/STS session information like CloudTrail	• Cannot view HTTP information (User-Agent, bytes transferred, etc.) • Billing burden exists

4) Amazon CloudWatch Logs

CloudWatch Logs is a feature that stores and manages log data collected by the Amazon CloudWatch service from AWS resources and applications. This feature is designed to centrally and comprehensively manage various log sources (such as EC2 instance system logs, Lambda execution logs, VPC Flow Logs, CloudTrail event logs, etc.). CloudWatch Logs enables the detection of abnormal activity among operational and security events.

Logs are recorded in a structured JSON-based format, containing information such as the occurrence time, log group, log stream, and message. It enables the identification of attack behaviors and log correlation analysis, making it a core data source for DFIR.

5) Amazon RDS Logs

Amazon RDS Logs are files containing records of various activities and events occurring on Amazon RDS DB instances. These logs play a critical role in monitoring the operational status of the database, troubleshooting performance issues, conducting security audits, and detecting potential security threats. Amazon RDS DB instances support MariaDB, Microsoft SQL Server, MySQL, Oracle, and PostgreSQL as database engines.

Amazon RDS Logs are categorized into database logs, which record events occurring within the DB engine (such as connections, query executions, and errors), and AWS DB instance events, which record changes related to the AWS DB instance. The structure of database logs varies depending on the database engine (MySQL, PostgreSQL, MariaDB, etc.) and the log type. The logs provide various types depending on the database engine, with representative logs including the following:

[Table104] Representative Amazon RDS Logs

Log Type	Description
Error Log (Error Log)	• Records diagnostic messages such as database start/stop times, errors, warnings, and notes
Slow Query Log (Slow Query Log)	• Records SQL queries that take a long time to execute, helping identify the cause of database performance degradation
General Query Log (General Query Log)	• Records client connections and disconnections, along with all executed SQL queries
Audit Log (Audit Log)	• A log for tracking access and activity on the database, including successful and failed logins, access to specific data, data modifications, and other activities

6) AWS GuardDuty Findings

AWS GuardDuty Findings are detailed security alerts generated when GuardDuty detects potential security threats within your AWS environment. GuardDuty analyzes various data sources (VPC Flow Logs, AWS CloudTrail event logs, DNS logs, etc.) using machine learning, anomaly detection, and integrated threat intelligence. If a threat is identified, it generates a Finding and notifies the user.

Each Finding contains rich information about the detected security issue, helping security personnel quickly understand the situation and take appropriate action. For example, a relevant Finding is generated if activities such as a specific EC2 instance communicating with a malicious IP address or an IAM user making API calls from an unusual location are detected.

GuardDuty Findings use a standardized JSON format and include various details essential for breach analysis.

7) AWS WAF Log

AWS WAF Log is a log that stores detailed information about all web requests processed by AWS WAF, the web application firewall provided by Amazon Web Services. It allows you to view detailed information for each request, including whether incoming HTTP/HTTPS requests to a website or web application were allowed (ALLOW) or blocked (BLOCK) based on AWS WAF rules. Using this log, security personnel can identify potential threats, analyze abnormal access attempts, and monitor whether configured security rules are functioning effectively.

AWS WAF Log is recorded in JSON format and contains various field information for each request. The log structure may vary depending on the WAF version.

4.2. Classification of Collected Data Types and Collection Procedures

In incident response within the AWS environment, collected data is categorized based on collection method and characteristics into Command-based Data, Log-based Data, and Forensic Image.

Each data type is utilized complementarily during the real-time response (Detection, Containment) and post-analysis (Analysis, Post-IR) phases. This procedural distinction clarifies the objectives for each incident response phase and ensures the reliability and reproducibility of collected data.

[Table105] Data Types by Collection Procedure

Collection Sequence	Classification	Purpose and Characteristics
1	Command-Based Data Collection	• Purpose - Secure real-time data such as instance state and configuration information at the time of an incident Preserving volatile information • Key Features - Captures volatile data that may be automatically modified or deleted over time - Captured via API command calls using AWS CLI, AWS SSM, etc. - Must be collected before logs (due to high likelihood of change)
2	Log-Based Data Collection	• Purpose - Analyze behavioral history before and after incidents • Key Features - Logs from CloudTrail, VPC Flow Logs, S3 Access Log, CloudWatch, etc., that are enabled - Lower volatility than command-based data, making it suitable for collection during the investigation phase
3	Forensic Image Collection	• Purpose - Secure data for in-depth incident analysis to determine root cause and ensure reproducibility • Key Features - Composed of relatively large-volume evidence such as disks, memory, and snapshots - For post-incident recovery and legal evidence preservation

Data collection methods by type are as follows.

1) Command-Based Data Collection

Command-based data refers to information retrieved in real-time via commands and APIs about the current state of running instances or services. In on-premises environments, this means system status data obtained by directly executing commands with administrator privileges. However, in AWS cloud environments, the structure shifts to remote collection via APIs, SSM, and console commands instead of local command execution.

[Table106] Overview of Command-based Data in AWS Environments

Category	Description
Definition	• Data queried or collected in real time by administrators or automation tools via APIs and commands during incident response • Information centered on operational state (State), such as internal instance status, processes, and network connections
Primary collection methods	• AWS Systems Manager (SSM) Run Command / Session Manager • EC2 Instance Connect CLI • AWS CLI / SDK • Remote diagnostic command execution via Lambda
Data Characteristics	• Content changes depending on collection time - Volatile • High real-time capability • Requires collection permissions • Focused on temporary data (Instance State)
Purpose of DFIR Utilization	• Identify internal instance activities and detect abnormal processes during an attack • Real-time assessment of breach scope and pre-isolation investigation
Example	• AWS SSM Run Command: Process list (ps), network connections (netstat) • EC2 describe-instances, describe-network-interfaces: Verify instance configuration status • AWS CLI get-console-screenshot: Capture instance screen status

Common methods for collecting command-based data include using AWS CLI, SSM, and Prowler. The collection methods and recommendations for each service are as follows.

AWS CLI

Using the AWS CLI, data can be collected categorized into: account and authentication management, instance and server status, network configuration and interfaces, storage-related data, log and monitoring settings, application/serverless configuration, key management/encryption, and other operational configurations.

[Table107] Command-based data collection methods using AWS CLI

Category (Primary Collection Target)	Purpose of Collection and Example Execution Command
Account and Authentication Related (IAM users, roles, policies)	• Account creation/deletion, permission changes, suspicious account verification - aws iam list-users - aws iam list-roles - aws iam list-policies - aws iam get-account-authorization-details
Account and Authentication Related (Active Access Keys)	• Verify old or unauthorized keys - aws iam list-access-keys --user-name <user>
Account and Authentication Related (MFA Configuration Status)	• Detect accounts without MFA - aws iam list-virtual-mfa-devices
Account and Authentication Related (Currently Authenticated User)	• Verify the IAM entity (user/role) for the current session - aws sts get-caller-identity
Instance and server status (EC2 instance list)	• Detect currently running instances and attacker-created instances - aws ec2 describe-instances
Instance and Server Status (Security Groups)	• Port openness, verify inbound/outbound rules - aws ec2 describe-security-groups
Instance and Server Status (Network ACL, Routing Table)	• Analyze network path manipulation - aws ec2 describe-network-acls - aws ec2 describe-route-tables
Instance and Server Status (Instance Metadata)	• Acquire IAM Role, AMI, IP, and region information - curl http://169.254.169.254/latest/meta-data/
Network Configuration and Interfaces (VPC Configuration)	• Verifying VPC Structure and Subnet Relationships - aws ec2 describe-vpcs
Network Configuration and Interfaces (ENI (Elastic Network Interface))	• Verify Associated IP, Security Groups, and Traffic Routing - aws ec2 describe-network-interfaces
Network Configuration and Interfaces (Elastic IP)	• Verify if the external IP used by the attacker - aws ec2 describe-addresses
Storage-related (EBS Volume)	• Identify disk configuration, size, and associated instances - aws ec2 describe-volumes
Storage-related (EBS Snapshot)	• Verify unauthorized creation/deletion of snapshots - aws ec2 describe-snapshots --owner-ids self
Storage-related (S3 Bucket List)	• Verify existence of sensitive data buckets - aws s3 ls
Storage-related (S3 Bucket Policy)	• Configure public access, verify bucket permissions - aws s3api get-bucket-policy --bucket <bucket>
Storage-related (S3 ACL Settings)	• Verify Unauthorized Access - aws s3api get-bucket-acl --bucket <bucket>
Logging and Monitoring Configuration (CloudTrail Configuration)	• Verify logging status and log destination bucket - aws cloudtrail describe-trails
Logging and Monitoring Configuration (VPC Flow Logs)	• Configure network traffic logging - aws ec2 describe-flow-logs

Category (Primary Collection Target)	Purpose of Collection and Example Execution Command
Logging and Monitoring Configuration (CloudWatch Log Groups)	• Log storage location and collection status - aws logs describe-log-groups
Log and Monitoring Configuration (Config Settings)	• Check for resource modification history - aws config describe-configuration-recorders
Application/Serverless Configuration (Lambda function)	• Detect attacker code injection or malicious Lambda - aws lambda list-functions
Application/Serverless Configuration (Lambda Environment Variables)	• Detect attacker code injection or malicious Lambda - aws lambda get-function-configuration --function-name <fn>
Application/Serverless Configuration (API Gateway)	• Creation of Abnormal API Endpoints - aws apigateway get-rest-apis
Key Management/Encryption (KSM Key)	• Encryption Key Management and Access Permissions Check - aws kms list-keys
Key Management/Encryption (Key Policy)	• Verify key access policy integrity - aws kms get-key-policy --key-id <id>
Other Operational Configuration (CloudFormation Stack)	• Verify whether the attacker used an automated deployment stack - aws cloudformation describe-stacks
Other operational configurations (ECS/EKS status)	• Detect container-based compromise - aws ecs list-clusters - aws eks list-clusters
Other Operational Configuration (Elastic Beanstalk/RDS)	• Application·DB Environment Configuration Tracking - aws elasticbeanstalk describe-environments - aws rds describe-db-instances

The following are recommended practices when collecting data via the AWS CLI.

[Table108] Recommended items when collecting via AWS CLI

Order	Details
1	• Always save AWS CLI output in JSON format - Example command: aws ec2 describe-instances --output json > ec2_status.json
2	• Ensure integrity - Example command: sha256sum ec2_status.json >> evidence_hash.log
3	• Upload to evidence storage bucket (set to Read-Only) - Example execution command: aws s3 cp ec2_status.json s3://dfir-evidence-bucket/

SSM (AWS Systems Manager)

To utilize SSM (AWS Systems Manager) commands for collection, certain prerequisites must be configured beforehand.

[Table109] AWS SSM Pre-Configuration Requirements

Number	Description
1	SSM Agent must be installed and running on the target instance
2	The IAM Role (Instance Profile) associated with the instance must have the AmazonSSMManagedInstanceCore permission
3	When uploading to S3, the instance profile must have s3:PutObject permission
4	Command executor requires ssm:SendCommand and ssm:GetCommandInvocation permissions

The SSM command execution structure can be categorized into three forms.

[Table110] AWS SSM Command Execution Structure

Category	Description
Actual Command (Script)	• Local commands executed on the instance to generate results - Example: Linux: ps awx, Windows: tasklist
SSM Document	• Execution templates provided by AWS or created by users - Example: AWS-RunshellScript (executes Linux Bash commands), AWS-RunPowerShellScript (execute Windows PowerShell)
AWS CLI call	• Execute actual commands on the instance - Example: aws ssm send-command --document-name "AWS-RunShellScript" --parameters commands=["..."] --instance-ids ... format

Data collection using SSM commands can be performed by calling an SSM document via the AWS CLI to remotely execute local commands within an instance.

An example of executing an SSM command is shown below, with the actual command running on the instance highlighted in red. Execution is possible by providing the SSM document type (--*document-name*) and the actual command (--*parameters commands*) as parameters, tailored to the operating system type.

[Table111] SSM Command Execution Example

Category	Example Execution Command
Collect instance metadata (Linux)	aws ssm send-command \\ --instance-ids i-0123456789abcdef0 \\ --document-name "AWS-RunShellScript" \\ --parameters commands=["curl -s http://169.254.169.254/latest/meta-data/ > /tmp/metadata.txt","sha256sum /tmp/metadata.txt > /tmp/metadata.txt.sha256"] \\ --output-s3-bucket-name my-evidence-bucket \\ --output-s3-key-prefix "incidents/IR-2025-10-12" \\ --comment "collect instance metadata"
Event Log (PowerShell) Collection (Windows)	aws ssm send-command \\ --instance-ids i-0123456789abcdef0 \\ --document-name "AWS-RunPowerShellScript" \\ --parameters commands=["Get-WinEvent -LogName Security -MaxEvents 200 | Export-Clixml -Path C:\\\\temp\\\\SecurityEvents.xml","Get-FileHash C:\\\\temp\\\\SecurityEvents.xml | Out-File C:\\\\temp\\\\SecurityEvents.hash"] \\ --output-s3-bucket-name my-evidence-bucket \\ --output-s3-key-prefix "incidents/IR-2025-10-12"

The result (file) is generated within the command itself, not by SSM, and can be configured to automatically save to S3.

[Table112] Verification and Extraction Methods for Results Executed via SSM Commands

Category	Description
SSM automatically saves stdout/stderr to S3 (using the '--output-s3-bucket-name' option)	When executing send-command, SSM generates stdout/stderr files to the specified S3 bucket (Recommended: Store in an evidence bucket and verify integrity)
View results using get-command-invocation	If S3 auto-saving is not used, next view the results of individual commands # Command execution example aws ssm get-command-invocation --command-id <command-id> --instance-id i-0123456789abcdef0 * command-id is the Command.CommandId value returned when executing send-command

Additionally, you can specify targets like '--targets "Key=tag:name,Values=webserver-*"' to execute commands on multiple instances simultaneously.

[Table113] SSM Command Execution Example

Category	Example Execution Command
Execute on multiple instances Execute	aws ssm send-command \\ --targets "Key=tag:Role,Values=web" \\ --document-name "AWS-RunShellScript" \\ --parameters commands=["ps aux > /tmp/ps.txt","sha256sum /tmp/ps.txt"] \\ --output-s3-bucket-name my-evidence-bucket

Using SSM commands, you can collect instance metadata, process lists, network connection history, authentication and access logs, and more. Additionally, you can apply the collection methods below to gather additional data needed for incident response.

[Table114] Command-based data collection methods using SSM commands

Category	Purpose of Collection and Example Execution Command
Instance Metadata (Linux)	• Instance ID, AMI, IAM Role, Local/Public IP, AZ, etc. Acquire metadata for instance identification # Linux --parameters commands=["curl -s http://169.254.169.254/latest/meta-data/ > /tmp/metadata.txt","sha256sum /tmp/metadata.txt"] # Windows --parameters commands=["Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/"]
Process list	• Collecting a list of running processes (identifying malicious processes and autorun programs) # Linux --parameters commands=["ps aux > /tmp/ps.txt","sha256sum /tmp/ps.txt"] # Windows --parameters commands=["tasklist > C:\\\\temp\\\\tasklist.txt", "Get-FileHash C:\\\\temp\\\\tasklist.txt"]
User/Session Login History	• Login Users, Remote Session History, Intrusion Path Tracking # Linux --parameters commands=["w; who; last -n 10 > /tmp/login.txt"] # Windows --parameters commands=["Get-EventLog -LogName Security -InstanceId 4624 -Newest 20 > C:\temp\logon.txt"]

Category	Purpose of Collection and Example Execution Command
System/Security Log (Recent)	• Capture system and security-related logs (logon failures, privilege escalations, etc.) # Linux --parameters commands=["tail -n 500 /var/log/auth.log > /tmp/auth_tail.log"] # Windows --parameters commands=["Get-WinEvent -LogName Security -MaxEvents 100 > C:\temp\Security.evtx"]
Disk mount information	• Check disk usage, mount status, and external volume connections # Linux --parameters commands=["df -h > /tmp/df.txt; lsblk > /tmp/lsblk.txt"] # Windows --parameters commands=["Get-Volume > C:\temp\volumes.txt"]
Container status (ECS/Docker)	• Identifying Container-Based Attacks # Linux --parameters commands=["docker ps -a > /tmp/docker_ps.txt; docker images > /tmp/docker_images.txt"]
Command History (Recent Activity)	• Identifying attacker activity traces through recently executed commands # Linux --parameters commands=["~/.bash_history > /tmp/history.txt"] # Windows --parameters commands=["Get-Content (Get-PSReadlineOption).HistorySavePath > C:\temp\history.txt"]
Task Scheduler List	• Automatic Execution and Persistence Maintenance Schedule Detection # Linux --parameters commands=["crontab -l > /tmp/cron.txt; ls /etc/cron* > /tmp/cron_dir.txt"] #Windows --parameters commands=["schtasks /query /fo LIST /v > C:\temp\schtasks.txt"]
Services, Drive List	• Detection of Malicious Services and Drives # Linux --parameters commands=["systemctl list-units --type=service > /tmp/services.txt"] #Windows --parameters commands=["Get-Service Select Name,Status,DisplayName > C:\temp\services.txt"]
Network Configuration Information	• Network interfaces, routing tables, DNS configuration # Linux --parameters commands=["systemctl list-units --type=service > /tmp/services.txt"] #Windows --parameters commands=["Get-Service Select Name,Status,DisplayName > C:\temp\services.txt"]
Network Connection Information	• External connections, listening ports, C2 connection verification # Linux --parameters commands=["ss -tunap > /tmp/netstat.txt"] # Windows --parameters commands=["netstat -ano > C:\temp\netstat.txt"]
Compress and upload results	• Batch compress collected data and store evidence in S3 # Linux --parameters commands=["tar zcvf /tmp/evidence.tar.gz /tmp/*.txt; aws s3 cp /tmp/evidence.tar.gz s3://my-evidence-bucket/"] #Windows --parameters commands=["Compress-Archive -Path C:\temp\* -DestinationPath C:\temp\evidence.zip; aws s3 cp C:\temp\evidence.zip s3://my-evidence-bucket/"]

Precautions when collecting via SSM commands are as follows.

[Table115] Recommended practices when collecting via SSM commands

Number	Description
1	Document Type (--document-name) Note (Linux → "AWS-RunShellScript", Windows → "AWS-RunPowerShellScript")
2	Utilizing S3 Auto-Save Options (--output-s3-bucket-name my-evidence-bucket --output-s3-key-prefix "incident/IR-2025-10-12")
3	Enable session logging (When using SSM Manager, log session logs to CloudWatch/S3)
4	Caution when creating an array of command strings as parameters In --parameters commands=["cmd1","cmd2"], commands must be written as a string array (Note: Use double quotes and shell escaping; Windows PowerShell commands must follow PowerShell syntax)

Prowler

Prowler does not execute shell commands inside instances like SSM does. Its strength lies in quickly scanning the configuration and settings state across the entire AWS account (IAM policies, MFA implementation, CloudTrail settings, S3 visibility, etc.) via API calls to generate a security status snapshot.

Prowler primarily checks configuration criteria and does not provide runtime evidence such as "who is currently logged into the session and what they are executing." Additionally, command-based data collected via Prowler is in the form of AWS API response results, necessitating supplementary data collection via SSM for completeness.

Prowler is highly valuable during the initial stages of DFIR (Digital Forensics and Incident Response) because it can rapidly gather the current state of AWS configurations and settings. Examples of collectable command-based data items include:

[Table116] Command-based data items collectable with Prowler

Category	Description
IAM	Users/Roles/Policies (broad permissions, root usage, outdated access keys, etc.)
CloudTrail	Trail activation status, log file placement location, management event logging settings
S3	Bucket visibility, versioning/encryption settings, ACL policies, etc.
VPC and Network	Flow log activation status, public subnet configuration, etc.
KMS	Key policies and external accessibility
Config	AWS Config activation status (whether resource changes are captured)
Other	EBS encryption, RDS security settings, excessive Lambda permissions, etc.

Each item in Prowler is internally defined by a check ID. It is recommended to specify the check ID when executing commands. The check ID and corresponding check behavior may vary depending on the Prowler version.

[Table117] How to check Prowler version and check list

Category	Command
Check Current Prowler Version	`prowler -v` or `prowler --version`
Check Available Checks	`prowler aws --list-checks`

The recommended workflow and configuration for collecting command-based data with Prowler is as follows.

[Table118] Recommended workflow when running Prowler

Order	Category	Description
1	Prepare Permissions	Grant read-only (minimum) permissions to the investigation role (or profile) Configure AssumeRole for multi-account investigations
2	Run Prowler (create snapshot)	Run Prowler in target account/region → Generate JSON/CSV output
3	Save Result Data	Upload the output file (e.g., prowler-results.json) to the evidence S3 bucket and Generate and store hash values
4	Verify and Filter	Identify 'High/Fail' items in Prowler results to prioritize response actions (e.g., public S3, inactive CloudTrail, etc.)
5	Collect via SSM integration	Based on suspicious points identified by Prowler, collect real-time status of the relevant instance via SSM commands Collect real-time status of the relevant instance
6	Reporting and Timeline Reconstruction	Analyze Prowler and SSM results together to determine root cause and identify indicators of compromise

[Table119] Recommended configuration when running Prowler

Number	Description
1	Output in -M json or -M csv format when running Prowler
2	Generate SHA256 hash for result file and store securely separately
3	Log Prowler execution logs (who ran it and when) to CloudTrail/CI logs
4	Convert High/Fail items detected by Prowler into automated tasks to invoke SSM action templates

Prowler enables assessment of overall account security posture, IAM security configuration checks, CloudTrail activation status, and S3 bucket access control reviews.

[Table120] Command-based data collection method using Prowler (v5.15.0)

Category	Collection Purpose and Example Execution Command
Entire Account Security Status Snapshot	• AWS account-wide configuration and security settings status check ./prowler aws --compliance cis_1.5_aws -M csv
IAM account and Permissions Configuration Check	• Identify account access vulnerabilities such as root account usage, lack of MFA, weak password policies ./prowler aws --checks iam_root_mfa_enabled iam_avoid_root_usage \ iam_user_mfa_enabled_console_access \ iam_password_policy_minimum_length_14 \ iam_password_policy_symbol \ iam_password_policy_lowercase \ iam_password_policy_uppercase \ iam_password_policy_number \ iam_password_policy_expires_passwords_within_90_days_or_less -M csv
Access Key and Credential Management	• Identify long-term unused keys and assess potential key exposure risks ./prowler aws --checks \ iam_rotate_access_key_90_days \ iam_user_accesskey_unused \ iam_user_two_active_access_key iam_no_root_access_key -M csv
CloudTrail logs Verify collection configuration	• Verify log collection status and integrity validation activation (CloudTrail active status, log integrity verification, CloudTrail log integration with CloudWatch) ./prowler aws --checks \ cloudtrail_multi_region_enabled \ cloudtrail_log_file_validation_enabled \ cloudtrail_cloudwatch_logging_enabled -M csv
CloudWatch Check Detection Configuration	• Verify security event detection and alarm configuration activation ./prowler aws –checks \ cloudwatch_log_group_not_publicly_accessible \ cloudwatch_log_group_kms_encryption_enabled \ cloudwatch_alarm_actions_enabled -M csv
S3 bucket Access Control Check	• Check for Data Exposure and Logging Enablement Status ./prowler aws --checks s3_bucket_public_access s3_bucket_object_versioning \ s3_bucket_default_encryption -M csv (Check S3 public access, Check S3 versioning status, Check S3 default encryption status)
GuardDuty Configuration Verification	• Verify GuardDuty service activation status ./prowler aws --checks guardduty_s3_protection_enabled -M csv
VPC Flow Logs, Security Group inspection	• Enable Flow Logs, Check Security Groups for Excessive Inbound Rules ./prowler aws --checks \ vpc_flow_logs_enabled \ ec2_securitygroup_allow_ingress_from_internet_to_any_port \ ec2_securitygroup_allow_ingress_from_internet_to_all_ports -M csv (Check if VPC Flow Logs are enabled, if excessive inbound ports (0.0.0.0/0) are exposed, if specific ports are exposed)
EC2 and EBS Access Configuration Check	• Instance encryption, metadata protection, external exposure verification ./prowler aws --checks \ ec2_instance_imdsv2_enabled ec2_launch_template_no_public_ip -M csv (IMDSv2 enabled status, public IP exposure status)

Category	Collection Purpose and Example Execution Command
KMS Key Management Check	• Verify KMS usage ./ prowler aws --checks kms_cmk_are_used -M csv
Verify RDS Security Configuration	• Check DB access control, storage encryption, backup functionality ./ prowler aws --checks rds_instance_no_public_access rds_instance_storage_encrypted -M csv (RDS public access configuration status, RDS storage encryption configuration status)
Check Lambda configuration	• Verify container execution environment isolation and encryption ./prowler aws --checks \\ awslambda_function_inside_vpc awslambda_function_not_publicly_accessible -M csv (Lambda function placement within VPC, Lambda public access blocked)
CloudFront and Route53 Configuration Check	• Verify HTTPS-only configuration ./prowler aws --checks cloudfront_distributions_custom_ssl_certificate -M csv
Save and preserve results	• Save the result JSON file to an evidence bucket (S3) and perform hash verification (integrity preservation) ./prowler -M json -r ap-northeast-2 -g all; aws s3 cp prowler-output.json \\ s3://dfir-evidence-bucket/IR-2025/prowler-output.json

2) Log-based Data Collection

Log-based data refers to event data automatically generated or stored by AWS. It logs API calls, traffic flows, access history, configuration changes, etc., and is utilized to reproduce attack behaviors or perform correlation analysis.

[Table121] Log-based Data in AWS Environments

Category	Description
Definition	• Behavior-based log data automatically generated and retained by AWS services • Account activity, network traffic, access history, configuration changes, etc. for long-term analysis
Primary Collection Methods	• Automatic log collection from CloudTrail, Config, VPC Flow Logs, S3 Access Log, WAF Log, etc. • Centralized storage via Security Lake or S3 • Collection also possible via management console or command-based methods
Data Characteristics	• Preservable - Non-Volatile • High automation and persistence • Service-specific format variations • Suitable for long-term timeline analysis • Some logs require administrator activation to be recorded
Purpose of DFIR Utilization	• Reconstructing attack paths and attack behavior timelines • Correlation analysis of intrusion traces and automated response • Post-incident audit and compliance verification
Example	• AWS CloudTrail: API call history • AWS Config: Resource Change Tracking • VPC Flow Logs: Network Flow • S3 Access Log: Object Access Log • GuardDuty Findings: Threat Detection Events

The primary methods for collecting log-based data involve gathering it through each console, service, or bucket. However, some logs require prior activation, so it is necessary to verify that they are enabled.

The collection methods and recommended practices for each log type are as follows.

[Table122] Log-based Data Collection Methods

Log Type	Collection Method
CloudTrail	• Verify Enablement - Path: AWS Management Console → CloudTrail Console → Trails - Can be stored in S3 buckets and CloudWatch Logs - Event history is automatically enabled, but only logs corresponding to 90 days are retained by default (To retain logs longer, you must explicitly create a Trail) • If no Trail exists - Path: AWS Management Console → CloudTrail Console → Event history • Stored in an S3 bucket - Log files are compressed in JSON format and stored in a date-based directory structure at the following path: (s3://<bucket-name>/AWSLogs/<account-id>/CloudTrail/<region>/<YYYY>/<MM> /<DD>/<filename>.json.gz) • Store in CloudWatch Logs - Path: CloudWatch Console → Logs → Locate the relevant group under Log groups - Export data to Amazon S3 via Actions → Export data to Amazon S3 to export logs to an S3 bucket - Download CloudWatch logs after exporting to S3
VPC Flow Logs	• Verify activation status - Path: AWS Management Console → VPC Console → Select VPC → Flow Log - Requires prior activation; can be stored in an S3 bucket and CloudWatch Logs • Store in S3 bucket - Log files are compressed in JSON format and stored in a date-based directory structure at the following path: (s3://<bucket-name>/AWSLogs/<account-id>/CloudTrail/<region>/<YYYY>/<MM> /<DD>/<filename>.json.gz) • Stored in CloudWatch Logs - Path: CloudWatch Console → Logs → Locate the relevant group under Log groups - Export data to Amazon S3 via Actions → Export data to Amazon S3 to export logs to an S3 bucket - Export VPC Flow Logs to S3 and download
S3 Server Access Log	• Verify activation status - Path: AWS Management Console → S3 Console → Select Bucket → Properties - Requires prior activation and can be stored in an S3 bucket - Verify the status (Enabled, Disabled) of the Server access logging item • Stored in S3 bucket - Stored as plain text separated by spaces in the following path (s3://<target-bucket>/<target-prefix>/<source-bucket-name> /YYYY-MM-DD-HH-MM-SS-<UniqueString>)
CloudWatch Logs	• Verify activation status - Path: AWS Management Console → CloudWatch Console → Logs → Log groups - Must be pre-activated; can be stored in an S3 bucket • Store in S3 Bucket - Logs can be exported to an S3 bucket via Actions → Export data to Amazon S3

Log Type	Collection Method
RDS Logs	• Verify activation status - Path: Amazon RDS Console → Databases → Select DB instance → Logs & events - Requires prior activation; logs can be collected via the RDS console, S3 bucket, or CloudWatch Logs - If inactive, neither CloudWatch groups nor log files exist • Collect Directly from the RDS Console - Path: Amazon RDS Console → Databases → Select DB instance → Logs & events → Select log file → Download • Save to S3 bucket - Save to the following path (s3://<bucket-name>/AWSLogs/<account-id>/rds/<db-instance>/...) • Save to CloudWatch Logs - Path: CloudWatch Console → Logs → Locate the relevant group under Log groups - Export logs to an S3 bucket via Actions → Export data to Amazon S3 - Export RDS Logs to S3 and download
GuardDuty Findings	• Check activation status - Path: AWS Management Console → AWS GuardDuty Console → Detectors - Requires pre-activation; findings can be collected from the GuardDuty console and S3 buckets - Findings do not exist unless a Detector is created • Collect directly from the GuardDuty console - Path: AWS Management Console → AWS GuardDuty Console → Findings (Download after reviewing detection events) • Stored in an S3 bucket - Path: AWS GuardDuty Console → Findings → Export → S3 Settings - Saves JSON files to the specified bucket at the following path (s3://my-guardduty-findings/AWSLogs/<account-id>/GuardDuty/<region>/YYYY/MM/DD/findings.json)
WAF Log	• Verify Enabled Status - Path: AWS WAF → Web ACLs → Select Web ACL → Enable logging • Stored in S3 Bucket - Log files are stored compressed (Gzip) in a date-based directory structure (s3://<bucket-name>/<prefix>/AWSLogs/<account-id>/AWSWAFLogs/<web-acl-name>/region/YYYY/MM/DD/HH/<file-name>.gz) • Store in CloudWatch Logs - Path: CloudWatch Console → Logs → Locate the relevant group under Log groups - Export data to Amazon S3 via Actions → Export data to Amazon S3 to export logs to an S3 bucket - Export WAF logs to S3 and download

EC2 Forensic Data Collection Procedure

[Table123] EC2 Forensic Data Collection Procedure List

Order	Category	Description
1	Isolate Suspected Compromised Instances	Immediately network-isolate EC2 instances suspected of compromise to prevent further damage propagation Isolation blocks attacker sessions and preserves the forensic target instance's state (Isolation reconfigures security groups without deleting or terminating the instance). • Apply the following security group blocking policies (Ingress: Allow only one analyst IP for SSH or RDP, Egress: Block all traffic (remove default allow all rule) • AWS CLI Command Examples aws ec2 create-security-group --group-name Quarantine-SG --description "Forensic quarantine" aws ec2 authorize-security-group-ingress --group-name Quarantine-SG --protocol tcp --port 22 --cidr <Analyst IP>/32 aws ec2 modify-instance-attribute --instance-id i-xxxx --groups <Quarantine-SG-ID>
2	Collect instance metadata	Collect basic information and environment components for the isolated instance • Items Collected - Instance ID, Type, AMI ID, Private/Public IP, VPC/Subnet/Security Group, Attached EBS Volume ID, Start time and region information • AWS CLI Command aws ec2 describe-instances --instance-ids i-xxxx \ --query 'Reservations[].Instances[].{InstanceId:InstanceId, ImageId:ImageId,PrivateIP:PrivateIpAddress,PublicIP:PublicIpAddress, SecurityGroups:SecurityGroups[*].GroupId, VpcId:VpcId,SubnetId:SubnetId,LaunchTime:LaunchTime}'
3	Instance Protection Settings	Apply protection settings to prevent evidence tampering • AWS CLI Command - Disable API Termination aws ec2 modify-instance-attribute --instance-id i-xxxx --disable-api-termination • AWS CLI Command - Prevent EBS Volume Deletion (Disable DeleteOnTermination) aws ec2 modify-instance-attribute --instance-id i-xxxx \ --block-device-mappings \ "[{\"DeviceName\":\"/dev/sda1\",\"Ebs\":{\"DeleteOnTermination\":false}}]"

Order	Category	Description

4	Create EBS Snapshot	Create an EBS snapshot to preserve the disk state
		(Snapshots correspond to disk imaging in on-premises environments, enabling evidence preservation without data loss)
		• Creation Procedure
		1. Stop the instance (to stabilize the volume state)
		2. Verify the attached volume ID
		- Select the damaged instance, navigate to the Storage tab to view the volume ID, or execute the AWS CLI command below
		`aws ec2 describe-instances --instance-ids i-xxxx \`
		`--query 'Reservations[].Instances[].BlockDeviceMappings[].Ebs.VolumeId'`
		3. Create a snapshot
		`aws ec2 create-snapshot --volume-id vol-xxxx \`
		`  --description "Forensic snapshot - i-xxxx" \`
		`  --tag-specifications 'ResourceType=snapshot,`
		`  Tags=[{Key=Forensic,Value=True}]'`

5	Preparing the Forensic Workstation	Set up a dedicated workstation (EC2) for evidence analysis within the forensic environment
		(Requires configuration within a dedicated investigation VPC, separate account, and based on a Golden AMI)
		• Recommended Settings
		- AMI: Use a pre-built Forensics Golden AMI
		- Network: Block external internet access, allow access to S3 Evidence Bucket
		- Security Group: Allow SSH/RDP only for investigator IPs
		- IAM Role: Grant only read-only S3 access and Snapshot replication permissions
		• AWS CLI Command Example
		` aws ec2 run-instances --image-id ami-xxxx \`
		`--instance-type m5.large \`
		`--subnet-id subnet-forensic \`
		`--security-group-ids sg-forensic \`
		`--iam-instance-profile Name=ForensicRole \`
		`--tag-specifications 'ResourceType=instance,Tags=[{Key=Purpose,Value=Forensic}]'`

Order	Category	Description
6	Create and attach forensic volume	Create a new EBS volume from the snapshot and attach it to the forensic workstation • Creation Procedure (Including AWS Command Examples) 1. Create a new EBS volume based on the compromised instance's EBS snapshot aws ec2 create-volume \\ --availability-zone ap-northeast-2a \\ --snapshot-id snap-0abcd1234efgh5678 \\ --tag-specifications 'ResourceType=volume,Tags=[{Key=Source,Value=ForensicSnapshot}]' 2. Attach the volume to a forensic workstation for evidence analysis aws ec2 attach-volume \\ --volume-id vol-0abc1234def5678gh \\ --instance-id i-0123456789abcdef0 \\ --device /dev/sdfaws ec2 attach-volume \\ 3-a. Volume recognition and read-only mount (Linux) – Includes command examples - Connect to instance (SSH) ssh -i key.pem ec2-user@<Forensic-EC2-IP> - Verify device recognition lsblk - Identify filesystem type sudo file -s /dev/xvdf - Mount as read-only sudo mkdir /mnt/evidence sudo mount -o ro /dev/xvdf1 /mnt/evidence - Verify Mount df -h | grep evidence 3-b. Volume Recognition and Read-Only Configuration (Windows) – Includes Command Examples - Connect to the forensic workstation via RDP - Run Disk Management (diskmgmt.msc) - Verify new disk recognition (typically displayed as Offline) - Change the disk to Online, but set it to "Read-Only" to prevent writing (The command below is a PowerShell command example) Get-Disk | Where-Object IsOffline -Eq $true | Set-Disk -IsOffline $false Set-Disk -Number <DiskNumber> -IsReadOnly $true - Once the disk drive is recognized, it can be accessed by forensic tools (Never click "Initialize Disk" in Disk Management)
7	Perform forensic analysis	Perform analysis using forensic tools on the mounted volume (Analysis must be performed on a copy, not the original volume)
8	Follow-up Actions	Upon completion of evidence collection, terminate the instance and preserve it as an AMI if necessary

EKS Forensic Data Collection Procedure

[Table124] List of EKS Forensic Data Collection Procedures

Order	Category	Description

1	Isolate Suspected Compromised Pods and Nodes	In EKS environments, compromises can occur at the Pod, Deployment, or node level. Unlike EC2, instances are not immediately blocked; instead, logical isolation is performed using Kubernetes control commands.
		• kubectl command - Pod isolation
		- Apply a label to the suspected compromised Pod
		`kubectl label pods -n <namespace> <pod-name> status=quarantine`
		- Apply network policy to block Pod log access
		`kubectl apply -f quarantine-networkpolicy.yaml`
		• kubectl command - Node Isolation (Cordon + Drain)
		- Prevents new Pods from being scheduled on the node
		`kubectl cordon <node-name>`
		- (Optional) Move running Pods to another node
		`kubectl drain <node-name> --ignore-daemonsets --delete-emptydir-data`
		• AWS CLI Commands - Block EC2 node (if necessary)
		- Identify the EC2 instance ID of the node, then use the AWS CLI to change the security group
		to a quarantine-only group
		`aws ec2 modify-instance-attribute --instance-id i-xxxx --groups sg-quarantine`

Order	Category	Description
2	Cluster and node metadata collection	Acquire structural information for Kubernetes and EKS tiers (This becomes core data for later analysis of where workloads ran) • AWS CLI Command - Cluster Metadata aws eks describe-cluster --name <cluster-name> --region ap-northeast-2 • kubectl commands - Node and instance mapping kubectl get nodes <node-name> -n <namespace> --show-labels \ -o custom-columns=NAME:.metadata.name,INSTANCEID:.spec.providerID - Extract only EC2 instance ID • kubectl get nodes <node-name> -n <namespace> --show-labels \ -o custom-columns=NAME:.metadata.name,INSTANCEID:.spec.providerID | sed -e 's/aws:.*\///g' - Map node names to EC2 instance IDs for integration with EC2 forensic procedures • kubectl commands - Pod / Deployment / Service mapping - Identify Pods associated with a specific Deployment kubectl get pods -l app=<deployment-name> - Identify Pods running with a specific container image kubectl get pods --all-namespaces -o json | jq -r --arg image "<image_name>" \ '.items[] | select(.spec.containers[] | .image == $image) | "\(.metadata.namespace) \(.metadata.name)"' - Identify Pods running with a specific service account kubectl get pods -A -o json | jq -r \ '.items[] | select(.spec.serviceAccount == "<service_account>") | "\(.metadata.namespace) \(.metadata.name)"' - Identify service IP kubectl get service [--all-namespaces, -n <namespace>] - Check Pods, Cluster IP, and Worker Nodes within a specific namespace kubectl get pods -n <namespace> --show-labels -o wide kubectl get pods -n <namespace> --show-labels -o json - View details of a specific Pod kubectl get pods <pod-name> -n <namespace> --show-labels -o wide kubectl get pods <pod-name> -n <namespace> -o=jsonpath='{.spec.nodeName}{"\n"}' • kubectl commands – Labeling affected resources kubectl label pod -n <namespace> <pod-name> status=compromised kubectl label node <node-name> status=quarantine
3	Preventing Worker Node Termination	Enable Termination Protection for the instance in the EC2 console

<table>
<tr><th>Order</th><th>Category</th><th>Description</th></tr>
<tr><td>4</td><td>Collecting Memory and Running State</td><td>Preserve running processes and container status for Pods or Nodes (Volatile data can be captured using container-level commands)
• docker command - container state preservation
- Checking Container Processes
docker top <container_id>
- Collect container logs
docker logs <container_id> > /tmp/<container_id>_logs.txt
- Collect container configuration and environment information
docker inspect <container_id> > /tmp/<container_id>_inspect.json
• Local command – Node memory dump (AVML)
- Perform the dump by accessing the node manually or using SSM (Deploying the SSM Agent Addon from Amazon EKS Addons enables secure remote command execution)
- Download and run AVML
sudo curl \
-LO https://github.com/microsoft/avml/releases/download/v0.3.0/avml
sudo chmod +x avml
sudo ./avml /mnt/forensic/memory.dmp</td></tr>
<tr><td>5</td><td>Create EBS Snapshot</td><td>Since EKS Worker Nodes are EC2 instances, secure disk evidence using EBS snapshots, similar to EC2 forensics.
• AWS CLI Command – Create EBS Snapshot for Node (EC2)
- Obtain EC2 instance ID
NODE_INSTANCE=$(kubectl get node <node-name> \
-o jsonpath='{.spec.providerID}' | sed 's|.*instance/||')
- Extract the attached EBS volume ID
aws ec2 describe-instances --instance-ids $NODE_INSTANCE \
--query 'Reservations[].Instances[].BlockDeviceMappings[].Ebs.VolumeId' --output text
- Create a snapshot
aws ec2 create-snapshot --volume-id vol-xxxx --description "EKS node forensic snapshot"</td></tr>
<tr><td>6</td><td>Kubernetes Audit and Pod Log Collection</td><td>Kubernetes API calls, Pod creation/deletion, Role changes, etc. can be verified in Audit logs (Audit logs must be enabled in CloudWatch Logs and can be collected directly via CLI)
• AWS CLI Command – Collect Kubernetes Audit Logs
aws logs filter-log-events \
--log-group-name "/aws/eks/<ClusterName>/cluster" \
--start-time "$START_MS" --end-time "$END_MS" \
--output json > eks_audit_logs.json
• kubetcl command – Collect Pod and Container Logs
- General Workloads
kubectl logs -n <namespace> <pod-name> --all-containers > pod_logs.txt
- System namespace (kube-system)
kubectl logs -n kube-system <pod-name> > kube_system_logs.txt</td></tr>
<tr><td>7</td><td>Evidence Preservation and Security Tagging</td><td>Collected artifacts (memory, snapshots, logs) are immediately uploaded to the S3 Evidence bucket, their integrity verified, and included in the investigation record
• AWS CLI Command – Generate Hash and Upload to S3
- Generate hash (locally)
sha256sum memory.dmp > memory.hash
- S3 Upload Example
aws s3 cp memory.dmp s3://forensic-evidence-bucket/EKS/memory.dmp \
--sse aws:kms
aws s3 cp eks_audit_logs.json s3://forensic-evidence-bucket/EKS/audit/</td></tr>
</table>

5. Incident Analysis Techniques

The core objective of the DFIR analysis phase in a cloud environment is to identify attack activities based on collected data and reconstruct the sequence of events in a timeline format. Through accurate log interpretation and service-specific behavioral correlation analysis, it is possible to determine the attacker's intrusion path, privilege escalation, data manipulation, and potential data exfiltration. This serves as the basis for developing security enhancement measures and recurrence prevention actions.

This study developed a DFIR CheatSheet to systematize incident analysis in AWS environments by defining key analysis fields per log and service, identifying events frequently observed in attacks, and mapping major log events to attack tactics.

Furthermore, to enhance analyst efficiency during incident analysis in AWS environments, we implemented an analysis tool enabling attack-signature-based log event analysis and tactic-based event visualization from CloudTrail, VPC Flow Logs, and S3 Access Logs. This tool detects log patterns frequently observed in real attacks, presenting analysts with priority analysis points.

Chapter 5 covers the following content.

[Table125] Key Research Content – Incident Analysis Techniques

Number	Subtitle	Key Content
1	Key Analysis Fields and Event Analysis per Log Type	CloudTrail, VPC Flow Logs, S3 Access Log, CloudWatch Logs, etc. Key fields and event analysis critically utilized from a DFIR perspective for each major AWS log type
2	Log Event Mapping by Attack Tactics DFIR Cheat Sheet Development	Mapping and classifying AWS logs based on MITRE ATT&CK tactics Presenting a CheatSheet that organizes key events/operations per tactic and DFIR analysis criteria
3	Development of AWS DFIR Log Analysis Tool	Developed AWS DFIR tool (bitParser) that automatically analyzes CloudTrail, VPC Flow, and S3 logs Provides tactic-based log analysis and key detection points

5.1. Key analysis fields and event analysis per log type

Analyzed key fields and events from a DFIR perspective for major AWS log types such as CloudTrail, VPC Flow Logs, S3 Access Log, and CloudWatch Logs.

1) CloudTrail

CloudTrail logs are recorded in JSON format, with each event containing fields composed of multiple 'key-value' pairs. Key fields utilized for incident analysis are as follows.

[Table126] CloudTrail Log Example

Partial Log Example
{"Records": [{ "eventVersion": "1.08", "userIdentity": { "type": "IAMUser", "principalId": "AIDA6ON6E4XEGITEXAMPLE", "arn": "arn:aws:iam::888888888888:user/Mary", "accountId": "888888888888", "accessKeyId": "AKIAIOSFODNN7EXAMPLE", "userName": "Mary", "sessionContext": { "sessionIssuer": {}, "webIdFederationData": {}, "attributes": { "creationDate": "2023-07-19T21:11:57Z", "mfaAuthenticated": "false" } } }, - Omitted below -

[Table127] CloudTrail log fields primarily used for incident analysis

Field Category	Description
eventVersion	Version of the CloudTrail event structure
userIdentity	Information about the entity that performed the request - Identifies the target IAM user, role, AWS service, etc. - Clearly identifies the actor through type (Root, IAMUser, AssumedRole, etc.) and arn (Amazon Resource Name) - Event time
eventTime	Event occurrence time (UTC)
eventName	Name of the event performed
awsRegion	AWS region where the event occurred
sourceIPAddress	IP address from which the API call originated - An important clue for determining attacker IP or internal IP
userAgent	Information about the client that sent the request (e.g., AWS CLI, SDK, web console)
requestParameters	Parameters used in the API call - Allows verification of which resource was targeted and what values were used in the request
responseElements	Response values from the API call - Includes success status and created resource information
errorMessage	Displays an error message if the API call fails - Enables identification of the cause of the request failure

The primary events logged by CloudTrail can be categorized into administrative events, data events, and insight events. CloudTrail events corresponding to MITRE ATT&CK tactics are as follows.

[Table128] Key CloudTrail Events by MITRE ATT&CK Tactics

Event Type	Description
Initial Access (Initial Penetration)	Provides visibility into the attacker's initial penetration into the system (Detects actions to access AWS accounts or gain user privileges) • ConsoleLogin: Logging into the AWS Management Console • PasswordRecoveryRequested: Password recovery request • AssumeRoleWithWebIdentity: Assume a role using web credentials to obtain temporary security credentials • GetSessionToken: Requesting a temporary session token for the AWS API
Execution (Execution)	Provides visibility into malicious code execution within compromised environments (Detect actions to launch computing resources or execute commands within the AWS environment) • StartInstance: Starting stopped EC2 instances • StartInstances: Starting multiple stopped EC2 instances • Invoke: Call an AWS Lambda function • SendCommand: Sends a command to an EC2 instance
Persistence (Persistence)	Provides visibility into persistence actions where attackers attempt to maintain access after credential changes (Detect actions to create backdoors or gain persistent access within AWS accounts) • CreateAccessKey: Generating an access key for an AWS user or role • CreateUser: Create a new IAM user • CreateNetworkAclEntry: Creates a network access path by adding a NACL entry • CreateRoute: Create a network access path by adding an entry to the routing table • CreateLoginProfile: Create a login profile for an IAM user • AuthorizeSecurityGroupEgress: Modify the outbound rules of a security group • AuthorizeSecurityGroupIngress: Modify the inbound rules of a security group • CreateVirtualMFADevice: Create a virtual MFA device • CreateConnection: Create a Direct Connect connection • ApplySecurityGroupsToLoadBalancer: Apply security groups to a load balancer • SetSecurityGroups: Set security groups on a load balancer • AuthorizeDBSecurityGroupIngress: Allow inbound rules for RDS database security group • CreateDBSecurityGroup: Create an RDS database security group • ChangePassword: Change User Password
Privilege Escalation (Privilege Escalation)	Provides visibility into attempts by attackers to elevate privileges from lower to higher levels (Detect actions that modify IAM permissions to gain access to more AWS resources) • CreateGroup: Create an IAM group • CreateRole: Create an IAM role • UpdateAccessKey: Update an existing access key • PutGroupPolicy: Add and modify inline policies for a group • PutRolePolicy: Add and modify inline policies for a role • PutUserPolicy: Add and modify inline policies for a user • AddRoleToInstanceProfile: Add a role to a profile or group • AddUserToGroup: Add a user to a profile or group • AttachUserPolicy: Attach an IAM managed policy to a user • AttachRolePolicy: Attach an IAM managed policy to a role

Event Type	Description
Defense Evasion (Defense Evasion)	Provides visibility into attacker actions aimed at disabling detection and defense systems (Detecting actions such as stopping CloudTrail logging or deleting/modifying security solution configurations) • StopLogging: Disables CloudTrail logging • DeleteTrail: Deleting CloudTrail trails • UpdateTrail: Updates CloudTrail trail configuration • PutEventSelectors: Modify event selectors for a trail • DeleteFlowLogs: Deleting VPC Flow Logs • DeleteDetector: Delete GuardDuty Detector • DeleteMembers: Delete GuardDuty member accounts • DeleteSnapshot: Delete EBS or RDS Snapshot • DeactivateMFADevice: Deactivate MFA device for user account • DeleteCertificate: Delete SSL/TLS Certificate • DeleteConfigRule: Delete AWS Config rule • DeleteAccessKey: Delete access key • LeaveOrganization: Leave an AWS Organization • DisassociateFromMasterAccount: Disassociate account from GuardDuty master account • DisassociateMembers: Disassociate account from GuardDuty member • StopMonitoringMembers: Stop monitoring GuardDuty member accounts
Credential Access (Credential Access)	Provides visibility into attacker attempts to compromise credentials (Detect password or security credential lookup, creation, or modification attempts) • GetSecretValue: View secret values stored in AWS Secrets Manager • PutSecretValue: Modify secret values stored in AWS Secrets Manager • GetPasswordData: Retrieve administrator passwords for EC2 instances • RequestCertificate: Request a certificate from AWS Certificate Manager • UpdateAssumeRolePolicy: Update the trust policy for a role • CreateSecret: Create a secret in Secrets Manager • DeleteSecret: Delete a secret from Secrets Manager
Discovery (Exploration)	Providing visibility into an attacker's attempts to discover the system and network environment (Detect attempts to list resources, users, permissions, etc., within the AWS environment and gather information) • ListUsers: List IAM users • ListRoles: List IAM roles • ListIdentities: List IAM identities • ListAccessKeys: List IAM user access keys • ListServiceQuotas: List AWS service quotas • ListInstanceProfiles: List EC2 instance profiles • ListBuckets: List S3 buckets • ListGroups: List IAM groups • GetSendQuota: Check Simple Email Service (SES) send quota • GetCallerIdentity: Verify the credentials of the current user • DescribeInstances: View details about EC2 instances • GetBucketAcl: Check ACL for S3 bucket • GetBucketVersioning: Check the versioning status of an S3 bucket • GetAccountAuthorizationDetails: View detailed permission information for IAM entities (users, groups, roles, etc.) in an AWS account Verify detailed permission information for IAM Entities (users, groups, roles, etc.) in an AWS account

Event Type	Description
Lateral Movement (Internal Movement)	Provides visibility into an attacker's attempts to move to other systems within the network (Detect attempts to move within the AWS environment by switching from one role to another) • AssumeRole: Temporarily grants permissions from the current role to another role • SwitchRole: Temporarily grants permissions from the current role to another role
Exfiltration (Data leakage)	Provides visibility into an attacker's attempt to exfiltrate data from the current environment to an external location (Detect attempts to download data from S3 buckets or share snapshots to exfiltrate data externally) • GetObject: Verify objects in an S3 bucket • CopyObject: Copying objects from an S3 bucket • CreateSnapShot: Creating an EBS snapshot to share externally • ModifySnapshotAttributes: Modify EBS snapshot attributes to share externally • ModifyImageAttribute: Modify AMI (Amazon Machine Image) attributes to share externally • SharedSnapshotCopyInitiated: Shared snapshot copy • SharedSnapshotVolumeCreated: Volume created for shared snapshot • ModifyDBSnapshotAttribute: Modify RDS database snapshot attributes • CreateDBSnapshot: Create an RDS snapshot • PutBucketPolicy: Modify S3 bucket policy to make it publicly accessible • PutBucketAcl: Modify the ACL of an S3 bucket to make it publicly accessible
Impact (Impact)	Provides visibility into attacker actions targeting data, systems, and networks (Detect actions like data deletion or system disruption) • PutBucketVersioning: Enable or disable versioning for an S3 bucket • RunInstances: Launch new EC2 instances (Can be exploited for denial-of-service attacks by incurring event execution costs) • DeleteAccountPublicAccessBlock: Deletes the S3 public access block setting • DeleteObject: Delete objects in an S3 bucket • DeleteDBInstance: Delete an RDS database instance • ModifyDBInstance: Modify an RDS database instance

2) VPC Flow Logs

When stored in S3, VPC Flow Logs are logged in plain text or Parquet format (a column-based data format using Gzip compression). When stored in CloudWatch, the CloudWatch service console logs them. Key fields used for incident analysis are as follows.

[Table129] VPC Flow Logs Example

Log Example
123456789012 eni-1a2b3c4d 203.0.113.10 172.31.5.10 54321 22 6 1 40 1678886400 1678886401 REJECT OK

[Table130] VPC Flow Logs fields primarily used for incident analysis

Field Category	Description
account-id	AWS account ID of the owner of the source network interface where traffic is logged
interface-id	The network interface ID where traffic is logged
srcaddr	For incoming traffic: Source IP address of the traffic For outgoing traffic: IP address of the network interface sending the traffic
dstaddr	For outgoing traffic: Destination IP address of the traffic For incoming traffic: IP address of the network interface receiving the traffic
srcport	The port used in the srcaddr of the traffic
dstport	The port used in the traffic's destination address
protocol	IANA protocol number of the traffic (e.g., TCP 6, UDP 17)
packets	Number of packets transmitted in network traffic
bytes	Number of bytes transmitted in network traffic
start	Time when the first packet of network traffic was received within the aggregation interval (Unix Timestamp)
end	Time when the last packet of network traffic was received within the aggregation interval (Unix Timestamp)
action	Action associated with the traffic (ACCEPT, REJECT)

VPC Flow Logs events according to MITRE ATT&CK tactics are as follows.

[Table131] Key VPC Flow Logs events according to MITRE ATT&CK tactics

Event Type	Description
Reconnaissance (Reconnaissance)	Detection of an attacker collecting information about the target network (Detection of abnormal port scanning activity from external IP addresses) • An attempt to scan SSH ports on internal servers from an external IP (203.0.113.10) was detected, but the attempt was blocked (REJECTED) by security groups and network ACLs. 123456789012 eni-1a2b3c4d **203.0.113.10 172.31.5.10** 54321 **22** 6 1 40 1678886400 1678886401 **REJECT OK** 123456789012 eni-1a2b3c4d **203.0.113.10 172.31.5.11** 54321 **22** 6 1 40 1678886401 1678886402 **REJECT OK** 123456789012 eni-1a2b3c4d **203.0.113.10 172.31.5.12** 54321 **22** 6 1 40 1678886402 1678886402 **REJECT OK**
Initial Access (Initial Penetration)	Detection of the attacker's initial penetration into the system (Detection of access from an external IP, detection of access from unknown external IPs or malicious IPs) • Example log of successful RDP access from external IP (203.0.113.10) to internal server (172.31.5.10) 123456789012 eni-1a2b3c4d **203.0.113.10 172.31.5.10** 54321 **3389** 6 1 40 1678886400 1678886430 **ACCEPT OK**
Lateral Movement (Internal Movement)	Detection of an attacker's attempt to move from the initially compromised system to other internal systems (Analysis of abnormal communication between systems that do not normally communicate) • Example log of abnormal communication occurring via SMB port from one internal IP to another 123456789012 eni-1a2b3c4d **172.31.5.10 172.31.5.20 445** 55555 6 100 10000 1678888000 1678888010 **ACCEPT OK**
Command and Control (Command and Control)	Detection of attempts by attackers to communicate with compromised systems to execute commands or control malware (Analyzing the possibility of malware communicating with external C2 servers) • Transmission of 5GB data from internal server to external IP 123456789012 eni-1a2b3c4d **172.31.5.10 198.51.100.1 54321 443** 6 200 20000 1678889000 1678889030 **ACCEPT OK**
Exfiltration (Exfiltration)	Detection of an attacker attempting to exfiltrate data from a compromised system (Suspected data exfiltration when large amounts of outbound traffic occur at unusual times) • Example log of 5GB data transfer from internal server to external IP 123456789012 eni-1a2b3c4d **172.31.5.10 203.0.113.1** 54321 80 6 100000 **5368709120** 1678890000 1678910020 **ACCEPT OK**

3) S3 Server Access Log

The S3 Server Access Log consists of a list of fields separated by spaces and is logged in text file format. Each record contains information about a single S3 request. The key fields used for incident analysis are as follows.

[Table132] S3 Server Access Log Example

Log Example
123456789012 eni-1a2b3c4d 203.0.113.10 172.31.5.10 54321 22 6 1 40 1678886400 1678886401 REJECT OK

[Table133] S3 Server Access Log Fields Primarily Used in Incident Analysis

Field Description	Description
Bucket Owner	AWS ID of the S3 bucket owner
Bucket	Name of the S3 bucket where the request originated
Time	Time the request was completed (UTC)
Remote IP	The client IP address that sent the request
Requester	If the requester is an IAM user, the IAM user name and the AWS account to which the user belongs
Request ID	The ID generated to uniquely identify each request in Amazon S3
Operation	The action performed on an S3 bucket through a request
Key	The object name in the request
Request-URI	Content of the Request-URI within the HTTP request message
HTTP Status	HTTP response code for the request (200, 403, 404, etc.)
Error Code	S3 error code when an error occurs
Bytes Sent	Number of bytes sent in the response
Object Size	Size of the object
Total Time	Time taken for the S3 bucket to process the request
User-Agent	Information about the client application that sent the request

The S3 Server Access Log operations according to MITRE ATT&CK tactics are as follows.

[Table134] Key S3 Server Access Log operations according to MITRE ATT&CK tactics

Event Type	Description
Privilege Escalation (Privilege Escalation)	Provides visibility into an attacker's attempt to elevate from lower privileges to higher ones (Detection of actions modifying object or S3 bucket permissions) • REST.PUT.ACL: Modifying the ACL of an object or S3 bucket
Discovery (Scanning)	Provides visibility into an attacker's attempts to explore the environment of an S3 bucket (Detecting information gathering activities regarding S3 listings, permissions, and configuration details) • REST.GET.BUCKET: Retrieve S3 bucket list • REST.GET.ACL: Retrieve ACL for an object or S3 bucket • REST.GET.BUCKET.LOCATION: Retrieve S3 bucket region information • REST.GET.ENCRYPTION: Retrieve encryption configuration information
Exfiltration (Data Leakage)	Provides visibility into attempts by attackers to exfiltrate data outside the S3 bucket (Detect actions like downloading data from S3 buckets or copying data to attacker S3 buckets for external exfiltration) • REST.GET.OBJECT: Object Download Request • REST.COPY.OBJECT: Object copy request
Impact (Impact)	Provides visibility into attacker actions attempting to impact data (Detects attempts to delete objects within an S3 bucket and delete the S3 bucket itself) • REST.DELETE.OBJECT: Object Deletion Request • REST.DELETE.BUCKET: Request to delete an S3 bucket

4) CloudWatch Logs

By leveraging CloudWatch Logs Insights, you can perform anomaly-based detection queries on logs collected by AWS to counter attack tactics. The detection objectives and example queries for representative log types are as follows.

[Table135] Example of anomaly detection using CloudWatch

Behavior Detection Category	Description
Abnormal Login Attempt Detection (CloudTrail Log)	Aggregate failed console login (ConsoleLogin) events by IP address over the last 24 hours Detect brute-force attack attempts by analyzing abnormal login failure patterns • Detection Query Example fields @timestamp, @message | filter eventName = 'ConsoleLogin' and errorMessage = 'Failed authentication' | stats count(*) as login_failures by sourceIPAddress | sort login_failures desc
New Access Key Abuse Detection (CloudTrail Log)	Identify multiple API calls made within a short timeframe using a newly created access key Detect key theft or misuse • Detection Query Example fields @timestamp, eventName, userIdentity.arn, requestParameters.userName, sourceIPAddress | filter eventName = "CreateAccessKey" | sort @timestamp desc | limit 50
Port Scanning Detection (VPC Flow Logs)	Identify source IPs with abnormally high REJECT responses in network traffic Detect suspicious traffic corresponding to reconnaissance tactics like port scanning • Detection Query Example fields @timestamp, @message | filter action = 'REJECT' | stats count(*) as rejected_packets by srcAddr | sort rejected_packets desc | limit 10
Unauthorized Access Detection (S3 Access Log)	Identify unauthorized IP or IAM user access attempts to buckets Detect suspicious activities corresponding to data exfiltration tactics • Detection Query Example fields @timestamp, requester, bucket, requestUri, status | filter bucket = 'important-data-bucket' and status = 'AccessDenied' | stats count(*) as denied_access by requester, remoteIP | sort denied_access desc

5) RDS Logs

The structure of database logs varies depending on the database engine (MySQL, PostgreSQL, MariaDB, etc.) and log type. While each log typically includes common fields such as timestamp, connection information, and event details, the specific format differs. AWS DB Instance Events are logged in JSON format, with the basic event structure as follows.

[Table136] RDS Logs Example

Log Example
{ "version": "0", "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e", "detail-type": "RDS DB Instance Event", "source": "aws.rds", "account": "123456789012", "time": "2018-09-27T22:36:43Z", "region": "us-east-1", "resources": ["arn:aws:rds:us-east-1:123456789012:db:my-db-instance"], "detail": { "EventCategories": ["failover"], "SourceType": "DB_INSTANCE", "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance", "Date": "2018-09-27T22:36:43.292Z", "Message": "A Multi-AZ failover has completed.", "SourceIdentifier": "my-db-instance", "EventID": "RDS-EVENT-0049" } }

[Table137] AWS DB Instance Events Basic Structure

Field Separation	Description
ID	Event ID (Unique identifier)
Detail-type	Specific type of event (used as a filter key in EventBridge rules)
Account	AWS account ID that generated the event
Time	Time the event occurred (UTC)
Region	AWS region where the event occurred
EventCategories	Event classification (availability, security, configuration change, etc.)
Data	Time the event occurred (UTC)
Message	Description of the event
SourceIdentifier	Name of the resource where the event occurred
EventID	Unique ID for the event that occurred

RDS Log events according to MITRE ATT&CK tactics are as follows.

[Table138] Key RDS Log events according to MITRE ATT&CK tactics

Event Type	Content
Defense Evasion (Privilege Escalation)	Provides visibility into attacker actions aimed at disabling detection and defense systems (Detection of AWS DB instance event logging termination) • RDS-EVENT-0332: Disabling a dedicated log volume
Exfiltration (Data Leakage)	Provides visibility into attempts by attackers to exfiltrate data externally (Detecting attempts to expose DB instance classes externally by changing their visibility settings) • RDS-EVENT-0014: Changes to DB instance class applied
Impact (Impact)	Provides visibility into attempts by attackers to impact data (Detection of actions such as deleting a DB instance, deactivating backups before encryption, deleting snapshots, etc.) • RDS-EVENT-0003: DB instance deletion • RDS-EVENT-0041: User snapshot deletion • RDS-EVENT-0028: Automatic backup deactivation

6) GuardDuty Findings

GuardDuty Findings are logged in JSON format and contain various details essential for incident analysis. The most critical element is the Finding Type, which indicates the type of detected threat. The Finding Type structure is as follows.

[Table139] GuardDuty Findings Type Structure

Finding Type Structure Format	
ThreatPurpose:ResourceTypeAffected/ThreatFamilyName.DetectionMechanism!Artifact	
Field Separation	**Description**
ThreatPurpose	Primary purpose of the threat (Backdoor, DefenseEvasion, Discovery, Recon, etc.)
ResourceTypeAffected	AWS resources targeted by the attack
ThreatFamilyName	Threat or malicious activity name
DetectionMechanism	Method used to detect the threat (e.g., TCP, UDP)
Artifact	Artifacts related to the threat (additional information)

In addition, Findings include the following key information.

[Table140] GuardDuty Findings Key Information

Category	Description
Severity	Indicates the risk level of the threat, categorized as High, Medium, or Low
Account ID	The ID of the AWS account where the threat was detected
Region	The AWS region where the threat occurred
Resource Information	Specific details about the affected resources (e.g., EC2 instance ID, S3 bucket name, etc.)
Attacker Information	Information about the entity attempting the attack (IP, location, attack group, etc.)
Event occurrence time	Records the time when the threat activity first occurred and the time it was last detected

The GuardDuty Findings utilization plan is as follows.

[Table141] GuardDuty Findings Utilization Methods

Category	Description
Automated Alerts and Response	Integrate Amazon EventBridge with AWS Lambda to automatically receive notifications or take response actions when when Findings occur
Centralized Log Analysis and Visualization	Generated Findings are exported to an Amazon S3 bucket for long-term storage, and analyzed using SQL queries via Amazon Athena Integrate with visualization tools like Amazon OpenSearch Service or QuickSight to build dashboards to identify threat trends over time and potential security vulnerabilities.
Initial Incident Analysis	When an incident occurs, use the information in Findings—such as attacker IPs, affected resources, and event timestamps— to understand the overall flow and scope of the attack.
Utilizing Trusted IP and Threat Intelligence Lists	You can directly upload trusted IP lists and threat IP lists to improve detection accuracy (Add shared malicious IP lists to the threat list to quickly detect and respond to known threats)

7) WAF Log

WAF Logs are logged in JSON format and contain various field information for each request. The log structure may vary depending on the WAF version, but generally includes the following key fields.

[Table142] AWS Logs Example

Partial Log Example
"timestamp": 1758865233531, "formatVersion": 1, "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE", "terminatingRuleId": "Test_SQLi_XSS", "terminatingRuleType": "REGULAR", "action": "BLOCK", "terminatingRuleMatchDetails": [{ "conditionType": "SQL_INJECTION", "sensitivityLevel": "HIGH", "location": "HEADER", "matchedData": ["10", "AND", "1"] }], - Omitted below -

[Table143] WAF Log Key Fields

Field Category	Description
timestamp	Time the log was generated
formatVersion	Version of the log format
webaclId	ID of the Web ACL that processed the request
terminatingRuleId	The rule ID that ultimately allowed or blocked the request
action	Action performed by the rule (ALLOW, BLOCK, COUNT, etc.)
terminatingRuleMatchDetails	The specific conditions under which the request matched the rule
httpRequest	Detailed information about the HTTP request - clientIp: IP address of the client that sent the request - country: Country code of the client IP (e.g., KR, US) - headers: Request header information - uri: The requested URI path - args: Query string included in the request - httpVersion: HTTP version - httpMethod: HTTP method used in the request (e.g., GET, POST)
rateBasedRuleList	List of IPs managed by rate-based rules (if applicable)

WAF Log analysis enables detection of various security threats and abnormal activities. Key use cases include:

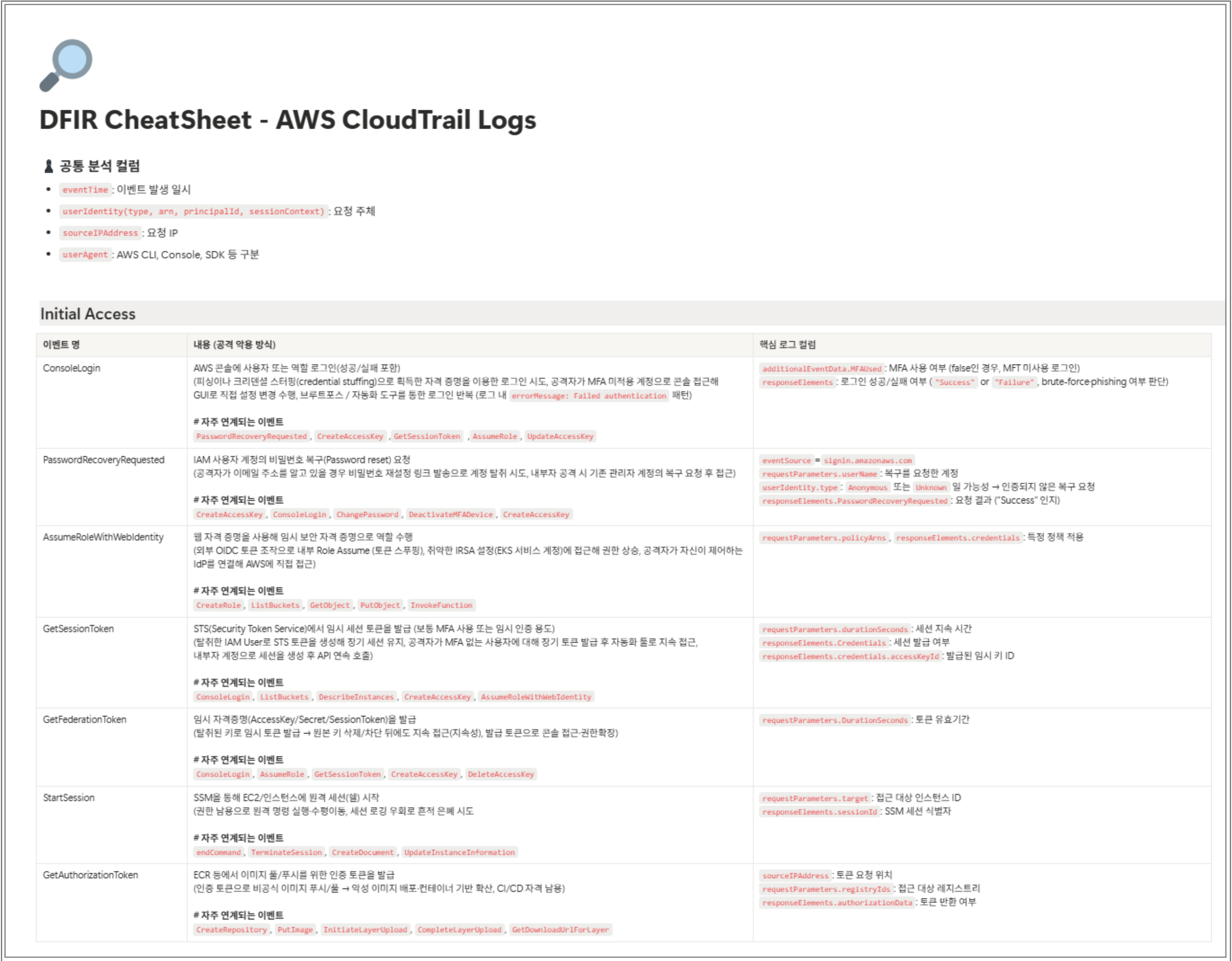
[Table144] WAF Log Use Cases

Category	Description
SQL Injection (SQL Injection)	When a pattern related to an SQL query is detected in the 'terminatingRuleMatchDetails' field
Cross-Site Scripting (XSS)	If script tags (<script>, </script>) or similar elements are found in the request's URI, query string, or body Malicious Scripts
Directory Traversal (Directory Traversal)	Attempts to navigate to a parent directory (../) are detected in the URI or parameters
Denial of Service (DoS/DDoS)	When an abnormally high number of requests originate from a specific client IP within a short timeframe

5.2. Log Event Mapping by Attack Tactics DFIR CheatSheet Development

This DFIR CheatSheet maps and classifies AWS logs (CloudTrail Log, S3 Access Log) based on MITRE ATT&CK tactics for incident response. It standardizes key events per tactic, frequently associated events, attacker exploitation patterns, and core DFIR columns for each event to enhance detection and analysis consistency and efficiency.

The CheatSheet includes 175 events from CloudTrail Log and 47 operations from S3 Access Log.



DFIR CheatSheet - AWS CloudTrail Logs

공통 분석 컬럼

- `eventTime` : 이벤트 발생 일시
- `userIdentity(type, arn, principalId, sessionContext)` : 요청 주체
- `sourceIPAddress` : 요청 IP
- `userAgent` : AWS CLI, Console, SDK 등 구분

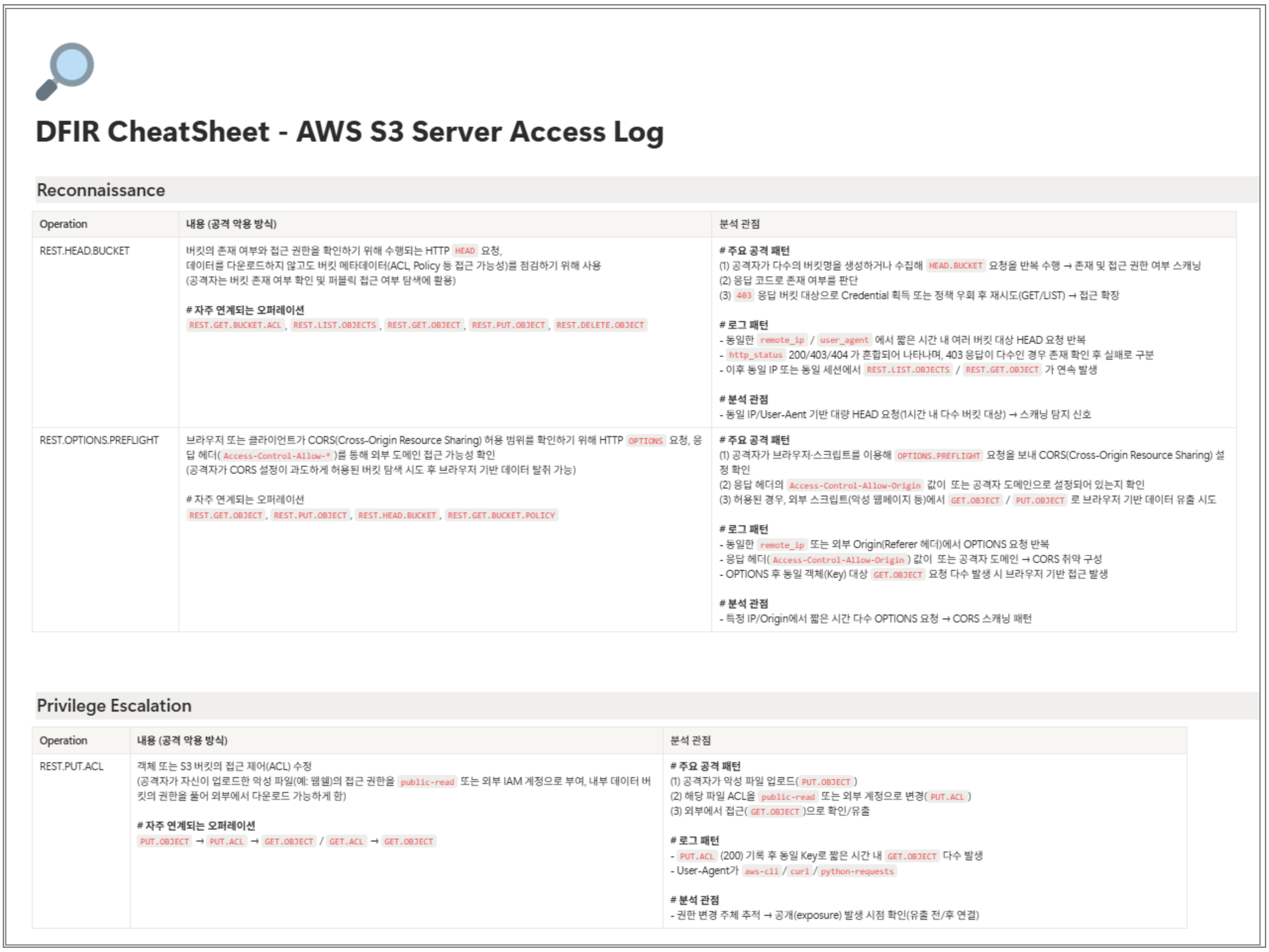
Initial Access

이벤트 명	내용 (공격 악용 방식)	핵심 로그 컬럼
ConsoleLogin	AWS 콘솔에 사용자 또는 역할 로그인(성공/실패 포함) (피싱이나 크리덴셜 스터핑(credential stuffing)으로 획득한 자격 증명을 이용한 로그인 시도, 공격자가 MFA 미적용 계정으로 콘솔 접근해 GUI로 직접 설정 변경 수행, 브루트포스 / 자동화 도구를 통한 로그인 반복 (로그 내 `errorMessage: Failed authentication` 패턴) **# 자주 연계되는 이벤트** `PasswordRecoveryRequested` , `CreateAccessKey` , `GetSessionToken` , `AssumeRole` , `UpdateAccessKey`	`additionalEventData.MFAUsed` : MFA 사용 여부 (false인 경우, MFT 미사용 로그인) `responseElements` : 로그인 성공/실패 여부 (`"Success"` or `"Failure"` , brute-force·phishing 여부 판단)
PasswordRecoveryRequested	IAM 사용자 계정의 비밀번호 복구(Password reset) 요청 (공격자가 이메일 주소를 알고 있을 경우 비밀번호 재설정 링크 발송으로 계정 탈취 시도, 내부자 공격 시 기존 관리자 계정의 복구 요청 후 접근) **# 자주 연계되는 이벤트** `CreateAccessKey` , `ConsoleLogin` , `ChangePassword` , `DeactivateMFADevice` , `CreateAccessKey`	`eventSource` = `signin.amazonaws.com` `requestParameters.userName` : 복구를 요청한 계정 `userIdentity.type` : `Anonymous` 또는 `Unknown` 일 가능성 → 인증되지 않은 복구 요청 `responseElements.PasswordRecoveryRequested` : 요청 결과 ("Success" 인지)
AssumeRoleWithWebIdentity	웹 자격 증명을 사용해 임시 보안 자격 증명으로 역할 수행 (외부 OIDC 토큰 조작으로 내부 Role Assume (토큰 스푸핑), 취약한 IRSA 설정(EKS 서비스 계정)에 접근해 권한 상승, 공격자가 자신이 제어하는 IdP를 연결해 AWS에 직접 접근) **# 자주 연계되는 이벤트** `CreateRole` , `ListBuckets` , `GetObject` , `PutObject` , `InvokeFunction`	`requestParameters.policyArns` , `responseElements.credentials` : 특정 정책 적용

GetSessionToken	STS(Security Token Service)에서 임시 세션 토큰을 발급 (보통 MFA 사용 또는 임시 인증 용도)
	(탈취한 IAM User로 STS 토큰을 생성해 장기 세션 유지, 공격자가 MFA 없는 사용자에 대해 장기 토큰 발급 후 자동화 툴로 지속 접근, 내부자 계정으로 세션을 생성 후 API 연속 호출)
	# 자주 연계되는 이벤트
	`ConsoleLogin` , `ListBuckets` , `DescribeInstances` , `CreateAccessKey` , `AssumeRoleWithWebIdentity`
		`requestParameters.durationSeconds` : 세션 지속 시간
		`responseElements.Credentials` : 세션 발급 여부
		`responseElements.credentials.accessKeyId` : 발급된 임시 키 ID

GetFederationToken	임시 자격증명(AccessKey/Secret/SessionToken)을 발급 (탈취된 키로 임시 토큰 발급 → 원본 키 삭제/차단 뒤에도 지속 접근(지속성), 발급 토큰으로 콘솔 접근·권한확장) **# 자주 연계되는 이벤트** `ConsoleLogin` , `AssumeRole` , `GetSessionToken` , `CreateAccessKey` , `DeleteAccessKey`	`requestParameters.DurationSeconds` : 토큰 유효기간
StartSession	SSM을 통해 EC2/인스턴스에 원격 세션(쉘) 시작 (권한 남용으로 원격 명령 실행·수평이동, 세션 로깅 우회로 흔적 은폐 시도 **# 자주 연계되는 이벤트** `endCommand` , `TerminateSession` , `CreateDocument` , `UpdateInstanceInformation`	`requestParameters.target` : 접근 대상 인스턴스 ID `responseElements.sessionId` : SSM 세션 식별자
GetAuthorizationToken	ECR 등에서 이미지 풀/푸시를 위한 인증 토큰을 발급 (인증 토큰으로 비공식 이미지 푸시/풀 → 악성 이미지 배포·컨테이너 기반 확산, CI/CD 자격 남용) **# 자주 연계되는 이벤트** `CreateRepository` , `PutImage` , `InitiateLayerUpload` , `CompleteLayerUpload` , `GetDownloadUrlForLayer`	`sourceIPAddress` : 토큰 요청 위치 `requestParameters.registryIds` : 접근 대상 레지스트리 `responseElements.authorizationData` : 토큰 반환 여부

[Figure6] DFIR Cheat Sheet – AWS CloudTrail Log Content Excerpt



DFIR CheatSheet - AWS S3 Server Access Log

Reconnaissance

Operation	내용 (공격 악용 방식)	분석 관점
REST.HEAD.BUCKET	버킷의 존재 여부와 접근 권한을 확인하기 위해 수행되는 HTTP `HEAD` 요청, 데이터를 다운로드하지 않고도 버킷 메타데이터(ACL, Policy 등 접근 가능성)를 점검하기 위해 사용 (공격자는 버킷 존재 여부 확인 및 퍼블릭 접근 여부 탐색에 활용) **# 자주 연계되는 오퍼레이션** `REST.GET.BUCKET.ACL` , `REST.LIST.OBJECTS` , `REST.GET.OBJECT` , `REST.PUT.OBJECT` , `REST.DELETE.OBJECT`	**# 주요 공격 패턴** (1) 공격자가 다수의 버킷명을 생성하거나 수집해 `HEAD.BUCKET` 요청을 반복 수행 → 존재 및 접근 권한 여부 스캐닝 (2) 응답 코드로 존재 여부를 판단 (3) `403` 응답 버킷 대상으로 Credential 획득 또는 정책 우회 후 재시도(GET/LIST) → 접근 확장 **# 로그 패턴** - 동일한 `remote_ip` / `user_agent` 에서 짧은 시간 내 여러 버킷 대상 HEAD 요청 반복 - `http_status` 200/403/404 가 혼합되어 나타나며, 403 응답이 다수인 경우 존재 확인 후 실패로 구분 - 이후 동일 IP 또는 동일 세션에서 `REST.LIST.OBJECTS` / `REST.GET.OBJECT` 가 연속 발생 **# 분석 관점** - 동일 IP/User-Aent 기반 대량 HEAD 요청(1시간 내 다수 버킷 대상) → 스캐닝 탐지 신호
REST.OPTIONS.PREFLIGHT	브라우저 또는 클라이언트가 CORS(Cross-Origin Resource Sharing) 허용 범위를 확인하기 위해 HTTP `OPTIONS` 요청, 응답 헤더(`Access-Control-Allow-*`)를 통해 외부 도메인 접근 가능성 확인 (공격자가 CORS 설정이 과도하게 허용된 버킷 탐색 시도 후 브라우저 기반 데이터 탈취 가능) # 자주 연계되는 오퍼레이션 `REST.GET.OBJECT` , `REST.PUT.OBJECT` , `REST.HEAD.BUCKET` , `REST.GET.BUCKET.POLICY`	**# 주요 공격 패턴** (1) 공격자가 브라우저·스크립트를 이용해 `OPTIONS.PREFLIGHT` 요청을 보내 CORS(Cross-Origin Resource Sharing) 설정 확인 (2) 응답 헤더의 `Access-Control-Allow-Origin` 값이 또는 공격자 도메인으로 설정되어 있는지 확인 (3) 허용된 경우, 외부 스크립트(악성 웹페이지 등)에서 `GET.OBJECT` / `PUT.OBJECT` 로 브라우저 기반 데이터 유출 시도 **# 로그 패턴** - 동일한 `remote_ip` 또는 외부 Origin(Referer 헤더)에서 OPTIONS 요청 반복 - 응답 헤더(`Access-Control-Allow-Origin`) 값이 또는 공격자 도메인 → CORS 취약 구성 - OPTIONS 후 동일 객체(Key) 대상 `GET.OBJECT` 요청 다수 발생 시 브라우저 기반 접근 발생 **# 분석 관점** - 특정 IP/Origin에서 짧은 시간 다수 OPTIONS 요청 → CORS 스캐닝 패턴

Privilege Escalation

Operation	내용 (공격 악용 방식)	분석 관점
REST.PUT.ACL	객체 또는 S3 버킷의 접근 제어(ACL) 수정 (공격자가 자신이 업로드한 악성 파일(예: 웹쉘)의 접근 권한을 `public-read` 또는 외부 IAM 계정으로 부여, 내부 데이터 버킷의 권한을 풀어 외부에서 다운로드 가능하게 함) **# 자주 연계되는 오퍼레이션** `PUT.OBJECT` → `PUT.ACL` → `GET.OBJECT` / `GET.ACL` → `GET.OBJECT`	**# 주요 공격 패턴** (1) 공격자가 악성 파일 업로드(`PUT.OBJECT`) (2) 해당 파일 ACL을 `public-read` 또는 외부 계정으로 변경(`PUT.ACL`) (3) 외부에서 접근(`GET.OBJECT`)으로 확인/유출 **# 로그 패턴** - `PUT.ACL` (200) 기록 후 동일 Key로 짧은 시간 내 `GET.OBJECT` 다수 발생 - User-Agent가 `aws-cli` / `curl` / `python-requests` **# 분석 관점** - 권한 변경 주체 추적 → 공개(exposure) 발생 시점 확인(유출 전/후 연결)

[Figure7] DFIR Cheatsheet – AWS S3 Server Access Logs Content Excerpt

The key events for each tactic in CloudTrail Log are as follows.

[Table145] Key Events per Tactics as Documented in the DFIR CheatSheet - CloudTrail

Function Category	Event Name	Event Description
Initial Access (Initial Penetration)	ConsoleLogin	User or role login to the AWS console
	PasswordRecoveryRequested	Password reset request for an IAM user account
	AssumeRoleWithWebIdentity	Assume a role using web credentials as temporary security credentials
	GetSessionToken	Issuing a temporary session token from the Security Token Service (STS)
	GetFederationToken	Issues temporary credentials (AccessKey/Secret/SessionToken)
	StartSession	Start a remote session (shell) on an EC2 instance via SSM
	GetAuthorizationToken	Issues an authorization token for image pull/push operations in ECR, etc.
Execution (Execute)	StartInstance	Start a stopped EC2 instance
	StartInstances	Start multiple stopped EC2 instances
	Invoke	Invoke an AWS Lambda function
	SendCommand	Send command to EC2 instance
Persistence (Persistence)	CreateAccessKey	Create an access key for an AWS user or role
	CreateUser	Create a new IAM user
	CreateNetworkAclEntry	Add inbound/outbound rules to a VPC network ACL
	CreateRoute	Add a new route to the routing table
	CreateLoginProfile	Generate a password for logging into the IAM user console
	AuthorizeSecurityGroupEgress	Modify the security group's egress rules (inbound/outbound) to allow network communication
	AuthorizeSecurityGroupIngress	Modify the security group's inbound rules to allow network communication
	CreateVirtualMFADevice	Create a virtual MFA device
	CreateConnection	Create a Direct Connect connection or VPN connection
	ApplySecurityGroupsToLoadBalancer	Apply security groups to load balancer (ELB)
	SetSecurityGroups	Apply security groups to resources such as EC2 Network Interface, Lambda, ENI, etc. Apply the load balancer's security group directly to resources
	AuthorizeDBSecurityGroupIngress	Add inbound allow rules to the DB security group for RDS
	CreateDBSecurityGroup	Create an RDS security group
	ChangePassword	Change IAM user password
	CreateFunction	Create a new Lambda function (code + configuration)
	CreateTags	Add metadata tags to AWS resources
	DeleteBucketCors	Remove bucket CORS (Cross-Origin Resource Sharing) settings
	DeleteBucketPolicy	Delete an S3 bucket policy
	CreateImage	Create an AMI (System Image) for an EC2 instance
Persistence (Persistence)	CreateInstance	Create an EC2 instance
	CreateKeyPair	Generate an SSH key pair (public key/private key) for SSH access Returns the private key (cannot reuse the key)
	CreateRepository	Create a container registry (Repository)
	PutImage	Upload a container image to an ECR repository

Function Category	Event Name	Event Description
	PutUserData	Configure or modify User Data (scripts executed at boot) for an EC2 instance Configure or modify the User Data (scripts executed at boot) to set up commands that run automatically when the instance starts
	EnableSerialConsoleAccess	Enable EC2 Serial Console functionality
Privilege Escalation (Privilege Escalation)	CreateGroup	Create an IAM group at the organization (group) level
	UpdateAccessKey	Change the status (active/inactive) of an IAM user's Access Key or update the AccessKey value
	PutGroupPolicy	Add or modify an inline policy for a specific IAM group
	PutRolePolicy	Add or modify an inline policy for a specific role
	PutUserPolicy	Grant inline policies to a specific IAM user
	AddRoleToInstanceProfile	Add a role to an EC2 instance profile
	AddUserToGroup	Add a specific IAM user to a specific group
	AttachUserPolicy	Attach an AWS-managed or custom managed policy to an IAM user
	AttachRolePolicy	Attach an IAM managed policy to a role
	AddPermission	Add a policy to allow a specific Principal to call the resource
	UpdateFunctionCode	Update the code package for an existing Lambda function
	CreatePolicy	Create a new IAM policy
	UpdateFunctionConfiguration	Change Lambda function settings
	CreatePolicyVersion	Create a new version for an existing IAM policy
	CreateInstanceProfile	Create an IAM instance profile that can be attached to an EC2 instance
	CreateRole	Create a new IAM role
	PassRole	Delegate an IAM role to a specific service (such as Lambda or EC2) Enable the service to use the granted permissions
Defense Evasion (Defense Evasion)	StopLogging	Disable logging for specific CloudTrail trails
	DeleteTrail	Permanently delete a trail from CloudTrail
	UpdateTrail	Modify CloudTrail trail settings
	PutEventSelectors	Configure logging for CloudTrail trail events (Data/Management)
	DeleteFlowLogs	Delete VPC Flow Logs
	DeleteDetector	Delete GuardDuty detectors to stop detection functionality
	DeleteMembers	Delete GuardDuty member accounts
Defense Evasion (Defense Evasion)	DeleteSnapshot	Delete EBS or RDS Snapshot
	Deactivate MFA Device	Deactivate MFA Device for User Account
	DeleteCertificate	Delete IAM Server/Client Certificate (SSL/TLS)
	DeleteConfigRule	Delete AWS Config rule
	DeleteAccessKey	Delete an IAM user's access key
	LeaveOrganization	Leave the AWS Organization (organization management account)
	DisassociateFromMasterAccount	Disconnect from the master account in AWS GuardDuty or Security Hub
	DisassociateMembers	Disassociate from GuardDuty member accounts
	StopMonitoringMembers	GuardDuty master account stops monitoring members

Function Category	Event Name	Event Description
	DeleteLogGroup	Delete a log group in CloudWatch Logs
	DetachUserPolicy	Detach a managed policy (Policy ARN) from an IAM user
	DeletePolicy	Delete an IAM managed policy
	DisableKey	Disable KMS key
	ScheduleKeyDeletion	Schedule KMS Key Deletion
	DeleteDBCluster	Delete Entire DB Cluster in Amazon RDS Environment
	DeleteDBClusterSnapshot	Delete a backup snapshot (DB Cluster Snapshot) of a DB cluster
	DeletePublicAccessBlock	Remove S3 Public Access Block settings
	RevokeSecurityGroupIngress	Delete inbound rules from EC2 service security group
	RevokeSecurityGroupEgress	Deleting Outbound Rules from EC2 Service Security Groups
	PutMetricAlarm	Create CloudWatch Alarm
	DeleteAlarms	Delete CloudWatch Alarms
	StopConfigurationRecorder	Stop monitoring resource configuration changes in AWS Config
	PutDeliveryChannel	Change AWS Config Data Delivery Channel
	PutKeyPolicy	Modify a specific KMS key policy
	DeleteAlias	Delete Alias in KMS, Lambda, etc.
	CreateAlias	Create an alias in KMS, Lambda, etc.
	DeleteBucketTagging	Delete bucket tagging (identifying metadata)
	PutBucketLifecycle	Set lifecycle rules for S3 buckets (e.g., object expiration) Change automatic deletion and archiving policies
	ModifyNetworkInterfaceAttribute	Change ENI attributes
Credential Access (Acquire credentials)	GetSecretValue	Retrieve secret values stored in AWS Secrets Manager
	PutSecretValue	Add/update a secret value stored in AWS Secrets Manager
	GetPasswordData	Retrieve the Windows administrator password for an EC2 instance in encrypted form
	RequestCertificate	Request a new SSL/TLS certificate from AWS Certificate Manager
Credential Access (Credential Acquisition)	CreateSecret	Create a secret in AWS Secrets Manager
	DeleteSecret	Delete a secret in AWS Secrets Manager
	UpdateAssumeRolePolicy	Modify the trust policy for an IAM role
	ListSecrets	Retrieve metadata (name, description, ARN, etc.) of secrets stored in AWS Secrets Manager
Discovery (Search)	ListUsers	Retrieve a list of IAM users
	ListRoles	List IAM Roles
	ListIdentities	List Users in Cognito/AWS Identity Pool
	ListAccessKeys	List Access Keys for IAM Users
	ListServiceQuotas	View quotas per AWS service
	ListInstanceProfiles	List instance profiles (for EC2 role association)
	ListBucket	List objects within a specific bucket
	ListBuckets	List S3 buckets
	ListGroups	Retrieve a list of IAM groups

Function Category	Event Name	Event Description
	GetSendQuota	SES Mail Send Quota Inquiry
	GetCallerIdentity	Current STS Session/Account Information Query
	DescribeInstances	EC2 Instance Details Lookup
	GetBucketAcl	Retrieve access control (ACL) for an S3 bucket
	GetBucketVersioning	Check Versioning Settings for S3 Bucket
	GetAccountAuthorizationDetails	Retrieve full details of IAM policies, users, and roles
	ListObjects	List objects in an S3 bucket
	HeadObject	Retrieve metadata of an S3 object without downloading its data
	GetBucketPolicy	Retrieve the bucket policy for an S3 bucket
	DescribeDBClusters	Retrieve RDS cluster configuration and endpoints
	DescribeDBClusterSnapshots	List RDS cluster backup snapshots
	GetPublicAccessBlock	Viewing Public Access Block Settings for an Account or Bucket
	GetObjectAcl	Retrieve S3 Object Access Control (ACL)
	GetConsoleScreenshot	Request a console screenshot (virtual screen) of an EC2 instance Receive as an image (Base64)
	BatchGetCommits	Retrieve multiple commit information from a CodeCommit repository Verify code change history
	DescribeTrails	Retrieve CloudTrail trail configuration information
	DescribeSnapshots	Retrieve the list and metadata of EBS snapshots (disk backups)
Lateral Movement (Internal Movement)	AssumeRole	Obtain temporary credentials for another IAM role via STS (Security Token Service) Issuing Temporary Credentials
Lateral Movement (Internal Movement)	SwitchRole	Switching roles within the AWS Management Console to start a session with the permissions of another role
	CreateVpcPeeringConnection	Create a VPC peering connection
	AuthorizeSecurityGroupIngress	Add inbound rules to the security group
	ReplaceRoute	Modify routes in the VPC's Route Table Change the destination of traffic
	CreateGrant	Delegate permissions for a KMS Key to another entity Allow encryption/decryption
	CreateNatGateway	Enable private subnets to communicate with external networks Create a NAT gateway
Exfiltration (Exfiltration)	GetObject	Download or read the actual content of an S3 object
	CopyObject	Copy an S3 object to the same bucket or a different bucket
	CreateSnapShot	Back up the state of an EBS volume as a snapshot
	CopySnapshot	Copy an existing snapshot to another region/account
	ModifySnapshotAttributes	Modify the sharing permissions of an EBS snapshot
	ModifyImageAttribute	Change sharing permissions for an EC2 AMI image
	SharedSnapshotCopyInitiated	Indicates that copying of a shared snapshot has begun
	SharedSnapshotVolumeCreated	New volume created from shared snapshot
	ModifyDBSnapshotAttribute	Modify RDS DB snapshot sharing permissions
	CreateDBSnapshot	Back up the current state of an RDS DB as a snapshot
	PutBucketPolicy	Modify S3 bucket policy

Function Category	Event Name	Event Description
	PutBucketAcl	Modify S3 bucket access control (ACL)
	ModifyDBClusterSnapshotAttribute	Modify the sharing permissions of an RDS cluster snapshot
	RestoreDBClusterFromSnapshot	Restore a new cluster using a cluster snapshot
	PutObjectAcl	Modify the access control list (ACL) for an S3 object
	PutPublicAccessBlock	Modify S3 Public Access Block settings
	CopyDBSnapshot	Copy a DB snapshot between regions/accounts
	RestoreDBInstanceFromDBSnapshot	Restore a new database instance from an existing RDS snapshot
	InvokeFunction	Invoke a Lambda function manually or automatically (triggered)
	DeleteBucketPublicAccessBlock	Delete the public access block setting for an S3 bucket
	CreateKey	Create a KMS master key
	DeleteBucketEncryption	Delete encryption settings for an S3 bucket
	StartExportTask	Start Export Task to an external destination (e.g., S3)
Impact (Impact)	PutBucketVersioning	Enable or disable object versioning for an S3 bucket
Impact (Impact)	RunInstances	Run New EC2 Instances
	DeleteAccountPublicAccessBlock	Delete S3 Public Access Block Policy for Entire AWS Account
	DeleteObject	Delete a single object in an S3 bucket
	DeleteObjects	Batch delete multiple objects (up to 1000) within an S3 bucket
	DeleteDBInstance	Delete an RDS instance
	ModifyDBInstance	Modify RDS instance settings
	PutObject	Upload an S3 object (create or overwrite)
	DeleteBucket	Delete an S3 bucket itself
	DeleteBucketLifecycle	Remove lifecycle rules from an S3 bucket
	DeleteDBSnapshot	Delete an RDS snapshot
	DeleteBucketReplication	Delete bucket replication configuration
	DisableKey	Disable KMS Key
	TerminateInstances	Terminate AWS EC2 instances
	DeleteVolume	Delete AWS EBS Volume
	DeleteRecoveryPoint	Delete Recovery Point in AWS Backup
	EncryptVolume	Encrypt an EBS volume or change encryption configuration
	PutBucketEncryption	Add or change server-side encryption settings for an S3 bucket
	PutBucketReplication	Configure replication rules between S3 buckets to automatically transfer data
	AttachInternetGateway	Connect an Internet Gateway to a VPC to configure external internet communication
	DeleteSecurityGroup	Delete the specified security group to remove network access control configuration

The core operations for each tactic in S3 Access Logs are as follows.

[Table146] Core Operations per Tactics as documented in the DFIR CheatSheet – S3 Access Log

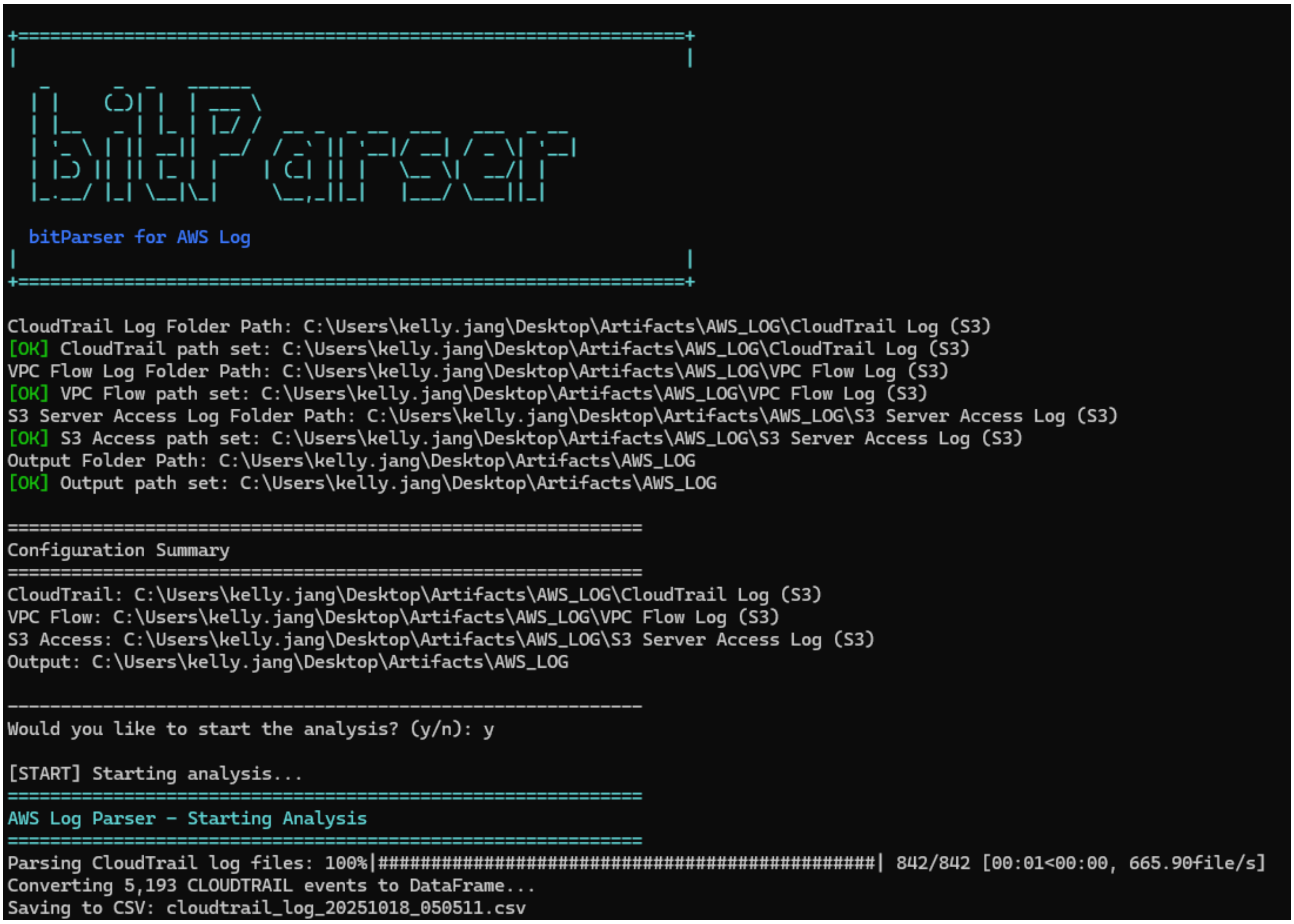
Function Category	Operation Name	Event Description
Reconnaissance (Reconnaissance)	REST.HEAD.BUCKET	To verify the existence and access permissions of a bucket HTTP HEAD request
	REST.OPTIONS.PREFLIGHT	To verify the CORS allowed scope of the browser or client HTTP OPTIONS request
Privilege Escalation (Privilege Escalation)	REST.PUT.ACL	Modifying the access control list (ACL) of an object or S3 bucket
Persistence (Persistence)	REST.PUT.OBJECT	Uploading an S3 object (file) or overwriting an existing object
	REST.PUT.BUCKETNOTIFICATION	Create or update event notification settings for an S3 bucket
	REST.GET.BUCKETNOTIFICATION	Retrieve the event notification configuration set for a bucket
Discovery (Explore)	REST.GET.BUCKET	Retrieve list of S3 buckets
	REST.GET.ACL	Retrieve ACL for an object or S3 bucket
	REST.GET.BUCKET.LOCATION	Retrieve S3 bucket region information
	REST.GET.ENCRYPTION	Retrieve the bucket's default encryption (SSE) setting
	REST.GET.BUCKETACL	Retrieve S3 bucket ACL
	REST.GET.BUCKETPOLICY	Retrieve S3 bucket policy content
	REST.GET.SERVICE	List all buckets existing at the account (service) level
	REST.LIST.MULTIPART.UPLOADS	Retrieve the list of currently in-progress multipart uploads in a specific bucket
	REST.HEAD.OBJECT	Check object existence/metadata (size, ETag, Content-Type, etc.)
	REST.GET.OBJECT.VERSION	Retrieve a specific version of an object with versioning enabled
	REST.LIST.OBJECT.VERSIONS	All versions per object in a bucket with versioning enabled Listing
Defense Evasion (Defense Evasion)	REST.DELETE.BUCKETPUBLICACCESSBLOCK	Remove S3 PublicAccessBlock settings at the organization/account level
	DELETE.OBJECT.VERSION	Delete a specific version of an object with versioning enabled
	REST.GET.OBJECT.TAGGING	Retrieve tags (metadata) for an object
	REST.PUT.OBJECT.TAGGING	Modify an object's tags (metadata)
	REST.PUT.BUCKETVERSIONING	Enable or disable bucket versioning
	REST.GET.BUCKETVERSIONING	Retrieve the bucket's versioning status (enabled or disabled)
	REST.GET.BUCKETLIFECYCLE	Retrieve lifecycle rules configured for a bucket
	REST.PUT.OBJECT.RETENTION	Set retention period for a specific object
	REST.PUT.OBJECT.LEGALHOLD	Enable/disable legal hold on an object
	REST.PUT.BUCKETLOGGING	Enable/change server access logging for a bucket (specify log destination)
	REST.GET.BUCKETLOGGING	Retrieve the access logging configuration set for a bucket
Defense Evasion (Defense Evasion)	REST.DELETE.BUCKETLOGGING	Disable access logging for a bucket
	REST.DELETE.BUCKETNOTIFICATION	Delete event notifications set for the bucket
	REST.DELETE.BUCKETREPLICATION	Remove replication settings from the bucket
Exfiltration	REST.GET.OBJECT	Download S3 object (file)

Function Category	Operation Name	Event Description
(Exfiltration)	REST.COPY.OBJECT	Copy S3 object to same/different bucket (or account)
	REST.PUT.BUCKETACL	Modify the ACL for the entire bucket
	REST.PUT.BUCKETPOLICY	Create and modify bucket policies to control access
	REST.PUT.BUCKET	Create a new S3 bucket
	REST.INITIATE.MULTIPART.UPLOAD	Initiate a multipart upload and issue an uploadId (session)
	REST.UPLOAD.PART	Upload an individual part of a multipart upload
	REST.COMPLETE.MULTIPART.UPLOAD	Combines uploaded parts into a single object and completes the upload
	REST.ABORT.MULTIPART.UPLOAD	Aborts an in-progress multipart upload and cleans up related parts
	REST.PUT.BUCKETREPLICATION	Configure bucket replication rules to automatically replicate objects to other buckets REST.PUT.BUCKETREPLICATION
	REST.GET.BUCKETREPLICATION	Retrieve replication settings configured for a bucket
	REST.RESTORE.OBJECT	Temporarily restore objects stored in Glacier/Archive and set access permissions
	REST.GET.OBJECT.TORRENT	(Legacy) Request to download an S3 object using BitTorrent
Impact (Impact)	REST.DELETE.OBJECT	Delete a specific object
	REST.DELETE.BUCKET	Delete the bucket itself

5.3. Development of AWS DFIR Log Analysis Tool

To support incident detection and investigation in the AWS cloud environment, we developed the AWS DFIR Log Analysis Tool (bitParser for AWS Log). This tool collects and parses CloudTrail Logs, VPC Flow Logs, and S3 Access Logs. Its goal is to automatically detect log patterns frequently observed in actual attacks and present key analysis points that analysts should prioritize for review.

Furthermore, for CloudTrail Logs and S3 Access Logs, it concurrently analyzes whether key events outlined in the previously presented DFIR CheatSheet are detected. This provides higher visibility from a tactical-based linkage and correlation perspective.



```
+==============================================================+
|                                                              |
   _     _ _   _____
  | |   (_) | |  __ \
  | |__  _| |_| |__) |_ _ _ __ ___  ___ _ __
  | '_ \| | __|  ___/ _` | '__/ __|/ _ \ '__|
  | |_) | | |_| |  | (_| | |  \__ \  __/ |
  |_.__/|_|\__|_|   \__,_|_|  |___/\___|_|

  bitParser for AWS Log
|                                                              |
+==============================================================+

CloudTrail Log Folder Path: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG\CloudTrail Log (S3)
[OK] CloudTrail path set: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG\CloudTrail Log (S3)
VPC Flow Log Folder Path: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG\VPC Flow Log (S3)
[OK] VPC Flow path set: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG\VPC Flow Log (S3)
S3 Server Access Log Folder Path: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG\S3 Server Access Log (S3)
[OK] S3 Access path set: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG\S3 Server Access Log (S3)
Output Folder Path: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG
[OK] Output path set: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG

============================================================
Configuration Summary
============================================================
CloudTrail: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG\CloudTrail Log (S3)
VPC Flow: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG\VPC Flow Log (S3)
S3 Access: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG\S3 Server Access Log (S3)
Output: C:\Users\kelly.jang\Desktop\Artifacts\AWS_LOG

------------------------------------------------------------
Would you like to start the analysis? (y/n): y

[START] Starting analysis...
============================================================
AWS Log Parser - Starting Analysis
============================================================
Parsing CloudTrail log files: 100%|##################################################| 842/842 [00:01<00:00, 665.90file/s]
Converting 5,193 CLOUDTRAIL events to DataFrame...
Saving to CSV: cloudtrail_log_20251018_050511.csv
```

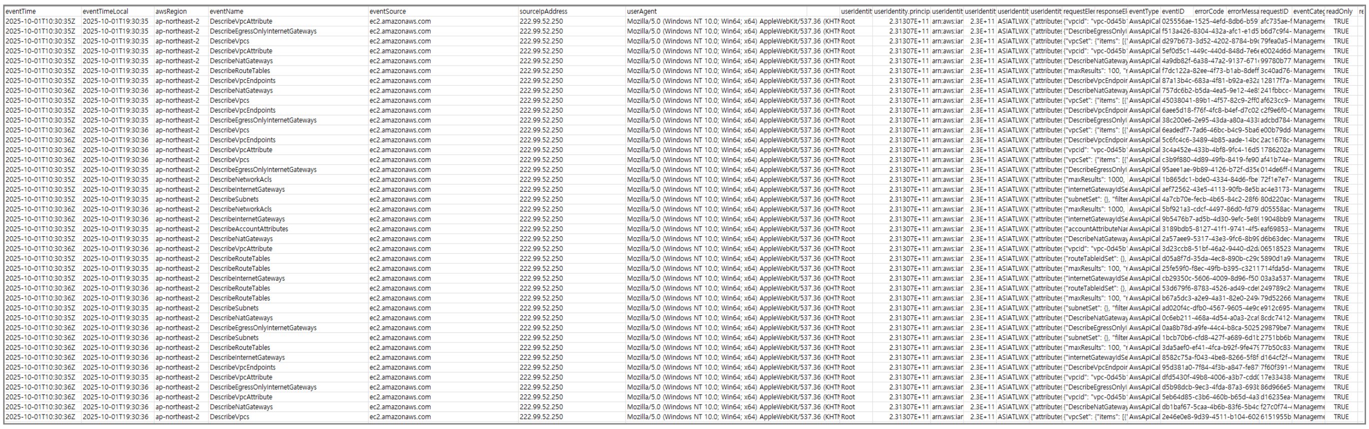
[Figure8] bitParser for AWS Log execution screen example

The tool's main features are as follows.

[Table147] bitParser for AWS Log Key Features

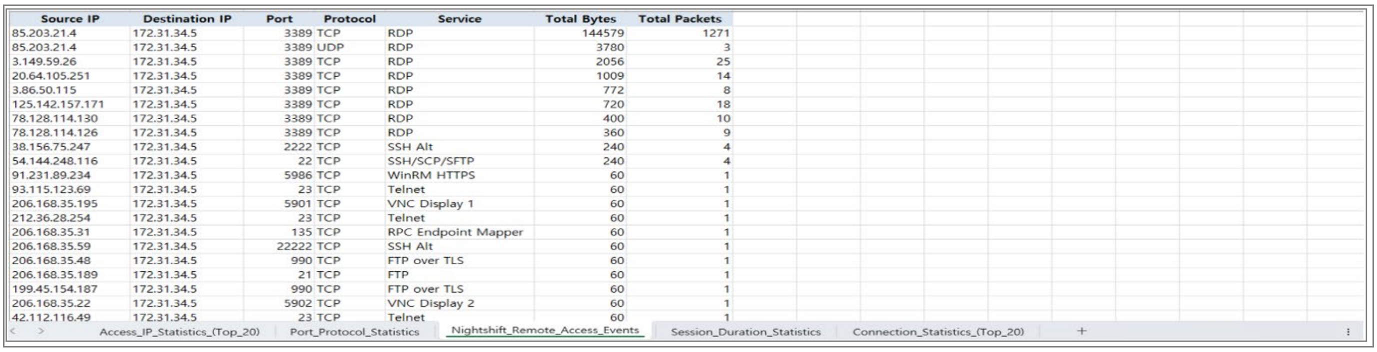
Function Category	Description
Integrated Log Parsing	• CloudTrail Log: Parsing AWS API call events (JSON flattening) • VPC Flow Logs: Parsing network traffic logs • S3 Access Log: Parsing S3 bucket access logs
CloudTrail Log Analysis	• Access history statistics for the top 20 IPs that triggered events (first/last access date and time, number of accesses) • IP statistics for events called during overnight hours (22:00 – 06:00) • Analysis of event frequency occurring during overnight hours (22:00 – 06:00) • Overall event-based statistics and frequency analysis • Detailed User-Agent Classification and Statistics (AWS CLI, SDK, Browser, etc.) • Account creation history analysis • Analysis of AWS Management Console login history • Failed Authentication/Permissions Statistics • AWS Region Statistics and Frequency Analysis • MITRE ATT&CK Tactics-Based Event Statistics (175 events listed in the DFIR Cheat Sheet) • All CloudTrail Log Detailed Logs Mapped to MITRE ATT&CK Tactics Events
VPC Flow Logs Analysis	• Top 20 IPs by Traffic Volume (srcIP, dstIP) • Network Traffic Statistics for Top 20 Ports • Remote access events (RDP, SSH, etc.) occurring during overnight hours (22:00 – 06:00) • Top 20 session duration statistics • Top 20 Network Traffic Statistics by Total Bytes
S3 Access Log analysis	• Top 20 requester ARN and IP statistics • Operation occurrence frequency and statistics (Operation, S3 Bucket, Prefix, Occurrence Count) • User-Agent Detailed Classification and Statistics (AWS CLI, SDK, Browser, etc.) • MITRE ATT&CK Tactics-Based Operation Statistics (47 operations listed in the DFIR Cheat Sheet) • All S3 Access Log Detailed Logs Mapped to Operations by MITRE ATT&CK Tactics
Multiple Output Formats	• Parsed raw logs (saved as CSV files in the Parse_Logs folder within the output folder) • Log analysis results sheet (saved as an xlsx file in the Analysis_Log folder within the output folder) • Summary report based on log analysis results (saved as an HTML file in the Report folder within the output folder)
Log time conversion	• Provides additional log times converted based on the AWS region recorded in CloudTrail logs

The tool result files are structured as follows.



eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent		userIdentit	userIdentity.princip	userIdentit	userIdentit	userIdentit	userIdentit	requestEler	responseEl	eventType	eventID	errorCode	errorMessa	requestID	eventCateg	readOnly	re
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeVpcAttribute	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcId": "vpc-0d45b		AwsApiCal	025556ae-1525-4efd-8db6-b59			afc735ae-f	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeEgressOnlyInternetGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeEgressOnlyI		AwsApiCal	f513a426-8304-432a-afc1-e1d5			b6d7c9f4-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeVpcs	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcSet": {"items": [{'		AwsApiCal	d297b673-3d52-4202-8784-b9			79fea0a5-I	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeVpcAttribute	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcId": "vpc-0d45b		AwsApiCal	5ef0d5c1-449c-440d-848d-7e6e			e0024d6d-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeNatGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeNatGateway		AwsApiCal	4a9db82f-6a38-47a2-9137-671			99780b77	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeRouteTables	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"maxResults": 100, "r		AwsApiCal	f7dc122a-82ee-4f73-b1ab-8deff			3c40ad76-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeVpcEndpoints	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeVpcEndpoir		AwsApiCal	87a13b4c-683a-4f81-b92a-e32a			12817f7a-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeNatGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeNatGateway		AwsApiCal	757dc6b2-b5da-4ea5-9e12-4e8			241fbbcc-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeVpcs	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcSet": {"items": [{'		AwsApiCal	45038041-89b1-4f57-82c9-2ff0			af623cc9-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeVpcEndpoints	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeVpcEndpoir		AwsApiCal	6aee5d18-f76f-4fc8-b4ef-d7c02			c2f9e6f0-(	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeEgressOnlyInternetGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeEgressOnlyI		AwsApiCal	38c200e6-2e95-43da-a80a-433			adcbd784-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeVpcs	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcSet": {"items": [{'		AwsApiCal	6eadedf7-7ad6-46bc-b4c9-5ba6			e00b79dd-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeVpcEndpoints	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeVpcEndpoir		AwsApiCal	5c6fc4c6-3489-4b85-aade-14bc			2ac1678c-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeVpcAttribute	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcId": "vpc-0d45b		AwsApiCal	3c4a452e-433b-4bf8-9fc4-16d5			1786202a-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeVpcs	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcSet": {"items": [{'		AwsApiCal	c3b9f880-4d89-49fb-8419-fe90			af41b74e-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeEgressOnlyInternetGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeEgressOnlyI		AwsApiCal	95aee1ae-9b89-4126-b72f-d35e			014de6ff-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeNetworkAcls	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"maxResults": 1000,		AwsApiCal	1b865dc1-bde0-4334-84d6-fbe			72f1e7e7-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeInternetGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"internetGatewayIdSe		AwsApiCal	aef72562-43e5-4113-90fb-8e5b			ac4e3173-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeSubnets	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"subnetSet": {}, "filter		AwsApiCal	4a7cb70e-fecb-4b65-84c2-28f6			80d220ac-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeNetworkAcls	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"maxResults": 1000,		AwsApiCal	5bf921a3-cdcf-4497-86d0-fd79			d05558ac-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeInternetGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"internetGatewayIdSe		AwsApiCal	9b5476b7-ad5b-4d30-9efc-5e8			19048bb9	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeAccountAttributes	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"accountAttributeNar		AwsApiCal	3189bdb5-8127-41f1-9741-4f5			eaf69853-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeNatGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeNatGateway		AwsApiCal	2a57aee9-5317-43e3-9fc6-8b9			d6b63dec-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeVpcAttribute	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcId": "vpc-0d45b		AwsApiCal	3d23ccb8-51bf-46a2-9440-d2d			06518523	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeRouteTables	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"routeTableIdSet": {},		AwsApiCal	d05a8f7d-35da-4ec8-890b-c29c			5890d1a9-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeRouteTables	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"maxResults": 100, "r		AwsApiCal	25fe59f0-f8ec-49fb-b395-c3211			714fda5d-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeInternetGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"internetGatewayIdSe		AwsApiCal	cb29350c-5606-4009-8d96-f50			03a3a537-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeRouteTables	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"routeTableIdSet": {},		AwsApiCal	53d679f6-8783-4526-ad49-cde			249789c2-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeRouteTables	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"maxResults": 100, "r		AwsApiCal	b67a5dc3-a2e9-4a31-82e0-249			79d52266	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeSubnets	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"subnetSet": {}, "filter		AwsApiCal	ad020f4c-dfb0-4567-9605-4e9c			e912c695-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeNatGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeNatGateway		AwsApiCal	0c6eb211-468a-4d54-a0a3-2ca			8cdc7412-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeEgressOnlyInternetGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeEgressOnlyI		AwsApiCal	0aa8b78d-a9fe-44c4-b8ca-5025			29879be7-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeSubnets	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"subnetSet": {}, "filter		AwsApiCal	1bcb70b6-cfd8-427f-a689-6d1b			2751bb6b	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeRouteTables	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"maxResults": 100, "r		AwsApiCal	3da5aef0-ef41-4fca-b92f-9fe47			77b50c83-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeInternetGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"internetGatewayIdSe		AwsApiCal	8582c75a-f043-4be8-8266-5f8f			d164cf2f-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeVpcEndpoints	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeVpcEndpoir		AwsApiCal	95d381a0-7f84-4f3b-a847-fe87			7f60f391-	Manageme	TRUE	
2025-10-01T10:30:35Z	2025-10-01T19:30:35	ap-northeast-2	DescribeVpcAttribute	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcId": "vpc-0d45b		AwsApiCal	dfd5430f-49b8-4006-a3b7-cdd			17e33438-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeEgressOnlyInternetGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeEgressOnlyI		AwsApiCal	d5b98dcb-9ec3-4fda-87a3-693b			86d966e5-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeVpcAttribute	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcId": "vpc-0d45b		AwsApiCal	5eb64d85-c3b6-460b-b65d-4a3			d16235ea-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeNatGateways	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"DescribeNatGateway		AwsApiCal	db1baf67-5caa-4b6b-83f6-5b4c			f27c0f74-	Manageme	TRUE	
2025-10-01T10:30:36Z	2025-10-01T19:30:36	ap-northeast-2	DescribeVpcs	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTN		Root	2.31307E+11	arn:aws:iar	2.3E+11	ASIATLWX	{"attributes	{"vpcSet": {"items": [{'		AwsApiCal	2e46e0e8-9d39-4511-b104-602			6151955b	Manageme	TRUE	

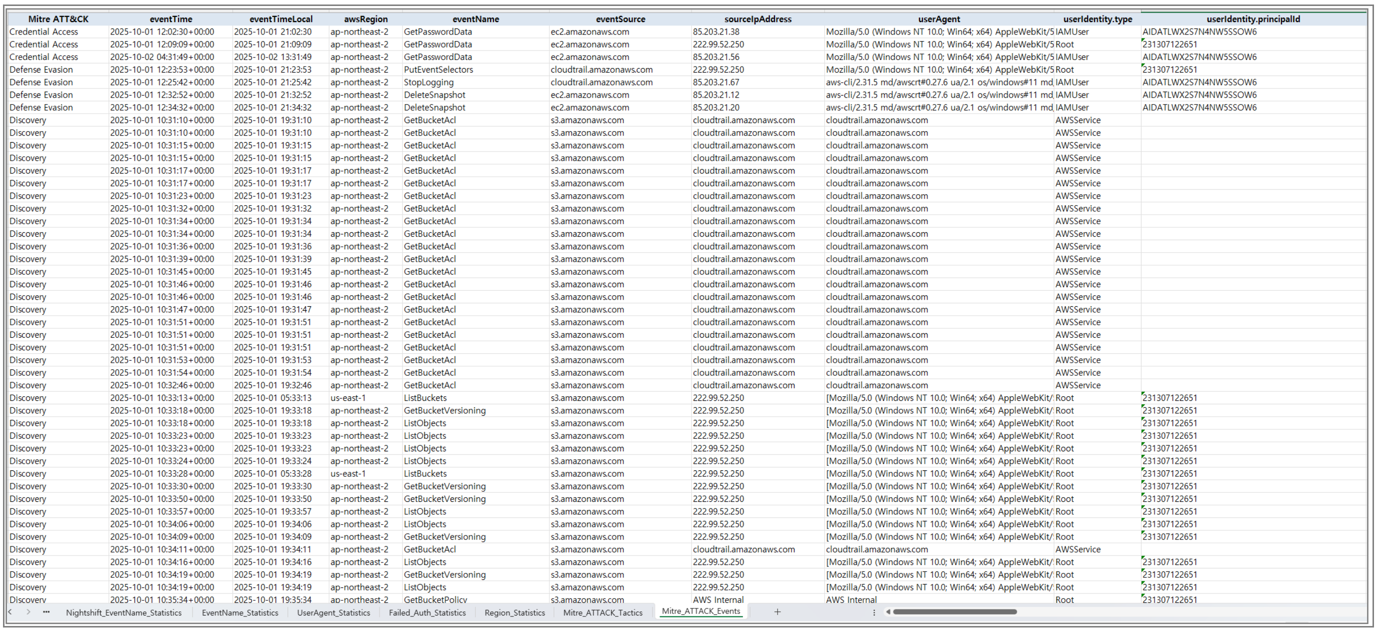
[Figure9] Example screen of original log parsing results (CloudTrail Log)



Source IP	Destination IP	Port	Protocol	Service	Total Bytes	Total Packets
85.203.21.4	172.31.34.5	3389	TCP	RDP	144579	1271
85.203.21.4	172.31.34.5	3389	UDP	RDP	3780	3
3.149.59.26	172.31.34.5	3389	TCP	RDP	2056	25
20.64.105.251	172.31.34.5	3389	TCP	RDP	1009	14
3.86.50.115	172.31.34.5	3389	TCP	RDP	772	8
125.142.157.171	172.31.34.5	3389	TCP	RDP	720	18
78.128.114.130	172.31.34.5	3389	TCP	RDP	400	10
78.128.114.126	172.31.34.5	3389	TCP	RDP	360	9
38.156.75.247	172.31.34.5	2222	TCP	SSH Alt	240	4
54.144.248.116	172.31.34.5	22	TCP	SSH/SCP/SFTP	240	4
91.231.89.234	172.31.34.5	5986	TCP	WinRM HTTPS	60	1
93.115.123.69	172.31.34.5	23	TCP	Telnet	60	1
206.168.35.195	172.31.34.5	5901	TCP	VNC Display 1	60	1
212.36.28.254	172.31.34.5	23	TCP	Telnet	60	1
206.168.35.31	172.31.34.5	135	TCP	RPC Endpoint Mapper	60	1
206.168.35.59	172.31.34.5	22222	TCP	SSH Alt	60	1
206.168.35.48	172.31.34.5	990	TCP	FTP over TLS	60	1
206.168.35.189	172.31.34.5	21	TCP	FTP	60	1
199.45.154.187	172.31.34.5	990	TCP	FTP over TLS	60	1
206.168.35.22	172.31.34.5	5902	TCP	VNC Display 2	60	1
42.112.116.49	172.31.34.5	23	TCP	Telnet	60	1

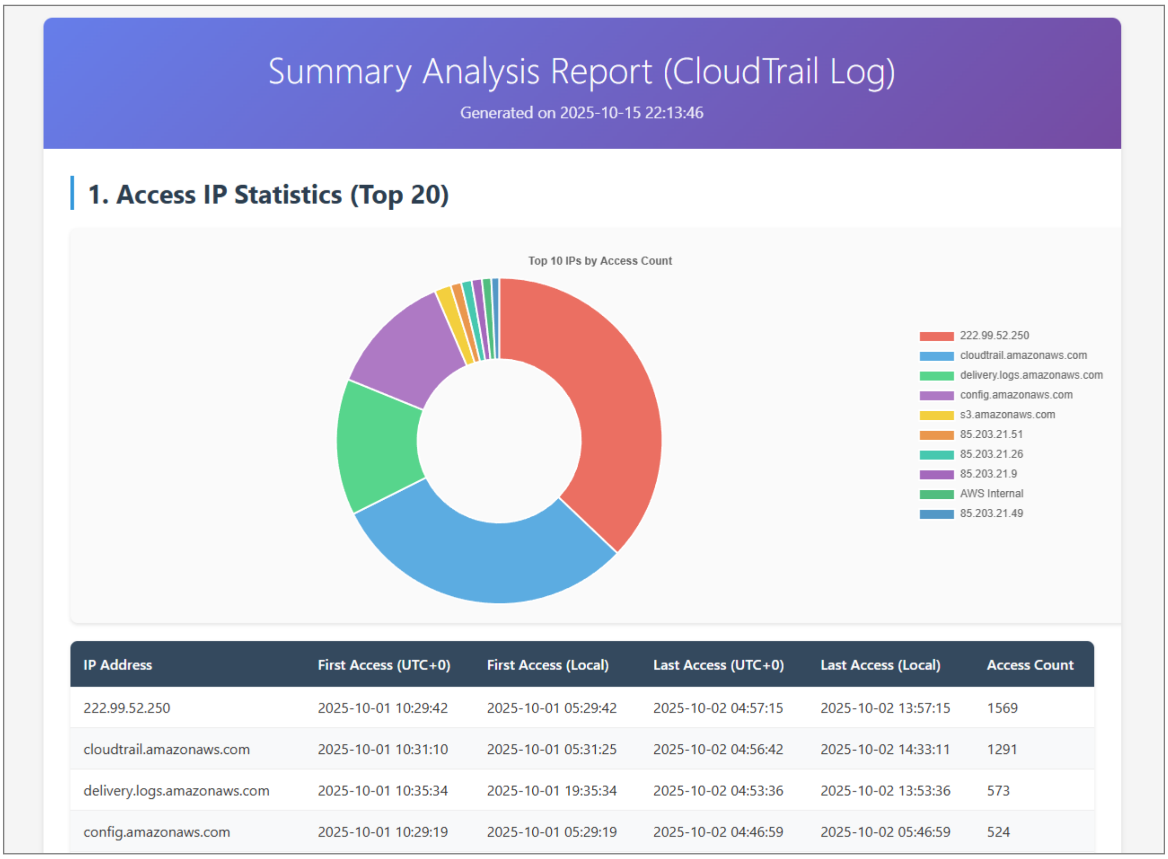
Access_IP_Statistics_(Top_20)
Port_Protocol_Statistics
Nightshift_Remote_Access_Events
Session_Duration_Statistics
Connection_Statistics_(Top_20)
+

[Figure10] Example log analysis results sheet (VPC Flow Logs – Remote access events occurring during overnight hours)



Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.principalId
Credential Access	2025-10-01 12:02:30+00:00	2025-10-01 21:02:30	ap-northeast-2	GetPasswordData	ec2.amazonaws.com	85.203.21.38	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5	IAMUser	AIDATLWX2S7N4NW5SSOW6
Credential Access	2025-10-01 12:09:09+00:00	2025-10-01 21:09:09	ap-northeast-2	GetPasswordData	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5	Root	231307122651
Credential Access	2025-10-02 04:31:49+00:00	2025-10-02 13:31:49	ap-northeast-2	GetPasswordData	ec2.amazonaws.com	85.203.21.56	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5	IAMUser	AIDATLWX2S7N4NW5SSOW6
Defense Evasion	2025-10-01 12:23:53+00:00	2025-10-01 21:23:53	ap-northeast-2	PutEventSelectors	cloudtrail.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5	Root	231307122651
Defense Evasion	2025-10-01 12:25:42+00:00	2025-10-01 21:25:42	ap-northeast-2	StopLogging	cloudtrail.amazonaws.com	85.203.21.67	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md	IAMUser	AIDATLWX2S7N4NW5SSOW6
Defense Evasion	2025-10-01 12:32:52+00:00	2025-10-01 21:32:52	ap-northeast-2	DeleteSnapshot	ec2.amazonaws.com	85.203.21.12	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md	IAMUser	AIDATLWX2S7N4NW5SSOW6
Defense Evasion	2025-10-01 12:34:32+00:00	2025-10-01 21:34:32	ap-northeast-2	DeleteSnapshot	ec2.amazonaws.com	85.203.21.20	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md	IAMUser	AIDATLWX2S7N4NW5SSOW6
Discovery	2025-10-01 10:31:10+00:00	2025-10-01 19:31:10	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:10+00:00	2025-10-01 19:31:10	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:15+00:00	2025-10-01 19:31:15	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:15+00:00	2025-10-01 19:31:15	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:17+00:00	2025-10-01 19:31:17	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:17+00:00	2025-10-01 19:31:17	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:23+00:00	2025-10-01 19:31:23	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:32+00:00	2025-10-01 19:31:32	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:34+00:00	2025-10-01 19:31:34	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:34+00:00	2025-10-01 19:31:34	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:36+00:00	2025-10-01 19:31:36	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:39+00:00	2025-10-01 19:31:39	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:45+00:00	2025-10-01 19:31:45	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:46+00:00	2025-10-01 19:31:46	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:46+00:00	2025-10-01 19:31:46	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:47+00:00	2025-10-01 19:31:47	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:51+00:00	2025-10-01 19:31:51	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:51+00:00	2025-10-01 19:31:51	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:51+00:00	2025-10-01 19:31:51	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:53+00:00	2025-10-01 19:31:53	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:31:54+00:00	2025-10-01 19:31:54	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:32:46+00:00	2025-10-01 19:32:46	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:33:13+00:00	2025-10-01 05:33:13	us-east-1	ListBuckets	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:33:18+00:00	2025-10-01 19:33:18	ap-northeast-2	GetBucketVersioning	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:33:18+00:00	2025-10-01 19:33:18	ap-northeast-2	ListObjects	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:33:23+00:00	2025-10-01 19:33:23	ap-northeast-2	ListObjects	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:33:23+00:00	2025-10-01 19:33:23	ap-northeast-2	ListObjects	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:33:24+00:00	2025-10-01 19:33:24	ap-northeast-2	ListObjects	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:33:28+00:00	2025-10-01 05:33:28	us-east-1	ListBuckets	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:33:30+00:00	2025-10-01 19:33:30	ap-northeast-2	GetBucketVersioning	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:33:50+00:00	2025-10-01 19:33:50	ap-northeast-2	GetBucketVersioning	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:33:57+00:00	2025-10-01 19:33:57	ap-northeast-2	ListObjects	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:34:06+00:00	2025-10-01 19:34:06	ap-northeast-2	ListObjects	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:34:09+00:00	2025-10-01 19:34:09	ap-northeast-2	GetBucketVersioning	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:34:11+00:00	2025-10-01 19:34:11	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.com	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 10:34:16+00:00	2025-10-01 19:34:16	ap-northeast-2	ListObjects	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:34:19+00:00	2025-10-01 19:34:19	ap-northeast-2	GetBucketVersioning	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:34:19+00:00	2025-10-01 19:34:19	ap-northeast-2	ListObjects	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/	Root	231307122651
Discovery	2025-10-01 10:35:34+00:00	2025-10-01 19:35:34	ap-northeast-2	GetBucketPolicy	s3.amazonaws.com	AWS Internal	AWS Internal	Root	231307122651

[Figure11] Example log analysis results sheet screen (CloudTrail Log – Event detection logs based on DFIR Cheat Sheet)



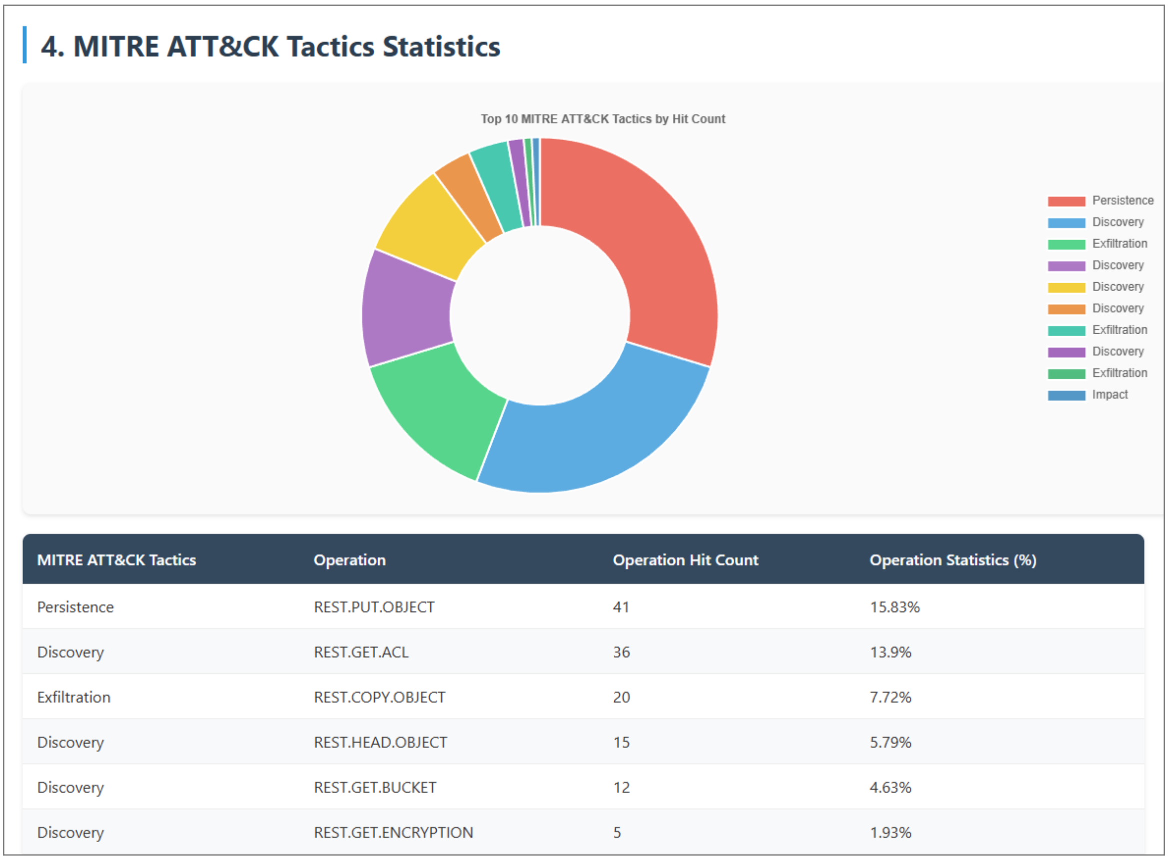
Summary Analysis Report (CloudTrail Log)

Generated on 2025-10-15 22:13:46

1. Access IP Statistics (Top 20)

IP Address	First Access (UTC+0)	First Access (Local)	Last Access (UTC+0)	Last Access (Local)	Access Count
222.99.52.250	2025-10-01 10:29:42	2025-10-01 05:29:42	2025-10-02 04:57:15	2025-10-02 13:57:15	1569
cloudtrail.amazonaws.com	2025-10-01 10:31:10	2025-10-01 05:31:25	2025-10-02 04:56:42	2025-10-02 14:33:11	1291
delivery.logs.amazonaws.com	2025-10-01 10:35:34	2025-10-01 19:35:34	2025-10-02 04:53:36	2025-10-02 13:53:36	573
config.amazonaws.com	2025-10-01 10:29:19	2025-10-01 05:29:19	2025-10-02 04:46:59	2025-10-02 05:46:59	524

[Figure12] Example Summary Report Screen of Analysis Results (CloudTrail Log)



4. MITRE ATT&CK Tactics Statistics

MITRE ATT&CK Tactics	Operation	Operation Hit Count	Operation Statistics (%)
Persistence	REST.PUT.OBJECT	41	15.83%
Discovery	REST.GET.ACL	36	13.9%
Exfiltration	REST.COPY.OBJECT	20	7.72%
Discovery	REST.HEAD.OBJECT	15	5.79%
Discovery	REST.GET.BUCKET	12	4.63%
Discovery	REST.GET.ENCRYPTION	5	1.93%

[Figure13] Sample Summary Report Screen for Analysis Results (S3 Access Log)

6. Scenario-Based Empirical Analysis

Based on attack tactics analyzed in prior research, we developed ransomware incident scenarios that could occur in an AWS cloud environment. We examined how to analyze key logs when ransomware strikes an AWS cloud environment. Furthermore, we analyzed what information can be obtained using the AWS DFIR log analysis tool (bitParser for AWS Log, hereafter bitParser) developed through this research. The results are as follows.

Chapter 6 covers the following content.

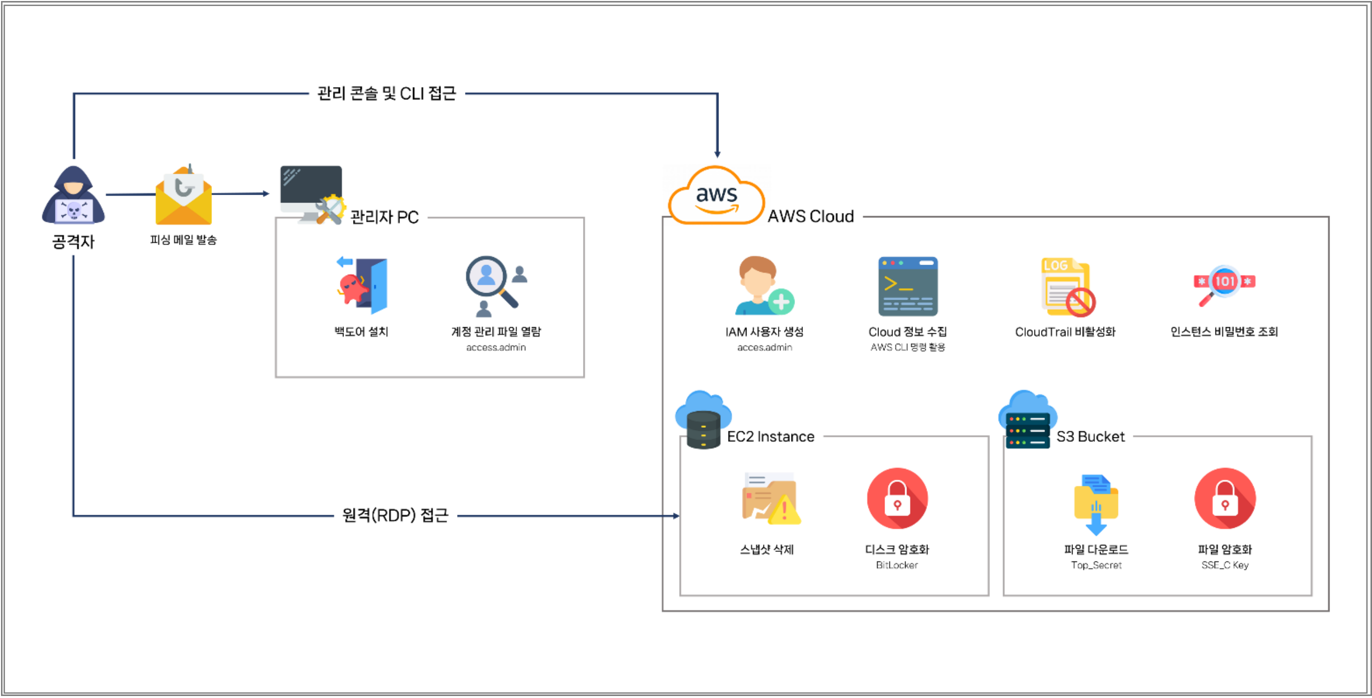
[Table148] Key Research Content – Scenario-Based Empirical Analysis

Number	Subtitle	Key Content
1	Attack Scenario Overview	Overview of AWS ransomware scenarios and explanation of attack tactics based on tactical analysis of prior research
2	Scenario Analysis Results	Presentation of key log analysis results and verification using the bitParser tool based on the ransomware scenario

6.1. Attack Scenario Overview

The attacker infiltrated an administrator PC infected with a backdoor via a phishing email, obtaining AWS IAM account credentials and key pair files (PEM) present within the account management file. Using the acquired AWS IAM account credentials, the attacker accessed the AWS Cloud console to create an IAM user. Through this newly created IAM user, the attacker collected Cloud information using AWS CLI commands. Additionally, the attacker obtained the EC2 Instance password using the pre-acquired key pair file (PEM).

Subsequently, the attacker deleted the EC2 instance snapshot, then remotely accessed it via RDP to encrypt the disk. Using AWS CLI commands, they downloaded files from the S3 bucket and encrypted internal files using the attacker's SSE_C Key.



[Figure14] Attack Scenario Overview Diagram

The attack actions performed by the attacker can be categorized by tactics as follows.

[Table149] Attack Actions Categorized by MITRE ATT&CK Tactics

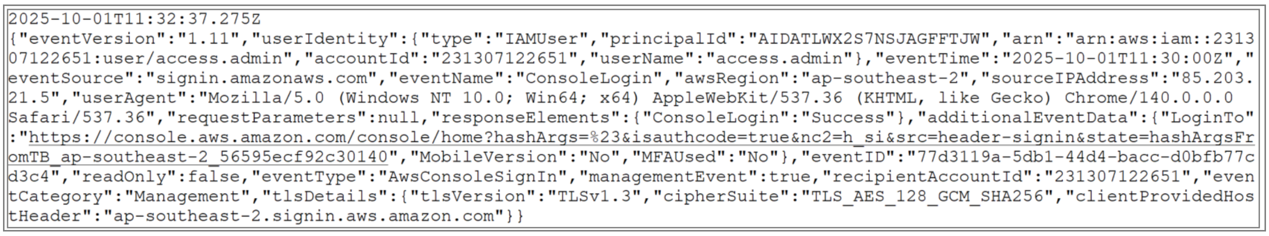
Tactics	Attack Technique Description
Initial Access (Initial Penetration)	• Installation of a backdoor via phishing email followed by penetration
Discovery & Collection (Information Gathering)	• Accessing account management files • Stealing key pair files (PEM) • Accessing the AWS Cloud Console to check EC2, S3, etc. • Information Gathering via AWS CLI
Persistence (Persistence)	• Create an attack-specific IAM user (access.admin) • Link AWS CLI account
Lateral Movement (Internal Movement)	• Remote Access to AWS EC2 Instances (RDP)
Defense Evasion (Defense Evasion)	• Disabling AWS CloudTrail
Impact (Impact)	• Deletion of AWS EC2 Instance Snapshot Data • AWS EC2 Instance Disk Encryption • Downloading AWS S3 Bucket Data • AWS S3 Bucket Data Encryption

6.2. Scenario Analysis Results

To examine how to analyze ransomware incidents in the AWS cloud environment, we collected and analyzed key logs (CloudTrail, VPC Flow Logs, S3 Access Log) and identified threats through the following events.

1) AWS Console login using credentials obtained from an administrator's PC

CloudTrail confirmed that the administrator's IAM user (access.admin) successfully logged into the AWS Console via the Chrome browser from the IP address 85.203.21.5 (Singapore) without MFA.



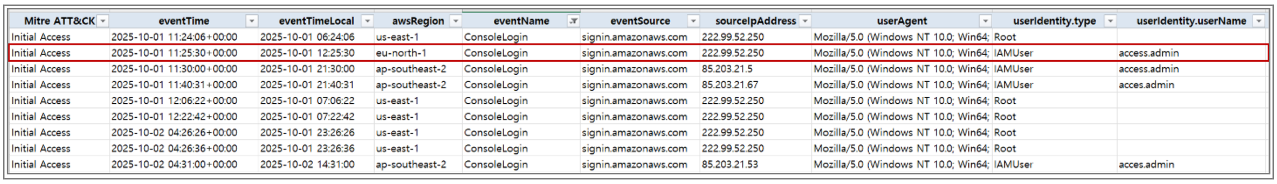
```
2025-10-01T11:32:37.275Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7NSJAGFFTJW","arn":"arn:aws:iam::2313
07122651:user/access.admin","accountId":"231307122651","userName":"access.admin"},"eventTime":"2025-10-01T11:30:00Z","
eventSource":"signin.amazonaws.com","eventName":"ConsoleLogin","awsRegion":"ap-southeast-2","sourceIPAddress":"85.203.
21.5","userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0
Safari/537.36","requestParameters":null,"responseElements":{"ConsoleLogin":"Success"},"additionalEventData":{"LoginTo"
:"https://console.aws.amazon.com/console/home?hashArgs=%23&isauthcode=true&nc2=h_si&src=header-signin&state=hashArgsFr
omTB_ap-southeast-2_56595ecf92c30140","MobileVersion":"No","MFAUsed":"No"},"eventID":"77d3119a-5db1-44d4-bacc-d0bfb77c
d3c4","readOnly":false,"eventType":"AwsConsoleSignIn","managementEvent":true,"recipientAccountId":"231307122651","even
tCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHos
tHeader":"ap-southeast-2.signin.aws.amazon.com"}}
```

[Figure15] AWS Console login event visible in CloudTrail

[Table150] Key field details of the AWS Console login event

Field	Key Field Details
AWS Console Login	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/access.admin - userName: access.admin • eventTime: 2025-10-01T11:30:00Z • eventSource: signin.amazonaws.com • eventName: ConsoleLogin • awsRegion: ap-southeast-2 • sourceIPAddress: 85.203.21.49 • userAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 • responseElements - ConsoleLogin: Success • additionalEventData - MobileVersion: No - MFAUsed: No

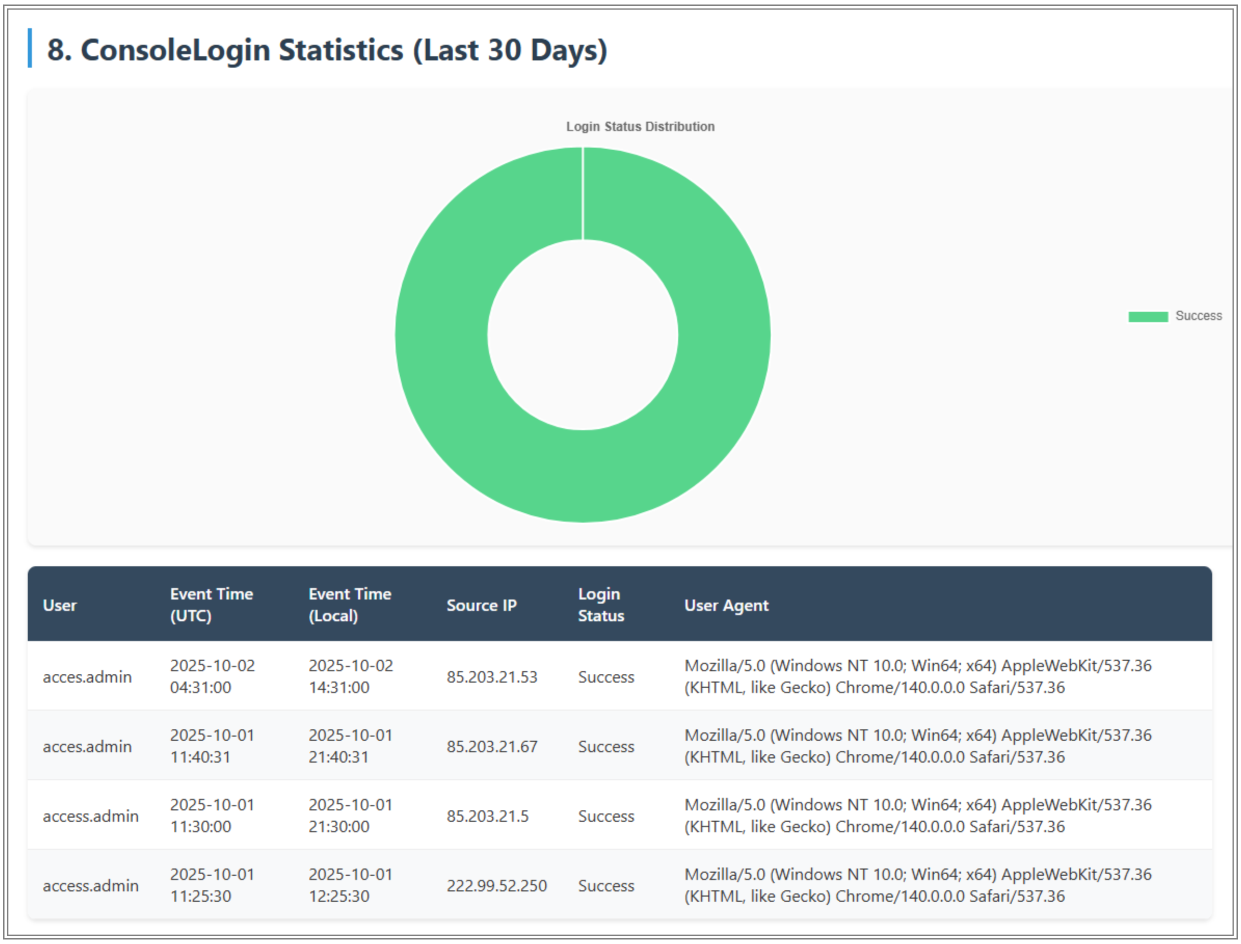
bitParser analysis results also confirm this behavior in the file as follows.



Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Initial Access	2025-10-01 11:24:06+00:00	2025-10-01 06:24:06	us-east-1	ConsoleLogin	signin.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Initial Access	2025-10-01 11:25:30+00:00	2025-10-01 12:25:30	eu-north-1	ConsoleLogin	signin.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin
Initial Access	2025-10-01 11:30:00+00:00	2025-10-01 21:30:00	ap-southeast-2	ConsoleLogin	signin.amazonaws.com	85.203.21.5	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin
Initial Access	2025-10-01 11:40:31+00:00	2025-10-01 21:40:31	ap-southeast-2	ConsoleLogin	signin.amazonaws.com	85.203.21.67	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin
Initial Access	2025-10-01 12:06:22+00:00	2025-10-01 07:06:22	us-east-1	ConsoleLogin	signin.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Initial Access	2025-10-01 12:22:42+00:00	2025-10-01 07:22:42	us-east-1	ConsoleLogin	signin.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Initial Access	2025-10-02 04:26:26+00:00	2025-10-01 23:26:26	us-east-1	ConsoleLogin	signin.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Initial Access	2025-10-02 04:26:36+00:00	2025-10-01 23:26:36	us-east-1	ConsoleLogin	signin.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Initial Access	2025-10-02 04:31:00+00:00	2025-10-02 14:31:00	ap-southeast-2	ConsoleLogin	signin.amazonaws.com	85.203.21.53	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin

[Figure16] AWS Console login event identified in the bitParser analysis result file

Additionally, you can identify whether IP addresses not previously accessed or external IP addresses exist by checking the 'ConsoleLogin Statistics' screen in the bitParser analysis summary report file. While the analysis sheet allows for a full history analysis of console logins, the report only displays the history for the last 30 days.



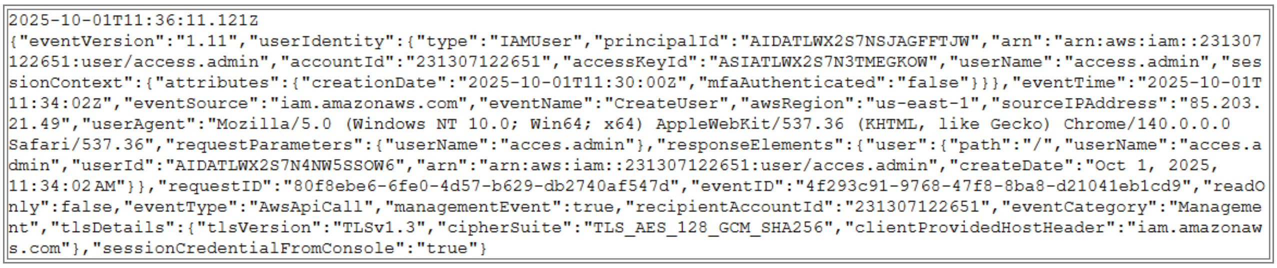
8. ConsoleLogin Statistics (Last 30 Days)

User	Event Time (UTC)	Event Time (Local)	Source IP	Login Status	User Agent
acces.admin	2025-10-02 04:31:00	2025-10-02 14:31:00	85.203.21.53	Success	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
acces.admin	2025-10-01 11:40:31	2025-10-01 21:40:31	85.203.21.67	Success	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
access.admin	2025-10-01 11:30:00	2025-10-01 21:30:00	85.203.21.5	Success	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
access.admin	2025-10-01 11:25:30	2025-10-01 12:25:30	222.99.52.250	Success	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36

[Figure17] 'ConsoleLogin Statistics' screen identified in the bitParser analysis summary report file

2) Creation of an AWS IAM user dedicated for attacks

CloudTrail showed that an IAM user (access.admin) created a new IAM user (access.admin) via the Chrome browser from the IP address 85.203.21.49 (Singapore).



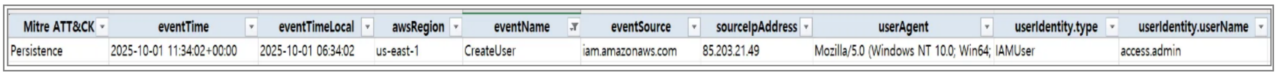
```
2025-10-01T11:36:11.121Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7NSJAGFFTJW","arn":"arn:aws:iam::231307
122651:user/access.admin","accountId":"231307122651","accessKeyId":"ASIATLWX2S7N3TMEGKOW","userName":"access.admin","ses
sionContext":{"attributes":{"creationDate":"2025-10-01T11:30:00Z","mfaAuthenticated":"false"}}},"eventTime":"2025-10-01T
11:34:02Z","eventSource":"iam.amazonaws.com","eventName":"CreateUser","awsRegion":"us-east-1","sourceIPAddress":"85.203.
21.49","userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0
Safari/537.36","requestParameters":{"userName":"acces.admin"},"responseElements":{"user":{"path":"/","userName":"acces.a
dmin","userId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307122651:user/acces.admin","createDate":"Oct 1, 2025,
11:34:02 AM"}},"requestID":"80f8ebe6-6fe0-4d57-b629-db2740af547d","eventID":"4f293c91-9768-47f8-8ba8-d21041eb1cd9","readO
nly":false,"eventType":"AwsApiCall","managementEvent":true,"recipientAccountId":"231307122651","eventCategory":"Manageme
nt","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"iam.amazonaw
s.com"},"sessionCredentialFromConsole":"true"}
```

[Figure18] AWS IAM user creation event observed in CloudTrail

[Table151] Key field details of the AWS IAM user creation event

Field	Key Field Details
AWS IAM User Creation	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/access.admin - userName: access.admin • eventTime: 2025-10-01T11:34:02Z • eventSource: iam.amazonaws.com • eventName: CreateUser • awsRegion: us-east-1 • sourceIPAddress: 85.203.21.49 • userAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 • responseElements - user:userName: acces.admin - user:userId: AIDATLWX2S7N4NW5SSOW6 - user:createDate: Oct 1, 2025, 11:34:02 AM

bitParser analysis results also confirm this activity in the file as follows.

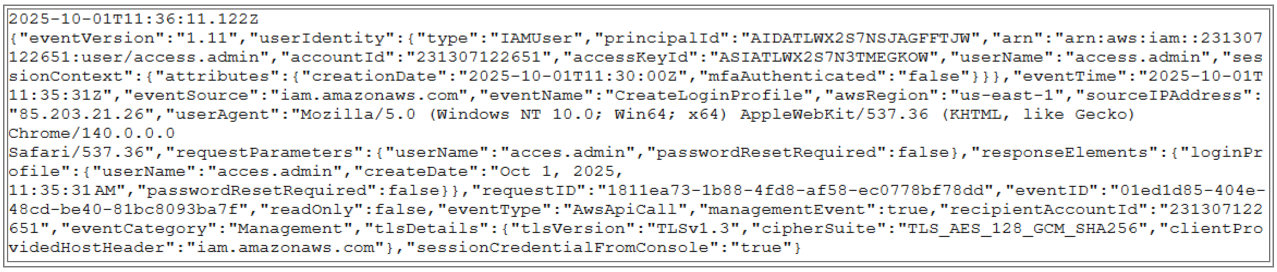


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Persistence	2025-10-01 11:34:02+00:00	2025-10-01 06:34:02	us-east-1	CreateUser	iam.amazonaws.com	85.203.21.49	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin

[Figure19] AWS IAM user creation event confirmed in the bitParser analysis result file

3) Enabling Console access for the attack-specific AWS IAM user

CloudTrail showed that the IAM user (access.admin) created a LoginProfile from the IP 85.203.21.26 (Singapore) via the Chrome browser to enable Console access for the IAM user (access.admin).



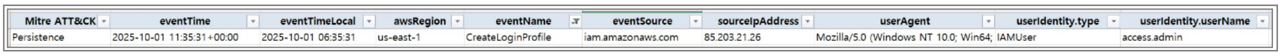
```
2025-10-01T11:36:11.122Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7NSJAGFFTJW","arn":"arn:aws:iam::231307
122651:user/access.admin","accountId":"231307122651","accessKeyId":"ASIATLWX2S7N3TMEGKOW","userName":"access.admin","ses
sionContext":{"attributes":{"creationDate":"2025-10-01T11:30:00Z","mfaAuthenticated":"false"}}},"eventTime":"2025-10-01T
11:35:31Z","eventSource":"iam.amazonaws.com","eventName":"CreateLoginProfile","awsRegion":"us-east-1","sourceIPAddress":
"85.203.21.26","userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/140.0.0.0
Safari/537.36","requestParameters":{"userName":"acces.admin","passwordResetRequired":false},"responseElements":{"loginPr
ofile":{"userName":"acces.admin","createDate":"Oct 1, 2025,
11:35:31 AM","passwordResetRequired":false}},"requestID":"1811ea73-1b88-4fd8-af58-ec0778bf78dd","eventID":"01ed1d85-404e-
48cd-be40-81bc8093ba7f","readOnly":false,"eventType":"AwsApiCall","managementEvent":true,"recipientAccountId":"231307122
651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientPro
videdHostHeader":"iam.amazonaws.com"},"sessionCredentialFromConsole":"true"}
```

[Figure20] AWS IAM User Console Access Enabled Event as seen in CloudTrail

[Table152] Key field details for AWS IAM user console access activation events

Field	Key Field Details
AWS IAM User Console Access Activation	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/access.admin - userName: access.admin • sessionContext - creationDate: 2025-10-01T11:30:00Z - mfaAuthenticated: false • eventTime: 2025-10-01T11:35:31Z • eventSource: iam.amazonaws.com • eventName: CreateLoginProfile • awsRegion: us-east-1 • sourceIPAddress: 85.203.21.26 • userAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 • responseElements - loginprofile:userName: acces.admin - loginprofile:createDate: Oct 1, 2025, 11:35:31 AM - loginprofile:passwordResetRequired: false

The bitParser analysis results file also confirms this activity as follows.

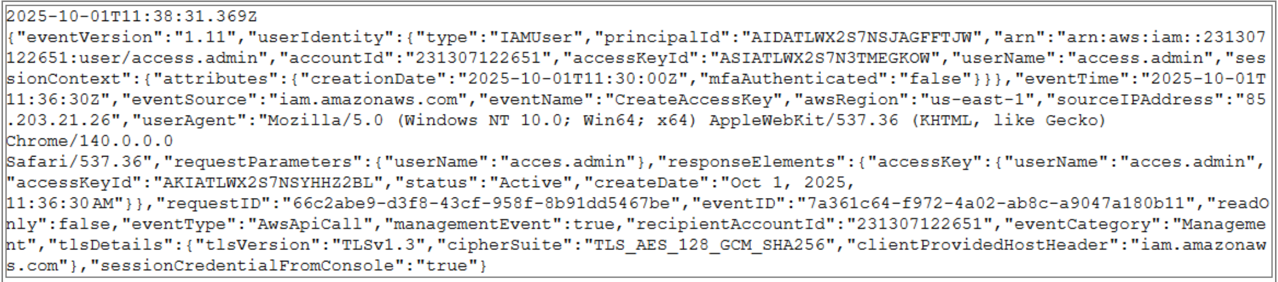


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Persistence	2025-10-01 11:35:31+00:00	2025-10-01 06:35:31	us-east-1	CreateLoginProfile	iam.amazonaws.com	85.203.21.26	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin

[Figure21] AWS IAM user Console access activation event confirmed in the bitParser analysis result file

4) Creation of an attack-specific AWS IAM user Access Key

CloudTrail confirmed that the IAM user (access.admin) created an Access Key for the IAM user (access.admin) via the Chrome browser from the IP address 85.203.21.26 (Singapore).



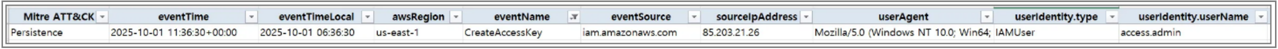
```
2025-10-01T11:38:31.369Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7NSJAGFFTJW","arn":"arn:aws:iam::231307
122651:user/access.admin","accountId":"231307122651","accessKeyId":"ASIATLWX2S7N3TMEGKOW","userName":"access.admin","ses
sionContext":{"attributes":{"creationDate":"2025-10-01T11:30:00Z","mfaAuthenticated":"false"}}},"eventTime":"2025-10-01T
11:36:30Z","eventSource":"iam.amazonaws.com","eventName":"CreateAccessKey","awsRegion":"us-east-1","sourceIPAddress":"85
.203.21.26","userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/140.0.0.0
Safari/537.36","requestParameters":{"userName":"acces.admin"},"responseElements":{"accessKey":{"userName":"acces.admin",
"accessKeyId":"AKIATLWX2S7NSYHHZ2BL","status":"Active","createDate":"Oct 1, 2025,
11:36:30 AM"}},"requestID":"66c2abe9-d3f8-43cf-958f-8b91dd5467be","eventID":"7a361c64-f972-4a02-ab8c-a9047a180b11","readO
nly":false,"eventType":"AwsApiCall","managementEvent":true,"recipientAccountId":"231307122651","eventCategory":"Manageme
nt","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"iam.amazonaw
s.com"},"sessionCredentialFromConsole":"true"}
```

[Figure22] AWS IAM user Access Key creation event observed in CloudTrail

[Table153] Key fields in the AWS IAM user Access Key creation event

Field	Key Field Details
AWS IAM User Access Key Creation	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/access.admin - userName: access.admin • sessionContext - creationDate: 2025-10-01T11:30:00Z - mfaAuthenticated: false • eventTime: 2025-10-01T11:36:30Z • eventSource: iam.amazonaws.com • eventName: CreateAccessKey • awsRegion: us-east-1 • sourceIPAddress: 85.203.21.26 • userAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 • responseElements - userName: acces.admin - accessKeyId: AKIATLWX2S7NSYHHZ2BL - status: Active - createDate: Oct 1, 2025, 11:36:30 AM

The bitParser analysis results also confirm this behavior in the file as follows.

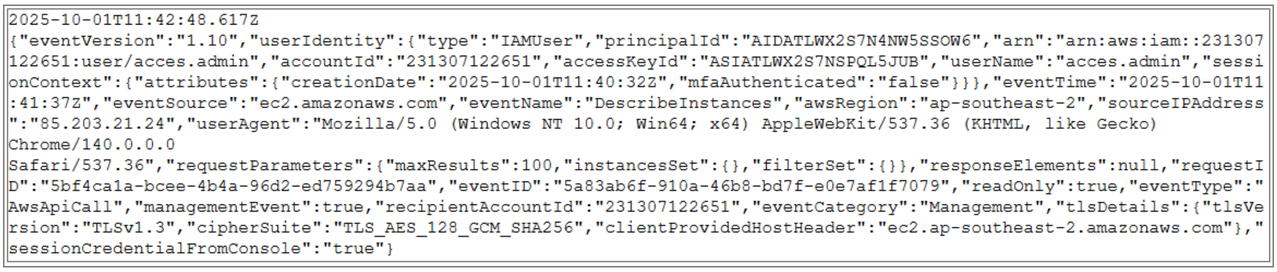


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Persistence	2025-10-01 11:36:30+00:00	2025-10-01 06:36:30	us-east-1	CreateAccessKey	iam.amazonaws.com	85.203.21.26	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin

[Figure23] AWS IAM user Access Key creation event identified in the file analyzed by bitParser

5) Accessing the EC2 page via the AWS Console

CloudTrail shows that the IAM user (acces.admin) accessed EC2 instance information via the Chrome browser from the IP address 85.203.21.24 (Singapore).



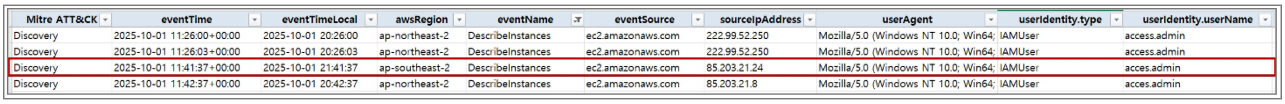
```
2025-10-01T11:42:48.617Z
{"eventVersion":"1.10","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"ASIATLWX2S7NSPQL5JUB","userName":"acces.admin","sessi
onContext":{"attributes":{"creationDate":"2025-10-01T11:40:32Z","mfaAuthenticated":"false"}}},"eventTime":"2025-10-01T11
:41:37Z","eventSource":"ec2.amazonaws.com","eventName":"DescribeInstances","awsRegion":"ap-southeast-2","sourceIPAddress
":"85.203.21.24","userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/140.0.0.0
Safari/537.36","requestParameters":{"maxResults":100,"instancesSet":{},"filterSet":{}},"responseElements":null,"requestI
D":"5bf4ca1a-bcee-4b4a-96d2-ed759294b7aa","eventID":"5a83ab6f-910a-46b8-bd7f-e0e7af1f7079","readOnly":true,"eventType":"
AwsApiCall","managementEvent":true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVe
rsion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"ec2.ap-southeast-2.amazonaws.com"},"
sessionCredentialFromConsole":"true"}
```

[Figure24] EC2 page access event via AWS Console observed in CloudTrail

[Table154] Key field details of the EC2 page access event via the AWS Console

Category	Key Field Details
EC2 page access via AWS Console EC2 Page Access	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • sessionContext - creationDate: 2025-10-01T11:40:32Z - mfaAuthenticated: false • eventTime: 2025-10-01T11:41:37Z • eventSource: ec2.amazonaws.com • eventName: DescribeInstances • awsRegion: ap-southeast-2 • sourceIPAddress: 85.203.21.24 • userAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36

The bitParser analysis results file also confirms this behavior as follows.

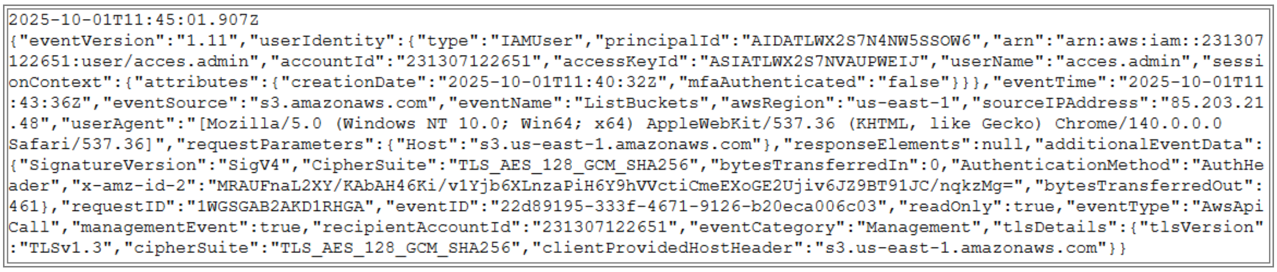


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 11:26:00+00:00	2025-10-01 20:26:00	ap-northeast-2	DescribeInstances	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin
Discovery	2025-10-01 11:26:03+00:00	2025-10-01 20:26:03	ap-northeast-2	DescribeInstances	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin
Discovery	2025-10-01 11:41:37+00:00	2025-10-01 21:41:37	ap-southeast-2	DescribeInstances	ec2.amazonaws.com	85.203.21.24	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin
Discovery	2025-10-01 11:42:37+00:00	2025-10-01 20:42:37	ap-northeast-2	DescribeInstances	ec2.amazonaws.com	85.203.21.8	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin

[Figure25] EC2 page access event via AWS Console confirmed in bitParser analysis results

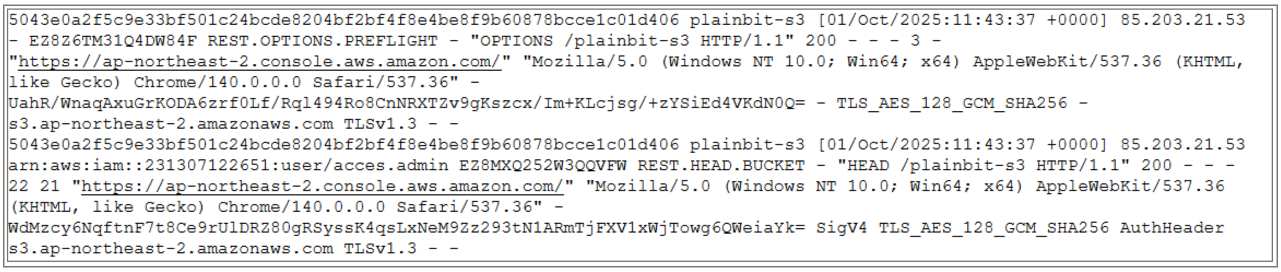
6) Accessing the S3 page via the AWS Console

CloudTrail and S3 Access Logs showed that the IAM user (acces.admin) accessed the S3 bucket list via the Chrome browser from the IP address 85.203.21.53 (Singapore).



```
2025-10-01T11:45:01.907Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"ASIATLWX2S7NVAUPWEIJ","userName":"acces.admin","sessi
onContext":{"attributes":{"creationDate":"2025-10-01T11:40:32Z","mfaAuthenticated":"false"}}},"eventTime":"2025-10-01T11
:43:36Z","eventSource":"s3.amazonaws.com","eventName":"ListBuckets","awsRegion":"us-east-1","sourceIPAddress":"85.203.21
.48","userAgent":"[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0
Safari/537.36]","requestParameters":{"Host":"s3.us-east-1.amazonaws.com"},"responseElements":null,"additionalEventData":
{"SignatureVersion":"SigV4","CipherSuite":"TLS_AES_128_GCM_SHA256","bytesTransferredIn":0,"AuthenticationMethod":"AuthHe
ader","x-amz-id-2":"MRAUFnaL2XY/KAbAH46Ki/v1Yjb6XLnzaPiH6Y9hVVctiCmeEXoGE2Ujiv6JZ9BT91JC/nqkzMg=","bytesTransferredOut":
461},"requestID":"1WGSGAB2AKD1RHGA","eventID":"22d89195-333f-4671-9126-b20eca006c03","readOnly":true,"eventType":"AwsApi
Call","managementEvent":true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion"
:"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"s3.us-east-1.amazonaws.com"}}
```

[Figure26] S3 page access event via AWS Console observed in CloudTrail



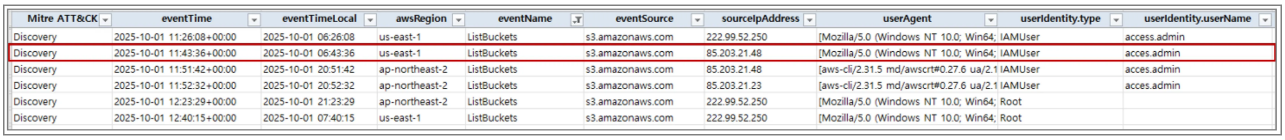
```
5043e0a2f5c9e33bf501c24bcde8204bf2bf4f8e4be8f9b60878bcce1c01d406 plainbit-s3 [01/Oct/2025:11:43:37 +0000] 85.203.21.53
- EZ8Z6TM31Q4DW84F REST.OPTIONS.PREFLIGHT - "OPTIONS /plainbit-s3 HTTP/1.1" 200 - - - 3 -
"https://ap-northeast-2.console.aws.amazon.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/140.0.0.0 Safari/537.36" -
UahR/WnaqAxuGrKODA6zrf0Lf/Rql494Ro8CnNRXTZv9gKszcx/Im+KLcjsg/+zYSiEd4VKdN0Q= - TLS_AES_128_GCM_SHA256 -
s3.ap-northeast-2.amazonaws.com TLSv1.3 - -
5043e0a2f5c9e33bf501c24bcde8204bf2bf4f8e4be8f9b60878bcce1c01d406 plainbit-s3 [01/Oct/2025:11:43:37 +0000] 85.203.21.53
arn:aws:iam::231307122651:user/acces.admin EZ8MXQ252W3QQVFW REST.HEAD.BUCKET - "HEAD /plainbit-s3 HTTP/1.1" 200 - - -
22 21 "https://ap-northeast-2.console.aws.amazon.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36" -
WdMzcy6NqftnF7t8Ce9rUlDRZ80gRSyssK4qsLxNeM9Zz293tN1ARmTjFXV1xWjTowg6QWeiaYk= SigV4 TLS_AES_128_GCM_SHA256 AuthHeader
s3.ap-northeast-2.amazonaws.com TLSv1.3 - -
```

[Figure27] S3 Access Log showing the AWS Console S3 page access event

[Table155] Key field details of S3 page access events via the AWS Console

Field	Key Field Details
(CloudTrail) Access to S3 pages via the AWS Console S3 Page Access	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • sessionContext - creationDate: 2025-10-01T11:40:32Z - mfaAuthenticated: false • eventTime: 2025-10-01T11:43:36Z • eventSource: s3.amazonaws.com • eventName: ListBuckets • awsRegion: us-east-1 • sourceIPAddress: 85.203.21.48 • userAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
(S3 Access Log) Via the AWS Console S3 page access	• eventTime: [01/Oct/2025 11:43:37 +0000] • sourceIPAddress: 85.203.21.53 • task: REST.OPTIONS.PREFLIGHT or REST.HEAD.BUCKET • StatusCode: 200 • User-Agent: https://ap-northeast-2.console.aws.amazon.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36

bitParser analysis results also confirm this behavior in the file as follows.

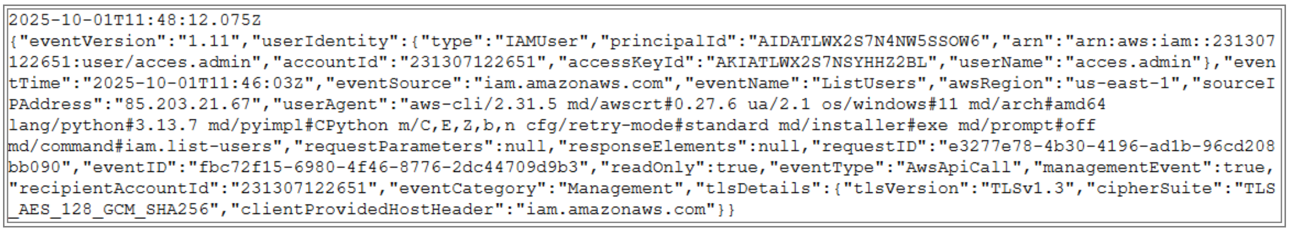


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 11:26:08+00:00	2025-10-01 06:26:08	us-east-1	ListBuckets	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin
Discovery	2025-10-01 11:43:36+00:00	2025-10-01 06:43:36	us-east-1	ListBuckets	s3.amazonaws.com	85.203.21.48	[Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin
Discovery	2025-10-01 11:51:42+00:00	2025-10-01 20:51:42	ap-northeast-2	ListBuckets	s3.amazonaws.com	85.203.21.48	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Discovery	2025-10-01 11:52:32+00:00	2025-10-01 20:52:32	ap-northeast-2	ListBuckets	s3.amazonaws.com	85.203.21.23	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Discovery	2025-10-01 12:23:29+00:00	2025-10-01 21:23:29	ap-northeast-2	ListBuckets	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Discovery	2025-10-01 12:40:15+00:00	2025-10-01 07:40:15	us-east-1	ListBuckets	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64;	Root	

[Figure28] S3 page access event via AWS Console confirmed in the bitParser analysis result file

7) Collection of IAM user list via AWS CLI command

CloudTrail shows that an IAM user (acces.admin) executed the 'iam list-users' command via AWS CLI from the IP 85.203.21.67 (Singapore) to collect the IAM user list.



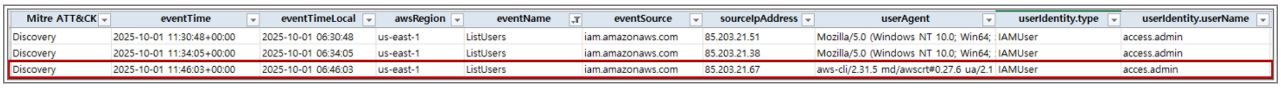
```
2025-10-01T11:48:12.075Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T11:46:03Z","eventSource":"iam.amazonaws.com","eventName":"ListUsers","awsRegion":"us-east-1","sourceI
PAddress":"85.203.21.67","userAgent":"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/C,E,Z,b,n cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#iam.list-users","requestParameters":null,"responseElements":null,"requestID":"e3277e78-4b30-4196-ad1b-96cd208
bb090","eventID":"fbc72f15-6980-4f46-8776-2dc44709d9b3","readOnly":true,"eventType":"AwsApiCall","managementEvent":true,
"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS
_AES_128_GCM_SHA256","clientProvidedHostHeader":"iam.amazonaws.com"}}
```

[Figure29] Event showing collection of IAM user list via AWS CLI command, as seen in CloudTrail

[Table156] Key field details of the IAM user list collection event via AWS CLI command

Field	Key Field Details
IAM user list collection via AWS CLI command IAM User List Collection	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T11:46:03Z • eventSource: iam.amazonaws.com • eventName: ListUsers • awsRegion: us-east-1 • sourceIPAddress: 85.203.21.67 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#Cpython m/C,E,Z,b,n cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#iam.list-users

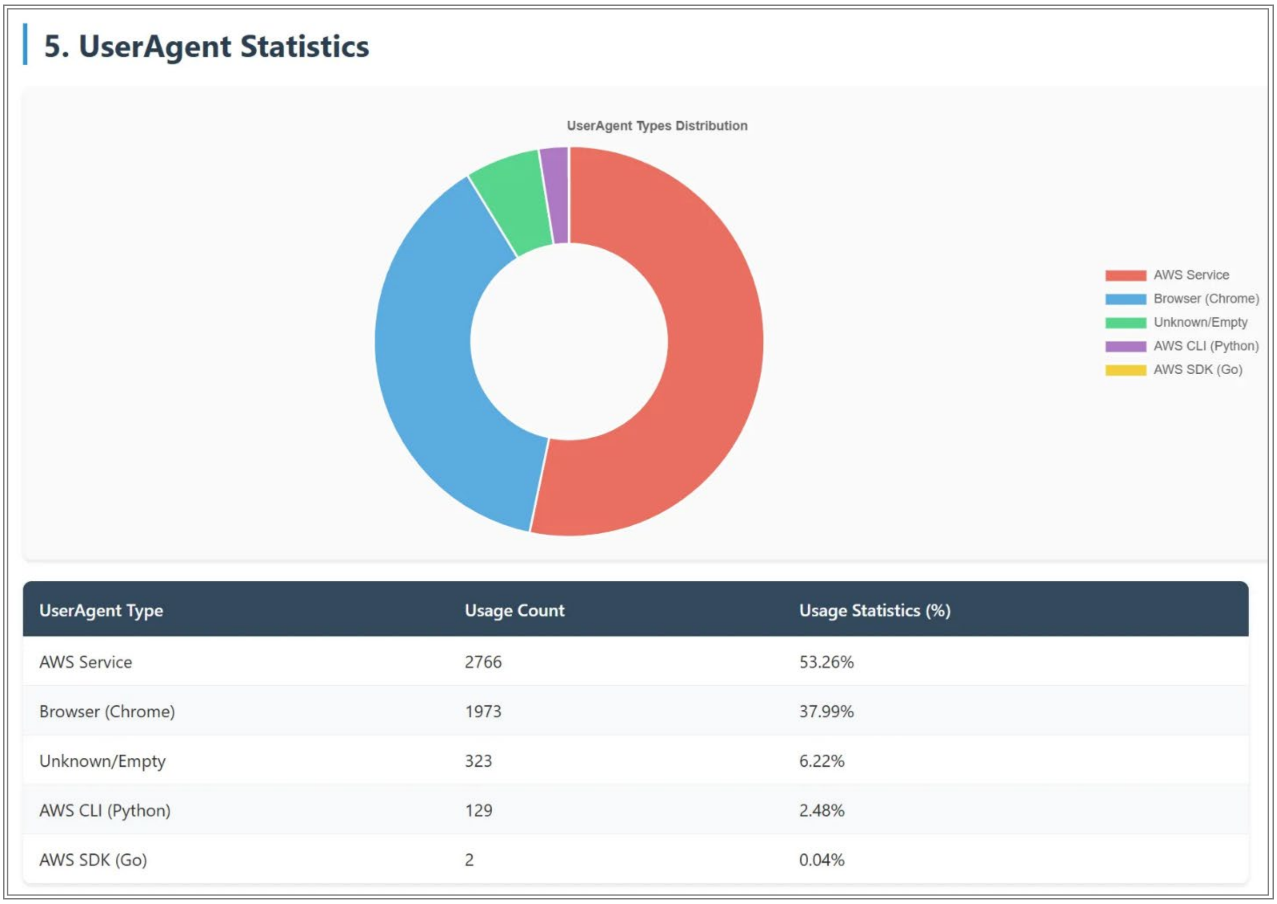
The bitParser analysis results file also confirms this behavior as follows.



Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 11:30:48+00:00	2025-10-01 06:30:48	us-east-1	ListUsers	iam.amazonaws.com	85.203.21.51	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin
Discovery	2025-10-01 11:34:05+00:00	2025-10-01 06:34:05	us-east-1	ListUsers	iam.amazonaws.com	85.203.21.38	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin
Discovery	2025-10-01 11:46:03+00:00	2025-10-01 06:46:03	us-east-1	ListUsers	iam.amazonaws.com	85.203.21.67	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin

[Figure30] IAM user list collection event identified in the bitParser analysis result file

Additionally, you can identify this by checking the 'UserAgent Statistics' screen in the bitParser analysis summary report file for any previously unused UserAgents.



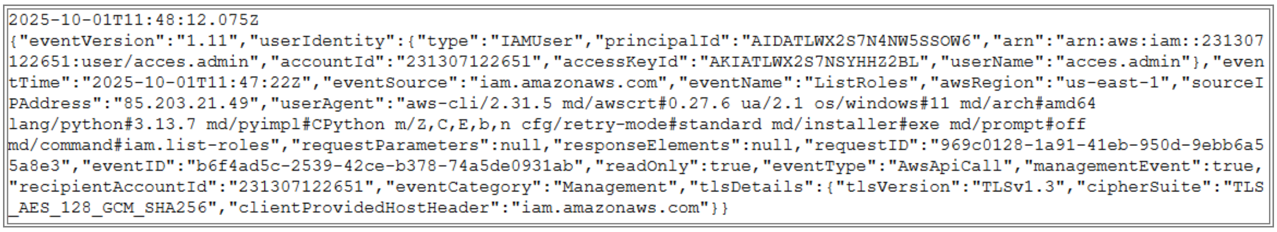
5. UserAgent Statistics

UserAgent Type	Usage Count	Usage Statistics (%)
AWS Service	2766	53.26%
Browser (Chrome)	1973	37.99%
Unknown/Empty	323	6.22%
AWS CLI (Python)	129	2.48%
AWS SDK (Go)	2	0.04%

[Figure31] 'UserAgent Statistics' screen identified in the bitParser analysis summary report file

8) Collecting IAM role list via AWS CLI command

CloudTrail confirmed that an IAM user (acces.admin) collected the IAM role list by executing the 'iam list-roles' command via AWS CLI from the IP address 85.203.21.49 (Singapore).



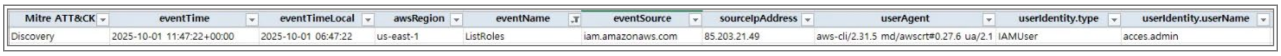
```
2025-10-01T11:48:12.075Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T11:47:22Z","eventSource":"iam.amazonaws.com","eventName":"ListRoles","awsRegion":"us-east-1","sourceI
PAddress":"85.203.21.49","userAgent":"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/Z,C,E,b,n cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#iam.list-roles","requestParameters":null,"responseElements":null,"requestID":"969c0128-1a91-41eb-950d-9ebb6a5
5a8e3","eventID":"b6f4ad5c-2539-42ce-b378-74a5de0931ab","readOnly":true,"eventType":"AwsApiCall","managementEvent":true,
"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS
_AES_128_GCM_SHA256","clientProvidedHostHeader":"iam.amazonaws.com"}}
```

[Figure32] Event showing collection of IAM role list via AWS CLI command in CloudTrail

[Table157] Key field details of the IAM role list collection event via AWS CLI command

Category	Key Field Details
Collection of IAM Role List via AWS CLI Command Collecting IAM Role List	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T11:47:22Z • eventSource: iam.amazonaws.com • eventName: ListRoles • awsRegion: us-east-1 • sourceIPAddress: 85.203.21.49 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/Z,C,E,b,n cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#iam.list-roles

The bitParser analysis results file also confirms this behavior as follows.

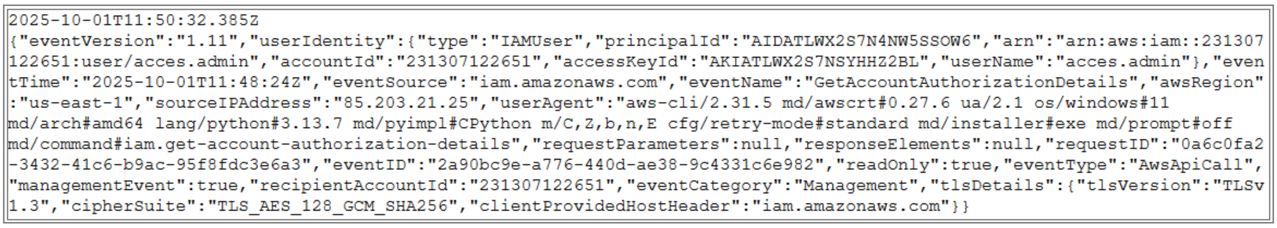


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 11:47:22+00:00	2025-10-01 06:47:22	us-east-1	ListRoles	iam.amazonaws.com	85.203.21.49	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin

[Figure33] IAM role collection event identified in the bitParser analysis result file

9) Collecting IAM information via AWS CLI commands

CloudTrail shows that the IAM user (acces.admin) executed the 'iam get-account-authorization-details' command via AWS CLI from the IP 85.203.21.25 (Singapore) to collect IAM information.



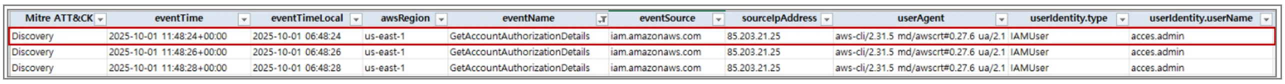
```
2025-10-01T11:50:32.385Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T11:48:24Z","eventSource":"iam.amazonaws.com","eventName":"GetAccountAuthorizationDetails","awsRegion"
:"us-east-1","sourceIPAddress":"85.203.21.25","userAgent":"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11
md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/C,Z,b,n,E cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#iam.get-account-authorization-details","requestParameters":null,"responseElements":null,"requestID":"0a6c0fa2
-3432-41c6-b9ac-95f8fdc3e6a3","eventID":"2a90bc9e-a776-440d-ae38-9c4331c6e982","readOnly":true,"eventType":"AwsApiCall",
"managementEvent":true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv
1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"iam.amazonaws.com"}}
```

[Figure34] IAM information collection event via AWS CLI command observed in CloudTrail

[Table158] Key field details of the IAM information collection event via AWS CLI command

Field	Key Field Details
IAM information collection via AWS CLI command IAM Information Collection	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T11:48:24Z • eventSource: iam.amazonaws.com • eventName: GetAccountAuthorizationDetails • awsRegion: us-east-1 • sourceIPAddress: 85.203.21.25 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/C,Z,b,n,E cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#iam.get-account-authorization-details

The bitParser analysis results file also confirms this behavior as follows.

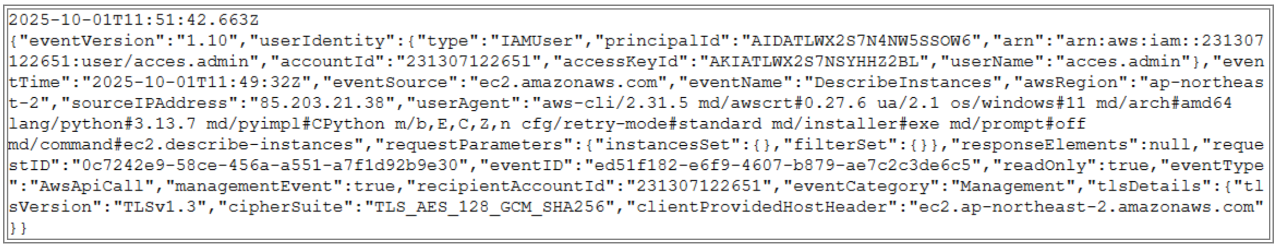


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 11:48:24+00:00	2025-10-01 06:48:24	us-east-1	GetAccountAuthorizationDetails	iam.amazonaws.com	85.203.21.25	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Discovery	2025-10-01 11:48:26+00:00	2025-10-01 06:48:26	us-east-1	GetAccountAuthorizationDetails	iam.amazonaws.com	85.203.21.25	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Discovery	2025-10-01 11:48:28+00:00	2025-10-01 06:48:28	us-east-1	GetAccountAuthorizationDetails	iam.amazonaws.com	85.203.21.25	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin

[Figure35] IAM information collection confirmed in the bitParser analysis result file

10) Collection of EC2 instance list via AWS CLI command

CloudTrail shows that the IAM user (acces.admin) used the 'ec2 describe-instances' command via AWS CLI from the IP 85.203.21.38 (Singapore) to collect the EC2 instance list.



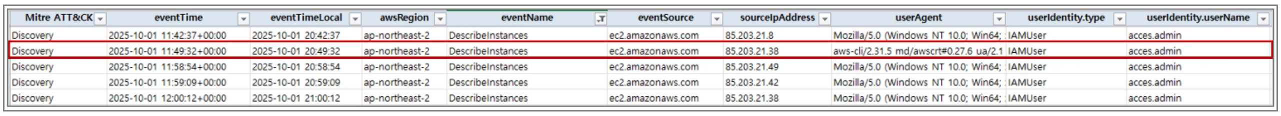
```
2025-10-01T11:51:42.663Z
{"eventVersion":"1.10","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T11:49:32Z","eventSource":"ec2.amazonaws.com","eventName":"DescribeInstances","awsRegion":"ap-northeas
t-2","sourceIPAddress":"85.203.21.38","userAgent":"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/b,E,C,Z,n cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#ec2.describe-instances","requestParameters":{"instancesSet":{},"filterSet":{}},"responseElements":null,"reque
stID":"0c7242e9-58ce-456a-a551-a7f1d92b9e30","eventID":"ed51f182-e6f9-4607-b879-ae7c2c3de6c5","readOnly":true,"eventType
":"AwsApiCall","managementEvent":true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tl
sVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"ec2.ap-northeast-2.amazonaws.com"
}}
```

[Figure36] EC2 instance list collection event via AWS CLI command observed in CloudTrail

[Table159] Key field details of the EC2 instance list collection event via AWS CLI command

Field	Key Field Details
EC2 instance list collection via AWS CLI command EC2 instance list collection	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T11:49:32Z • eventSource: ec2.amazonaws.com • eventName: DescribeInstances • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.38 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#Cpython m/b,E,C,Z,n cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#ec2.describe-instances

The bitParser analysis results file also confirms this behavior as follows.

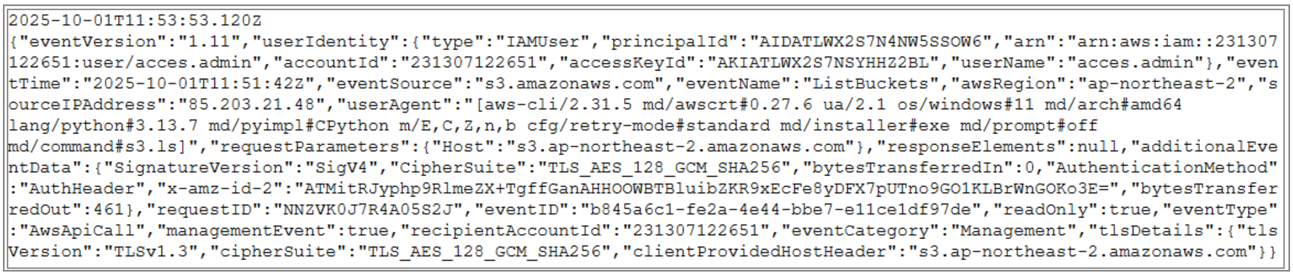


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 11:42:37+00:00	2025-10-01 20:42:37	ap-northeast-2	DescribeInstances	ec2.amazonaws.com	85.203.21.8	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin
Discovery	2025-10-01 11:49:32+00:00	2025-10-01 20:49:32	ap-northeast-2	DescribeInstances	ec2.amazonaws.com	85.203.21.38	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Discovery	2025-10-01 11:58:54+00:00	2025-10-01 20:58:54	ap-northeast-2	DescribeInstances	ec2.amazonaws.com	85.203.21.49	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin
Discovery	2025-10-01 11:59:09+00:00	2025-10-01 20:59:09	ap-northeast-2	DescribeInstances	ec2.amazonaws.com	85.203.21.42	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin
Discovery	2025-10-01 12:00:12+00:00	2025-10-01 21:00:12	ap-northeast-2	DescribeInstances	ec2.amazonaws.com	85.203.21.38	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin

[Figure37] EC2 instance list collection confirmed in the bitParser analysis result file

11) Collection of S3 bucket lists via AWS CLI commands

CloudTrail shows that the IAM user (acces.admin) collected S3 bucket lists using the 's3 ls' command via AWS CLI from the IP address 85.203.21.48 (Singapore).



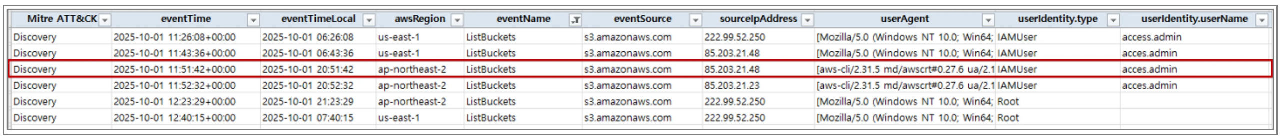
```
2025-10-01T11:53:53.120Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T11:51:42Z","eventSource":"s3.amazonaws.com","eventName":"ListBuckets","awsRegion":"ap-northeast-2","s
ourceIPAddress":"85.203.21.48","userAgent":"[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/E,C,Z,n,b cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#s3.ls]","requestParameters":{"Host":"s3.ap-northeast-2.amazonaws.com"},"responseElements":null,"additionalEve
ntData":{"SignatureVersion":"SigV4","CipherSuite":"TLS_AES_128_GCM_SHA256","bytesTransferredIn":0,"AuthenticationMethod"
:"AuthHeader","x-amz-id-2":"ATMitRJyphp9RlmeZX+TgffGanAHHOOWBTBluibZKR9xEcFe8yDFX7pUTno9GO1KLBrWnGOKo3E=","bytesTransfer
redOut":461},"requestID":"NNZVK0J7R4A05S2J","eventID":"b845a6c1-fe2a-4e44-bbe7-e11ce1df97de","readOnly":true,"eventType"
:"AwsApiCall","managementEvent":true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tls
Version":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"s3.ap-northeast-2.amazonaws.com"}}
```

[Figure38] S3 bucket listing collection event observed in CloudTrail

[Table160] Key field details of the S3 bucket listing event via AWS CLI command

Field	Key Field Details
S3 bucket list collection via AWS CLI command S3 bucket list collection	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T11:51:42Z • eventSource: s3.amazonaws.com • eventName: ListBuckets • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.48 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/E,C,Z,n,b cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3.ls

bitParser analysis results also confirm this behavior in the file as follows.

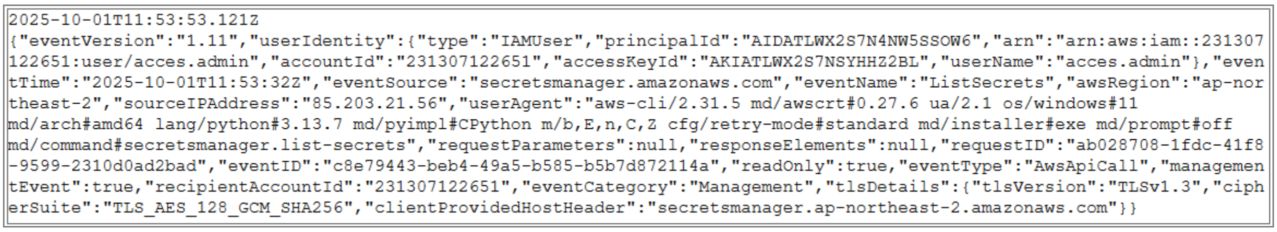


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 11:26:08+00:00	2025-10-01 06:26:08	us-east-1	ListBuckets	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	access.admin
Discovery	2025-10-01 11:43:36+00:00	2025-10-01 06:43:36	us-east-1	ListBuckets	s3.amazonaws.com	85.203.21.48	[Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin
Discovery	2025-10-01 11:51:42+00:00	2025-10-01 20:51:42	ap-northeast-2	ListBuckets	s3.amazonaws.com	85.203.21.48	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Discovery	2025-10-01 11:52:32+00:00	2025-10-01 20:52:32	ap-northeast-2	ListBuckets	s3.amazonaws.com	85.203.21.23	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Discovery	2025-10-01 12:23:29+00:00	2025-10-01 21:23:29	ap-northeast-2	ListBuckets	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Discovery	2025-10-01 12:40:15+00:00	2025-10-01 07:40:15	us-east-1	ListBuckets	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64;	Root	

[Figure39] S3 bucket list collection event confirmed in the bitParser analysis result file

12) Collecting SecretsManager information via AWS CLI commands

CloudTrail shows that the IAM user (acces.admin) collected SecretsManager information using AWS CLI commands from the IP address 85.203.21.56 (Singapore).



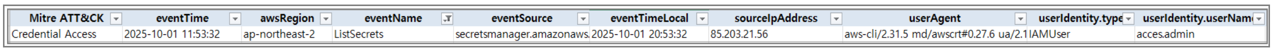
```
2025-10-01T11:53:53.121Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T11:53:32Z","eventSource":"secretsmanager.amazonaws.com","eventName":"ListSecrets","awsRegion":"ap-nor
theast-2","sourceIPAddress":"85.203.21.56","userAgent":"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11
md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/b,E,n,C,Z cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#secretsmanager.list-secrets","requestParameters":null,"responseElements":null,"requestID":"ab028708-1fdc-41f8
-9599-2310d0ad2bad","eventID":"c8e79443-beb4-49a5-b585-b5b7d872114a","readOnly":true,"eventType":"AwsApiCall","managemen
tEvent":true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","ciph
erSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"secretsmanager.ap-northeast-2.amazonaws.com"}}
```

[Figure40] SecretsManager information collection event observed in CloudTrail

[Table161] Key field details of the SecretsManager information collection event via AWS CLI commands

Field	Key Field Details
Collection of SecretsManager information via AWS CLI commands SecretsManager Information Collection	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T11:53:32Z • eventSource: secretsmanager.amazonaws.com • eventName: ListSecrets • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.56 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/b,E,n,C,Z cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#secretsmanager.list-secrets

The bitParser analysis results file also confirms this behavior as follows.

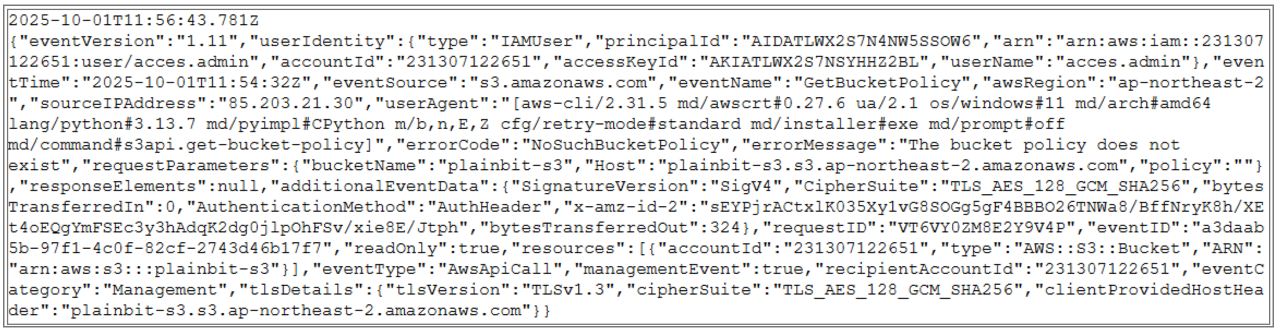


Mitre ATT&CK	eventTime	awsRegion	eventName	eventSource	eventTimeLocal	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Credential Access	2025-10-01 11:53:32	ap-northeast-2	ListSecrets	secretsmanager.amazonaws.	2025-10-01 20:53:32	85.203.21.56	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin

[Figure41] SecretsManager information collection event identified in the bitParser analysis result file

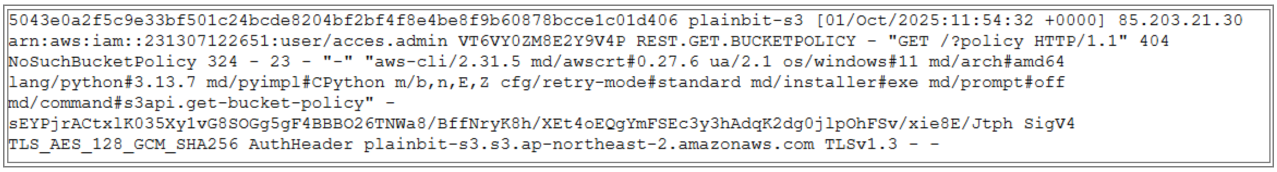
13) Attempt to collect S3 bucket policy information via AWS CLI commands

CloudTrail and S3 Server Access Logs revealed that an IAM user (acces.admin) attempted to collect S3 bucket (plainbit-s3) policy information using the 's3api get-bucket-policy' command via AWS CLI from the IP address 85.203.21.30 (Singapore).



```
2025-10-01T11:56:43.781Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T11:54:32Z","eventSource":"s3.amazonaws.com","eventName":"GetBucketPolicy","awsRegion":"ap-northeast-2
","sourceIPAddress":"85.203.21.30","userAgent":"[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/b,n,E,Z cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#s3api.get-bucket-policy]","errorCode":"NoSuchBucketPolicy","errorMessage":"The bucket policy does not
exist","requestParameters":{"bucketName":"plainbit-s3","Host":"plainbit-s3.s3.ap-northeast-2.amazonaws.com","policy":""}
,"responseElements":null,"additionalEventData":{"SignatureVersion":"SigV4","CipherSuite":"TLS_AES_128_GCM_SHA256","bytes
TransferredIn":0,"AuthenticationMethod":"AuthHeader","x-amz-id-2":"sEYPjrACtxlK035Xy1vG8SOGg5gF4BBBO26TNWa8/BffNryK8h/XE
t4oEQgYmFSEc3y3hAdqK2dg0jlpOhFSv/xie8E/Jtph","bytesTransferredOut":324},"requestID":"VT6VY0ZM8E2Y9V4P","eventID":"a3daab
5b-97f1-4c0f-82cf-2743d46b17f7","readOnly":true,"resources":[{"accountId":"231307122651","type":"AWS::S3::Bucket","ARN":
"arn:aws:s3:::plainbit-s3"}],"eventType":"AwsApiCall","managementEvent":true,"recipientAccountId":"231307122651","eventC
ategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHea
der":"plainbit-s3.s3.ap-northeast-2.amazonaws.com"}}
```

[Figure42] Event showing an attempt to collect S3 bucket policy information via AWS CLI command, as seen in CloudTrail



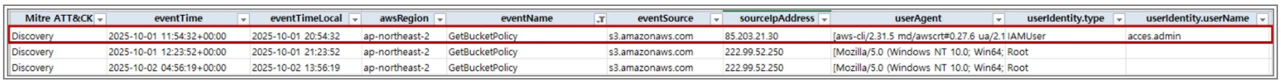
```
5043e0a2f5c9e33bf501c24bcde8204bf2bf4f8e4be8f9b60878bcce1c01d406 plainbit-s3 [01/Oct/2025:11:54:32 +0000] 85.203.21.30
arn:aws:iam::231307122651:user/acces.admin VT6VY0ZM8E2Y9V4P REST.GET.BUCKETPOLICY - "GET /?policy HTTP/1.1" 404
NoSuchBucketPolicy 324 - 23 - "-" "aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/b,n,E,Z cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#s3api.get-bucket-policy" -
sEYPjrACtxlK035Xy1vG8SOGg5gF4BBBO26TNWa8/BffNryK8h/XEt4oEQgYmFSEc3y3hAdqK2dg0jlpOhFSv/xie8E/Jtph SigV4
TLS_AES_128_GCM_SHA256 AuthHeader plainbit-s3.s3.ap-northeast-2.amazonaws.com TLSv1.3 - -
```

[Figure43] Event showing an attempt to collect S3 bucket policy information via AWS CLI command, as seen in the S3 Access Log

[Table162] Key field details for the S3 bucket policy information collection event via AWS CLI commands

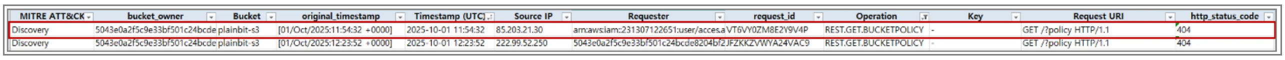
Category	Key Field Details
(CloudTrail) Attempt to collect S3 bucket policy information via AWS CLI commands S3 Bucket Policy Information Collection	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T11:54:32Z • eventSource: s3.amazonaws.com • eventName: GetBucketPolicy • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.30 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#Cpython m/b,n,E,Z cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.get-bucket-policy • errorCode: NoSuchBucketPolicy • requestParameters - bucketName: plainbit-s3
(S3 Server Access Log) Collecting S3 bucket policy information Collecting S3 bucket policy information	• bucketName: plainbit-s3 • eventTime: [01/Oct/2025:11:54:32 +0000] • sourceIPAddress: 85.203.21.30 • arn: arn:aws:iam::231307122651:user/acces.admin • task: REST.GET.BUCKETPOLICY • request: GET /?policy HTTP/1.1 • statusCode: 404 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#Cpython m/b,n,E,Z cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.get-bucket-policy

The bitParser analysis results also confirm this behavior in the file as follows.



Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 11:54:32+00:00	2025-10-01 20:54:32	ap-northeast-2	GetBucketPolicy	s3.amazonaws.com	85.203.21.30	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Discovery	2025-10-01 12:23:52+00:00	2025-10-01 21:23:52	ap-northeast-2	GetBucketPolicy	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Discovery	2025-10-02 04:56:19+00:00	2025-10-02 13:56:19	ap-northeast-2	GetBucketPolicy	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64;	Root	

[Figure44] Attempt to collect S3 bucket policy information (CloudTrail) identified in the bitParser analysis results file

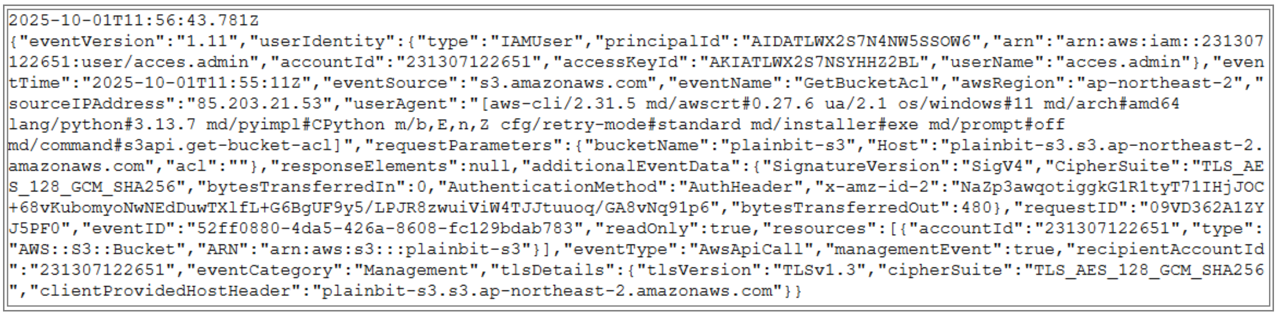


MITRE ATT&CK	bucket_owner	Bucket	original_timestamp	Timestamp (UTC)	Source IP	Requester	request_id	Operation	Key	Request URI	http_status_code
Discovery	5043e0a2f5c9e33bf501c24bcde	plainbit-s3	[01/Oct/2025:11:54:32 +0000]	2025-10-01 11:54:32	85.203.21.30	arn:aws:iam::231307122651:user/acces.a	VT6VY0ZM8E2Y9V4P	REST.GET.BUCKETPOLICY	-	GET /?policy HTTP/1.1	404
Discovery	5043e0a2f5c9e33bf501c24bcde	plainbit-s3	[01/Oct/2025:12:23:52 +0000]	2025-10-01 12:23:52	222.99.52.250	5043e0a2f5c9e33bf501c24bcde8204bf2	JFZKKZVWYA24VAC9	REST.GET.BUCKETPOLICY	-	GET /?policy HTTP/1.1	404

[Figure45] Attempted collection of S3 bucket policy information event identified in the bitParser analysis file (S3 Access Log)

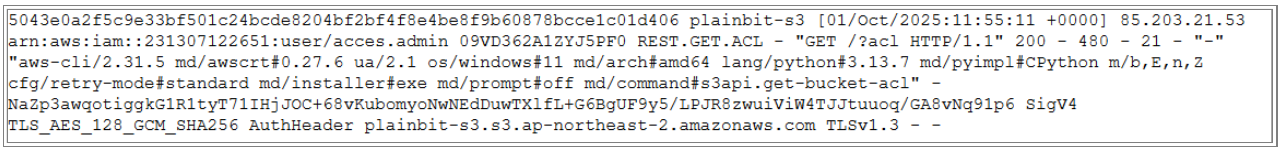
14) Collection of S3 bucket ACL information via AWS CLI commands

CloudTrail and S3 Access Log revealed that an IAM user (acces.admin) used the 's3api get-bucket-acl' command via AWS CLI from the IP address 85.203.21.53 (Singapore) to collect ACL information for the S3 bucket (plainbit-s3).



```
2025-10-01T11:56:43.781Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T11:55:11Z","eventSource":"s3.amazonaws.com","eventName":"GetBucketAcl","awsRegion":"ap-northeast-2","
sourceIPAddress":"85.203.21.53","userAgent":"[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/b,E,n,Z cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#s3api.get-bucket-acl]","requestParameters":{"bucketName":"plainbit-s3","Host":"plainbit-s3.s3.ap-northeast-2.
amazonaws.com","acl":""},"responseElements":null,"additionalEventData":{"SignatureVersion":"SigV4","CipherSuite":"TLS_AE
S_128_GCM_SHA256","bytesTransferredIn":0,"AuthenticationMethod":"AuthHeader","x-amz-id-2":"NaZp3awqotiggkG1R1tyT71IHjJOC
+68vKubomyoNwNEdDuwTXlfL+G6BgUF9y5/LPJR8zwuiViW4TJJtuuoq/GA8vNq91p6","bytesTransferredOut":480},"requestID":"09VD362A1ZY
J5PF0","eventID":"52ff0880-4da5-426a-8608-fc129bdab783","readOnly":true,"resources":[{"accountId":"231307122651","type":
"AWS::S3::Bucket","ARN":"arn:aws:s3:::plainbit-s3"}],"eventType":"AwsApiCall","managementEvent":true,"recipientAccountId
":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256
","clientProvidedHostHeader":"plainbit-s3.s3.ap-northeast-2.amazonaws.com"}}
```

[Figure46] Event showing collection of S3 bucket ACL information via AWS CLI command, as seen in CloudTrail



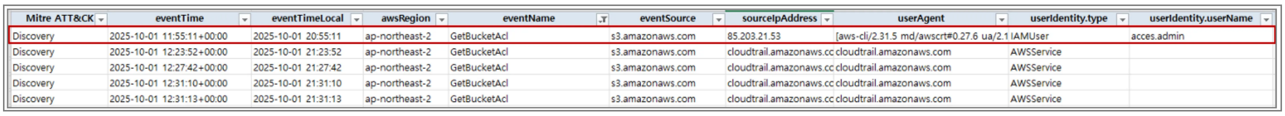
```
5043e0a2f5c9e33bf501c24bcde8204bf2bf4f8e4be8f9b60878bcce1c01d406 plainbit-s3 [01/Oct/2025:11:55:11 +0000] 85.203.21.53
arn:aws:iam::231307122651:user/acces.admin 09VD362A1ZYJ5PF0 REST.GET.ACL - "GET /?acl HTTP/1.1" 200 - 480 - 21 - "-"
"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/b,E,n,Z
cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.get-bucket-acl" -
NaZp3awqotiggkG1R1tyT71IHjJOC+68vKubomyoNwNEdDuwTXlfL+G6BgUF9y5/LPJR8zwuiViW4TJJtuuoq/GA8vNq91p6 SigV4
TLS_AES_128_GCM_SHA256 AuthHeader plainbit-s3.s3.ap-northeast-2.amazonaws.com TLSv1.3 - -
```

[Figure47] S3 Access Log showing the event of collecting S3 bucket ACL information via the AWS CLI command

[Table163] Key field details of the S3 bucket ACL information collection event via AWS CLI commands

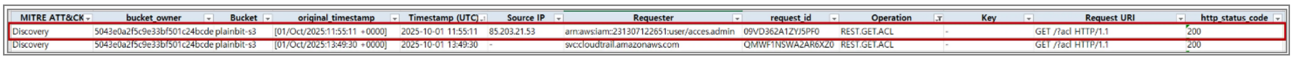
Category	Key Field Details
(CloudTrail) Collection of S3 bucket ACL information via AWS CLI commands S3 Bucket ACL Information Collection	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T11:55:11Z • eventSource: s3.amazonaws.com • eventName: GetBucketAcl • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.53 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#Cpython m/b,E,n,Z cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.get-bucket-acl • requestParameters - bucketName: plainbit-s3
(S3 Server Access Log) Collecting S3 bucket ACL information Collecting S3 bucket ACL information	• bucketName: plainbit-s3 • eventTime: [01/Oct/2025:11:55:11 +0000] • sourceIPAddress: 85.203.21.53 • arn: arn:aws:iam::231307122651:user/acces.admin • task: REST.GET.ACL • request: GET /?acl HTTP/1.1 • statusCode: 200 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#Cpython m/b,E,n,Z cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.get-bucket-acl

The bitParser analysis results file also confirms this activity as follows.



Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 11:55:11+00:00	2025-10-01 20:55:11	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	85.203.21.53	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Discovery	2025-10-01 12:23:52+00:00	2025-10-01 21:23:52	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.cc	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 12:27:42+00:00	2025-10-01 21:27:42	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.cc	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 12:31:10+00:00	2025-10-01 21:31:10	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.cc	cloudtrail.amazonaws.com	AWSService	
Discovery	2025-10-01 12:31:13+00:00	2025-10-01 21:31:13	ap-northeast-2	GetBucketAcl	s3.amazonaws.com	cloudtrail.amazonaws.cc	cloudtrail.amazonaws.com	AWSService	

[Figure48] S3 bucket ACL information collection event (CloudTrail) identified in the bitParser analysis result file

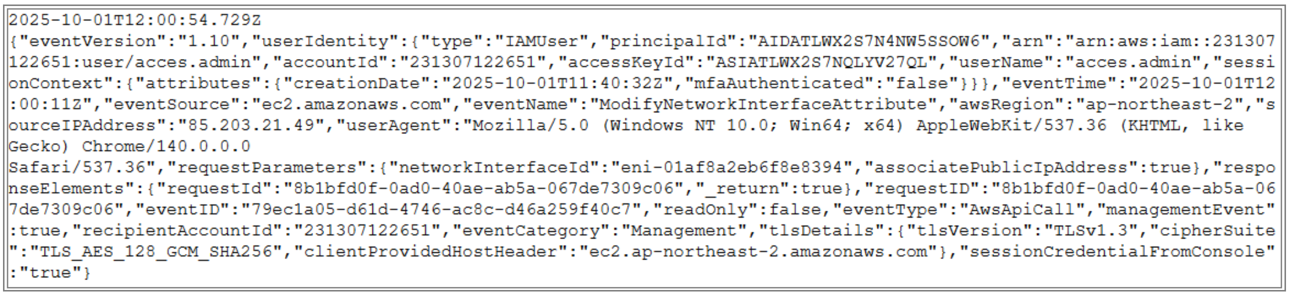


MITRE ATT&CK	bucket_owner	Bucket	original_timestamp	Timestamp (UTC)	Source IP	Requester	request_id	Operation	Key	Request URI	http_status_code
Discovery	5043e0a2f5c9e33bf501c24bcde	plainbit-s3	[01/Oct/2025:11:55:11 +0000]	2025-10-01 11:55:11	85.203.21.53	arn:aws:iam::231307122651:user/acces.admin	09VD362A1ZYJ5PF0	REST.GET.ACL	-	GET /?acl HTTP/1.1	200
Discovery	5043e0a2f5c9e33bf501c24bcde	plainbit-s3	[01/Oct/2025:13:49:30 +0000]	2025-10-01 13:49:30	-	svc:cloudtrail.amazonaws.com	QMWF1NSWA2AR6XZ0	REST.GET.ACL	-	GET /?acl HTTP/1.1	200

[Figure49] S3 bucket ACL information collection event (S3 Access Log) identified in the bitParser analysis result file

15) Assigning a Public IP Address to an EC2 Instance

CloudTrail showed that the IAM user (acces.admin) enabled the Public IP Address of the network interface (eni-01af8a2eb6f8e8394) via the Chrome browser from the IP address 85.203.21.49 (Singapore).



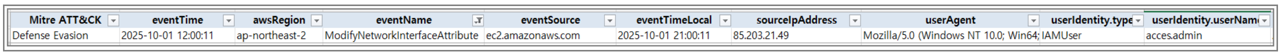
```
2025-10-01T12:00:54.729Z
{"eventVersion":"1.10","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"ASIATLWX2S7NQLYV27QL","userName":"acces.admin","sessi
onContext":{"attributes":{"creationDate":"2025-10-01T11:40:32Z","mfaAuthenticated":"false"}}},"eventTime":"2025-10-01T12
:00:11Z","eventSource":"ec2.amazonaws.com","eventName":"ModifyNetworkInterfaceAttribute","awsRegion":"ap-northeast-2","s
ourceIPAddress":"85.203.21.49","userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/140.0.0.0
Safari/537.36","requestParameters":{"networkInterfaceId":"eni-01af8a2eb6f8e8394","associatePublicIpAddress":true},"respo
nseElements":{"requestId":"8b1bfd0f-0ad0-40ae-ab5a-067de7309c06","_return":true},"requestID":"8b1bfd0f-0ad0-40ae-ab5a-06
7de7309c06","eventID":"79ec1a05-d61d-4746-ac8c-d46a259f40c7","readOnly":false,"eventType":"AwsApiCall","managementEvent"
:true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite
":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"ec2.ap-northeast-2.amazonaws.com"},"sessionCredentialFromConsole"
:"true"}
```

[Figure50] EC2 instance public IP address activation event visible in CloudTrail

[Table164] Key field details of the EC2 instance public IP address activation event

Field	Key Field Details
EC2 Instance Public IP Address Activation	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T12:00:11Z • eventSource: ec2.amazonaws.com • eventName: ModifyNetworkInterfaceAttribute • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.49 • userAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 • requestParameters - networkInterfaceId: eni-01af8a2eb6f8e8394 - associatePublicIpAddress: true

The bitParser analysis results file also confirms this behavior as follows.

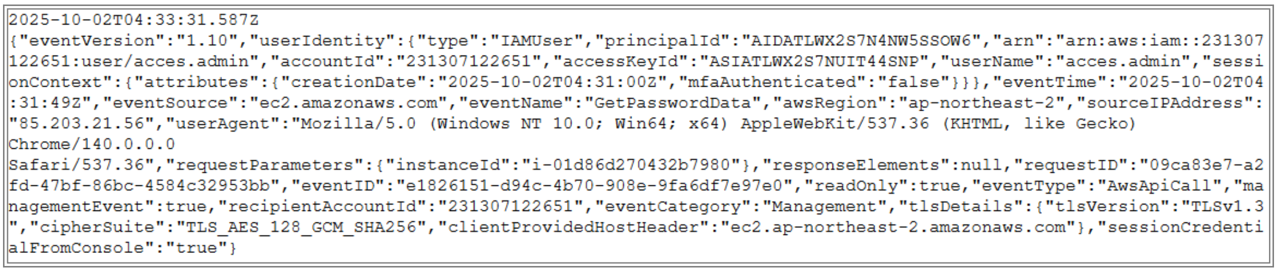


Mitre ATT&CK	eventTime	awsRegion	eventName	eventSource	eventTimeLocal	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userNam
Defense Evasion	2025-10-01 12:00:11	ap-northeast-2	ModifyNetworkInterfaceAttribute	ec2.amazonaws.com	2025-10-01 21:00:11	85.203.21.49	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin

[Figure51] EC2 instance public IP address activation event confirmed in the bitParser analysis result file

16) EC2 instance password acquisition

CloudTrail confirmed that an IAM user (acces.admin) obtained the password for an EC2 instance via the Chrome browser from the IP address 85.203.21.56 (Singapore).



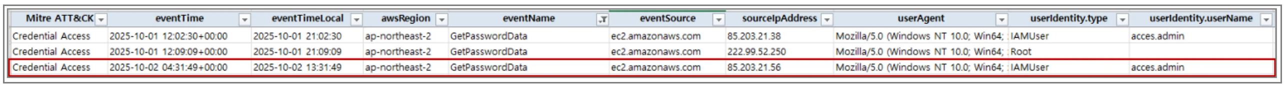
```
2025-10-02T04:33:31.587Z
{"eventVersion":"1.10","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"ASIATLWX2S7NUIT44SNP","userName":"acces.admin","sessi
onContext":{"attributes":{"creationDate":"2025-10-02T04:31:00Z","mfaAuthenticated":"false"}}},"eventTime":"2025-10-02T04
:31:49Z","eventSource":"ec2.amazonaws.com","eventName":"GetPasswordData","awsRegion":"ap-northeast-2","sourceIPAddress":
"85.203.21.56","userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/140.0.0.0
Safari/537.36","requestParameters":{"instanceId":"i-01d86d270432b7980"},"responseElements":null,"requestID":"09ca83e7-a2
fd-47bf-86bc-4584c32953bb","eventID":"e1826151-d94c-4b70-908e-9fa6df7e97e0","readOnly":true,"eventType":"AwsApiCall","ma
nagementEvent":true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3
","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"ec2.ap-northeast-2.amazonaws.com"},"sessionCredenti
alFromConsole":"true"}
```

[Figure52] EC2 instance password acquisition event observed in CloudTrail

[Table165] Key field details of the EC2 instance password acquisition event

Field	Key Field Details
EC2 Instance Password Acquisition	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-02T04:31:49Z • eventSource: ec2.amazonaws.com • eventName: GetPasswordData • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.56 • userAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 • requestParameters - instanceId: i-01d86d270432b7980

bitParser analysis results also confirm this behavior in the file as follows.

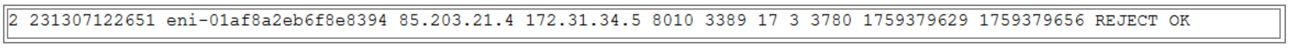


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Credential Access	2025-10-01 12:02:30+00:00	2025-10-01 21:02:30	ap-northeast-2	GetPasswordData	ec2.amazonaws.com	85.203.21.38	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin
Credential Access	2025-10-01 12:09:09+00:00	2025-10-01 21:09:09	ap-northeast-2	GetPasswordData	ec2.amazonaws.com	222.99.52.250	Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Credential Access	2025-10-02 04:31:49+00:00	2025-10-02 13:31:49	ap-northeast-2	GetPasswordData	ec2.amazonaws.com	85.203.21.56	Mozilla/5.0 (Windows NT 10.0; Win64;	IAMUser	acces.admin

[Figure53] bitParser analysis results showing the EC2 instance password acquisition event detected in the file

17) EC2 instance remote access (RDP)

VPC Flow Logs confirmed that the IP address 85.203.21.4 (Singapore) successfully accessed (ACCEPT) RDP (port 3389) to 172.31.34.5 (Public IP).



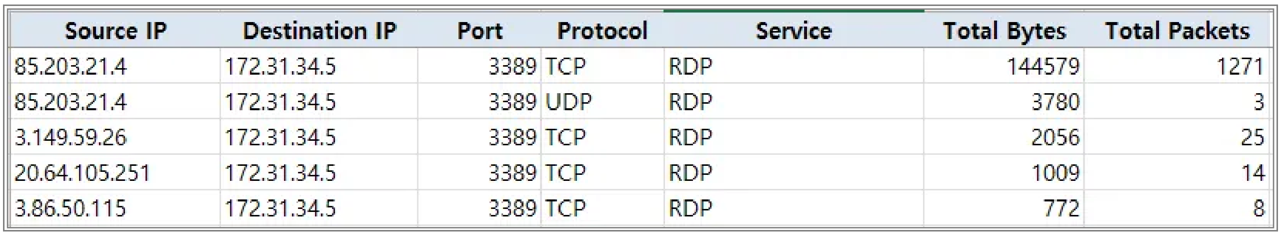
```
2 231307122651 eni-01af8a2eb6f8e8394 85.203.21.4 172.31.34.5 8010 3389 17 3 3780 1759379629 1759379656 REJECT OK
```

[Figure54] EC2 instance remote access (RDP) event observed in VPC Flow Logs

[Table166] Key field details of the remote access (RDP) event

Field	Key Field Details
Remote Access (RDP)	• account-id: 231307122651 • interface-id: eni-01af8a2eb6f8e8394 • srcaddr: 85.203.21.4 • Destination Address: 172.31.34.5 • srcport: 20813 • dstport: 3389 • protocol: 6 • packets: 1167 • bytes: 137,739 • start: 1759379663 • end: 1759379673 • action: ACCEPT

The bitParser analysis results file also confirms this activity as follows.

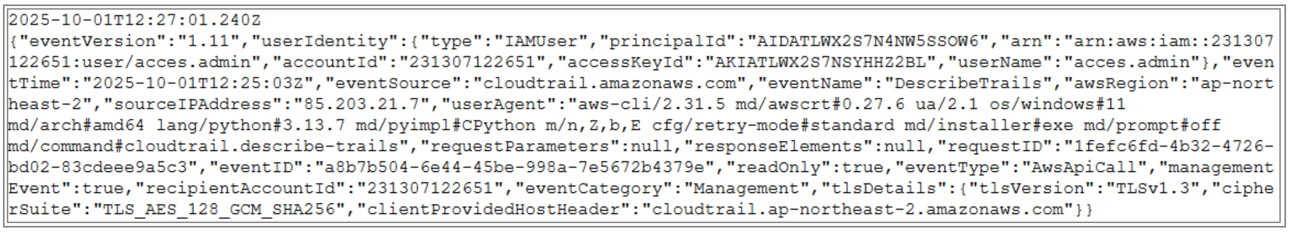


Source IP	Destination IP	Port	Protocol	Service	Total Bytes	Total Packets
85.203.21.4	172.31.34.5	3389	TCP	RDP	144579	1271
85.203.21.4	172.31.34.5	3389	UDP	RDP	3780	3
3.149.59.26	172.31.34.5	3389	TCP	RDP	2056	25
20.64.105.251	172.31.34.5	3389	TCP	RDP	1009	14
3.86.50.115	172.31.34.5	3389	TCP	RDP	772	8

[Figure55] Remote access (RDP) event identified in the bitParser analysis result file

18) Collecting CloudTrail lists via AWS CLI commands

CloudTrail shows that the IAM user (acces.admin) executed the 'cloudtrail describe-trails' command via AWS CLI from the IP 85.203.21.7 (Singapore) to collect the CloudTrail list.



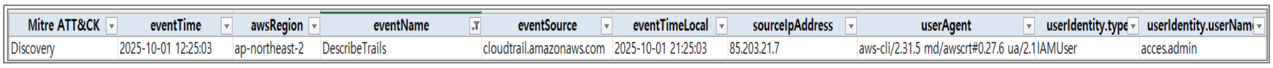
```
2025-10-01T12:27:01.240Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T12:25:03Z","eventSource":"cloudtrail.amazonaws.com","eventName":"DescribeTrails","awsRegion":"ap-nort
heast-2","sourceIPAddress":"85.203.21.7","userAgent":"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11
md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/n,Z,b,E cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#cloudtrail.describe-trails","requestParameters":null,"responseElements":null,"requestID":"1fefc6fd-4b32-4726-
bd02-83cdeee9a5c3","eventID":"a8b7b504-6e44-45be-998a-7e5672b4379e","readOnly":true,"eventType":"AwsApiCall","management
Event":true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","ciphe
rSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"cloudtrail.ap-northeast-2.amazonaws.com"}}
```

[Figure56] CloudTrail list collection event via AWS CLI command observed in CloudTrail

[Table167] Key field details of the CloudTrail list collection event via AWS CLI command

Field	Key Field Details
CloudTrail list collection via AWS CLI command CloudTrail List Collection	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T12:25:03Z • eventSource: cloudtrail.amazonaws.com • eventName: DescribeTrails • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.7 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/n,Z,b,E cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#cloudtrail.describe-trails

The bitParser analysis results file also confirms this behavior as follows.

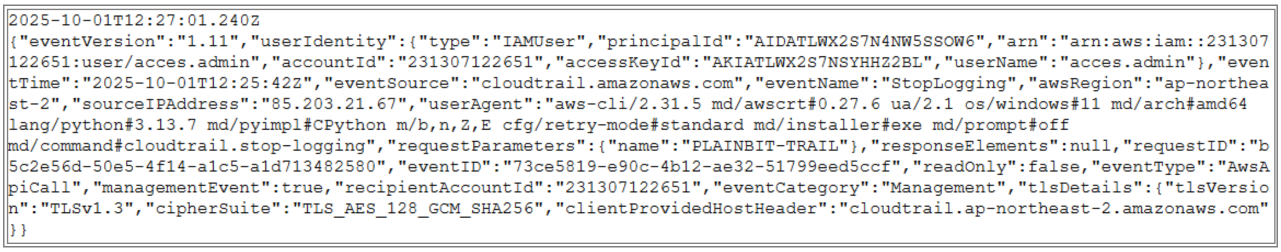


Mitre ATT&CK	eventTime	awsRegion	eventName	eventSource	eventTimeLocal	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userNam
Discovery	2025-10-01 12:25:03	ap-northeast-2	DescribeTrails	cloudtrail.amazonaws.com	2025-10-01 21:25:03	85.203.21.7	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin

[Figure57] CloudTrail list collection event identified in the bitParser analysis result file

19) Disabling CloudTrail via AWS CLI commands

CloudTrail showed that an IAM user (acces.admin) used the 'cloudtrail stop-logging' command via the AWS CLI from the IP address 85.203.21.67 (Singapore) to disable the operational status of CloudTrail (PLAINBIT-TRAIL).



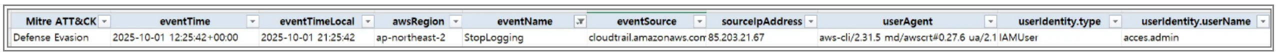
```
2025-10-01T12:27:01.240Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T12:25:42Z","eventSource":"cloudtrail.amazonaws.com","eventName":"StopLogging","awsRegion":"ap-northea
st-2","sourceIPAddress":"85.203.21.67","userAgent":"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/b,n,Z,E cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#cloudtrail.stop-logging","requestParameters":{"name":"PLAINBIT-TRAIL"},"responseElements":null,"requestID":"b
5c2e56d-50e5-4f14-a1c5-a1d713482580","eventID":"73ce5819-e90c-4b12-ae32-51799eed5ccf","readOnly":false,"eventType":"AwsA
piCall","managementEvent":true,"recipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersio
n":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"cloudtrail.ap-northeast-2.amazonaws.com"
}}
```

[Figure58] CloudTrail deactivation event via AWS CLI command observed in CloudTrail

[Table168] Key field details of the CloudTrail deactivation event via AWS CLI command

Field	Key Field Details
Disabling CloudTrail via AWS CLI command Disabling CloudTrail	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T12:25:42Z • eventSource: cloudtrail.amazonaws.com • eventName: StopLogging • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.67 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/b,n,Z,E cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#cloudtrail.stop-logging • requestParameters - name: PLAINBIT-TRAIL

bitParser analysis results file also confirms this activity as follows.

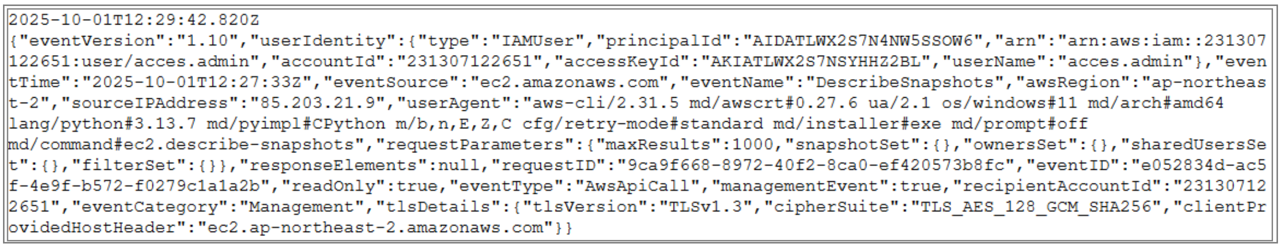


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Defense Evasion	2025-10-01 12:25:42+00:00	2025-10-01 21:25:42	ap-northeast-2	StopLogging	cloudtrail.amazonaws.cor	85.203.21.67	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin

[Figure59] CloudTrail deactivation event confirmed in the bitParser analysis result file

20) Collecting EC2 snapshot list via AWS CLI command

CloudTrail shows that an IAM user (acces.admin) collected the EC2 snapshot list using the 'ec2 describe-snapshots' command via AWS CLI from the IP address 85.203.21.9 (Singapore).



```
2025-10-01T12:29:42.820Z
{"eventVersion":"1.10","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T12:27:33Z","eventSource":"ec2.amazonaws.com","eventName":"DescribeSnapshots","awsRegion":"ap-northeas
t-2","sourceIPAddress":"85.203.21.9","userAgent":"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/b,n,E,Z,C cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#ec2.describe-snapshots","requestParameters":{"maxResults":1000,"snapshotSet":{},"ownersSet":{},"sharedUsersSe
t":{},"filterSet":{}},"responseElements":null,"requestID":"9ca9f668-8972-40f2-8ca0-ef420573b8fc","eventID":"e052834d-ac5
f-4e9f-b572-f0279c1a1a2b","readOnly":true,"eventType":"AwsApiCall","managementEvent":true,"recipientAccountId":"23130712
2651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientPr
ovidedHostHeader":"ec2.ap-northeast-2.amazonaws.com"}}
```

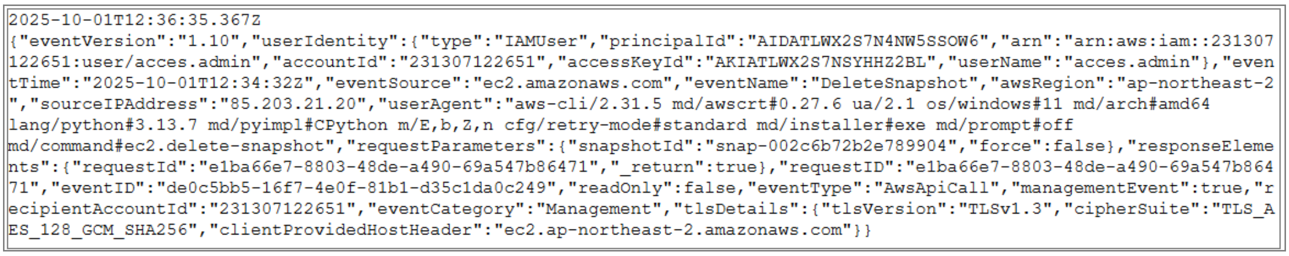
[Figure60] Event showing collection of EC2 snapshot list via AWS CLI command in CloudTrail

[Table169] Key field details of the EC2 snapshot list collection event via AWS CLI command

Field	Key Field Details
EC2 snapshot collection via AWS CLI command EC2 snapshot list collection	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T12:27:33Z ~ 2025-10-01T12:27:59Z • eventSource: ec2.amazonaws.com • eventName: DescribeSnapshots • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.9 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/b,n,E,Z,C cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#ec2.describe-snapshots

21) Deleting EC2 Snapshots via AWS CLI Commands

CloudTrail showed that an IAM user (acces.admin) deleted an EC2 snapshot using the 'ec2 delete-snapshot' command via AWS CLI from the IP 85.203.21.20 (Singapore).



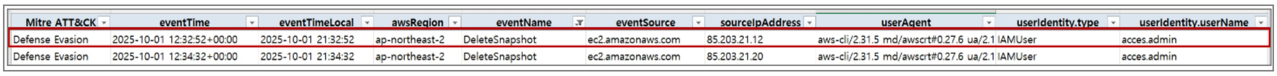
```
2025-10-01T12:36:35.367Z
{"eventVersion":"1.10","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T12:34:32Z","eventSource":"ec2.amazonaws.com","eventName":"DeleteSnapshot","awsRegion":"ap-northeast-2
","sourceIPAddress":"85.203.21.20","userAgent":"aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/E,b,Z,n cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#ec2.delete-snapshot","requestParameters":{"snapshotId":"snap-002c6b72b2e789904","force":false},"responseEleme
nts":{"requestId":"e1ba66e7-8803-48de-a490-69a547b86471","_return":true},"requestID":"e1ba66e7-8803-48de-a490-69a547b864
71","eventID":"de0c5bb5-16f7-4e0f-81b1-d35c1da0c249","readOnly":false,"eventType":"AwsApiCall","managementEvent":true,"r
ecipientAccountId":"231307122651","eventCategory":"Management","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_A
ES_128_GCM_SHA256","clientProvidedHostHeader":"ec2.ap-northeast-2.amazonaws.com"}}
```

[Figure61] EC2 snapshot deletion event via AWS CLI command observed in CloudTrail

[Table170] Key Fields in EC2 Snapshot Deletion Events via AWS CLI Commands

Field	Key Field Details
Deleting EC2 Snapshots via AWS CLI Commands EC2 Snapshot Deletion	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T12:34:32Z • eventSource: ec2.amazonaws.com • eventName: DeleteSnapshot • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.20 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/E,b,Z,n cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#ec2.delete-snapshot • requestParameters - snapshotId: snap-002c6b72b2e789904

bitParser analysis results also confirm this behavior in the file as follows.

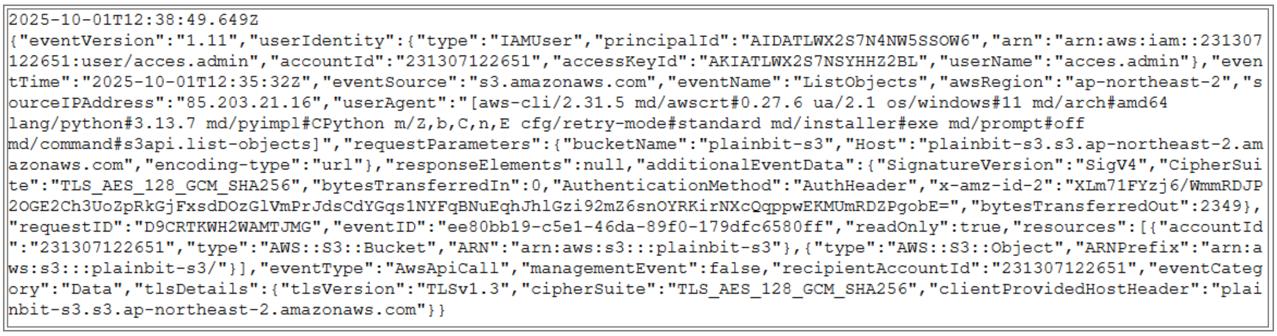


Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Defense Evasion	2025-10-01 12:32:52+00:00	2025-10-01 21:32:52	ap-northeast-2	DeleteSnapshot	ec2.amazonaws.com	85.203.21.12	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin
Defense Evasion	2025-10-01 12:34:32+00:00	2025-10-01 21:34:32	ap-northeast-2	DeleteSnapshot	ec2.amazonaws.com	85.203.21.20	aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1	IAMUser	acces.admin

[Figure62] EC2 snapshot deletion event confirmed in the bitParser analysis result file

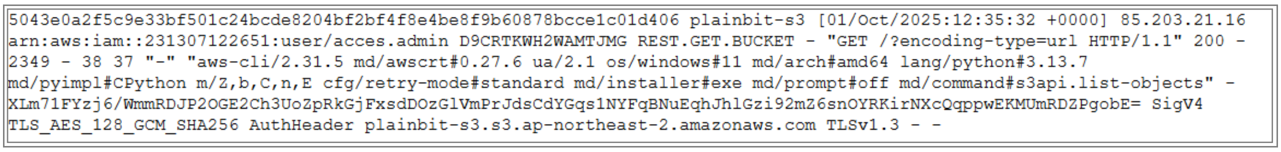
22) Collecting object list in S3 bucket via AWS CLI command

CloudTrail and S3 Access Logs revealed that the IAM user (acces.admin) used the 's3api list-objects' command via AWS CLI from the IP 85.203.21.16 (Singapore) to collect the object list from the S3 bucket (plainbit-s3).



```
2025-10-01T12:38:49.649Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T12:35:32Z","eventSource":"s3.amazonaws.com","eventName":"ListObjects","awsRegion":"ap-northeast-2","s
ourceIPAddress":"85.203.21.16","userAgent":"[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/Z,b,C,n,E cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#s3api.list-objects]","requestParameters":{"bucketName":"plainbit-s3","Host":"plainbit-s3.s3.ap-northeast-2.am
azonaws.com","encoding-type":"url"},"responseElements":null,"additionalEventData":{"SignatureVersion":"SigV4","CipherSui
te":"TLS_AES_128_GCM_SHA256","bytesTransferredIn":0,"AuthenticationMethod":"AuthHeader","x-amz-id-2":"XLm71FYzj6/WmmRDJP
2OGE2Ch3UoZpRkGjFxsdDOzGlVmPrJdsCdYGqs1NYFqBNuEqhJhlGzi92mZ6snOYRKirNXcQqppwEKMUmRDZPgobE=","bytesTransferredOut":2349},
"requestID":"D9CRTKWH2WAMTJMG","eventID":"ee80bb19-c5e1-46da-89f0-179dfc6580ff","readOnly":true,"resources":[{"accountId
":"231307122651","type":"AWS::S3::Bucket","ARN":"arn:aws:s3:::plainbit-s3"},{"type":"AWS::S3::Object","ARNPrefix":"arn:a
ws:s3:::plainbit-s3/"}],"eventType":"AwsApiCall","managementEvent":false,"recipientAccountId":"231307122651","eventCateg
ory":"Data","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHeader":"plai
nbit-s3.s3.ap-northeast-2.amazonaws.com"}}
```

[Figure63] Event showing collection of object list in S3 bucket via AWS CLI command, as seen in CloudTrail



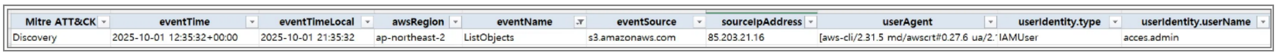
```
5043e0a2f5c9e33bf501c24bcde8204bf2bf4f8e4be8f9b60878bcce1c01d406 plainbit-s3 [01/Oct/2025:12:35:32 +0000] 85.203.21.16
arn:aws:iam::231307122651:user/acces.admin D9CRTKWH2WAMTJMG REST.GET.BUCKET - "GET /?encoding-type=url HTTP/1.1" 200 -
2349 - 38 37 "-" "aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7
md/pyimpl#CPython m/Z,b,C,n,E cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.list-objects" -
XLm71FYzj6/WmmRDJP2OGE2Ch3UoZpRkGjFxsdDOzGlVmPrJdsCdYGqs1NYFqBNuEqhJhlGzi92mZ6snOYRKirNXcQqppwEKMUmRDZPgobE= SigV4
TLS_AES_128_GCM_SHA256 AuthHeader plainbit-s3.s3.ap-northeast-2.amazonaws.com TLSv1.3 - -
```

[Figure64] Event of collecting object list in S3 bucket via AWS CLI command as seen in S3 Access Log

[Table171] Key field details of the event for collecting the list of objects in an S3 bucket via AWS CLI commands

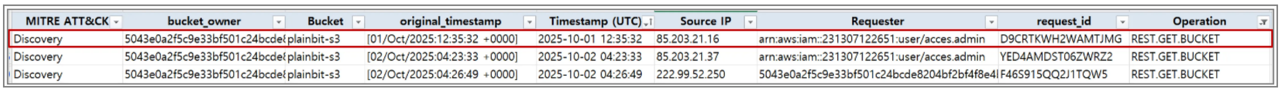
Field	Key Field Details
(CloudTrail) Collection of object list in S3 bucket via AWS CLI command Collection of object list in S3 bucket	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T12:35:32Z • eventSource: s3.amazonaws.com • eventName: ListObjects • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.16 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/Z,b,C,n,E cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.list-objects • requestParameters - bucketName: plainbit-s3 - encoding-type: url
(S3 Server Access Log) Collecting the list of objects within an S3 bucket Collecting object list within S3 bucket	• bucketName: plainbit-s3 • eventTime: [01/Oct/2025:12:35:32 +0000] • sourceIPAddress: 85.203.21.16 • arn:aws:iam::231307122651:user/access.admin • task: REST.GET.BUCKET • request: GET /?encoding-type=url HTTP/1.1 • statusCode: 200 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/Z,b,C,n,E cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.list-objects

The bitParser analysis results file also confirms this behavior as follows.



Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Discovery	2025-10-01 12:35:32+00:00	2025-10-01 21:35:32	ap-northeast-2	ListObjects	s3.amazonaws.com	85.203.21.16	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin

[Figure65] S3 bucket object list collection event (CloudTrail) identified in the bitParser analysis result file

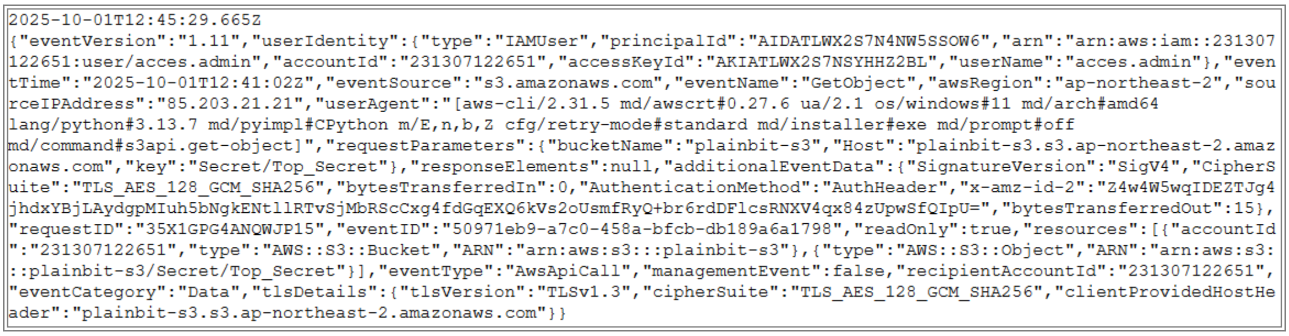


MITRE ATT&CK	bucket_owner	Bucket	original_timestamp	Timestamp (UTC)	Source IP	Requester	request_id	Operation
Discovery	5043e0a2f5c9e33bf501c24bcde8	plainbit-s3	[01/Oct/2025:12:35:32 +0000]	2025-10-01 12:35:32	85.203.21.16	arn:aws:iam::231307122651:user/acces.admin	D9CRTKWH2WAMTJMG	REST.GET.BUCKET
Discovery	5043e0a2f5c9e33bf501c24bcde8	plainbit-s3	[02/Oct/2025:04:23:33 +0000]	2025-10-02 04:23:33	85.203.21.37	arn:aws:iam::231307122651:user/acces.admin	YED4AMDST06ZWRZ2	REST.GET.BUCKET
Discovery	5043e0a2f5c9e33bf501c24bcde8	plainbit-s3	[02/Oct/2025:04:26:49 +0000]	2025-10-02 04:26:49	222.99.52.250	5043e0a2f5c9e33bf501c24bcde8204bf2bf4f8e4	F46S915QQ2J1TQW5	REST.GET.BUCKET

[Figure66] Event to collect object list within S3 bucket (S3 Access Log) identified in the bitParser analysis result file

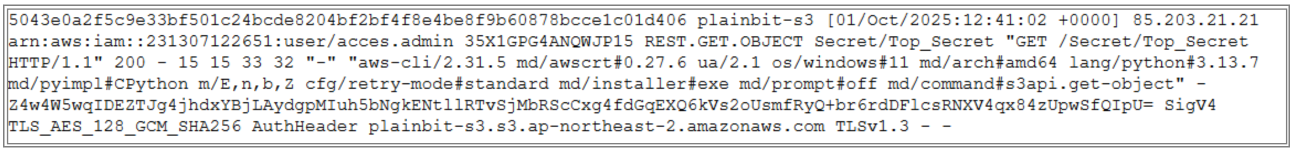
23) Downloading objects from an S3 bucket via AWS CLI commands

CloudTrail and S3 Access Log revealed that the IAM user (acces.admin) executed the 's3api get-objects' command via AWS CLI from the IP address 85.203.21.21 (Singapore) to download the object (Top_Secret) from the S3 bucket (plainbit-s3).



```
2025-10-01T12:45:29.665Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-01T12:41:02Z","eventSource":"s3.amazonaws.com","eventName":"GetObject","awsRegion":"ap-northeast-2","sou
rceIPAddress":"85.203.21.21","userAgent":"[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/E,n,b,Z cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#s3api.get-object]","requestParameters":{"bucketName":"plainbit-s3","Host":"plainbit-s3.s3.ap-northeast-2.amaz
onaws.com","key":"Secret/Top_Secret"},"responseElements":null,"additionalEventData":{"SignatureVersion":"SigV4","CipherS
uite":"TLS_AES_128_GCM_SHA256","bytesTransferredIn":0,"AuthenticationMethod":"AuthHeader","x-amz-id-2":"Z4w4W5wqIDEZTJg4
jhdxYBjLAydgpMIuh5bNgkENtllRTvSjMbRScCxg4fdGqEXQ6kVs2oUsmfRyQ+br6rdDFlcsRNXV4qx84zUpwSfQIpU=","bytesTransferredOut":15},
"requestID":"35X1GPG4ANQWJP15","eventID":"50971eb9-a7c0-458a-bfcb-db189a6a1798","readOnly":true,"resources":[{"accountId
":"231307122651","type":"AWS::S3::Bucket","ARN":"arn:aws:s3:::plainbit-s3"},{"type":"AWS::S3::Object","ARN":"arn:aws:s3:
::plainbit-s3/Secret/Top_Secret"}],"eventType":"AwsApiCall","managementEvent":false,"recipientAccountId":"231307122651",
"eventCategory":"Data","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientProvidedHostHe
ader":"plainbit-s3.s3.ap-northeast-2.amazonaws.com"}}
```

[Figure67] Event showing object download from S3 bucket via AWS CLI command in CloudTrail



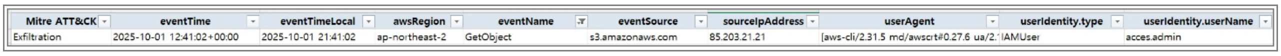
```
5043e0a2f5c9e33bf501c24bcde8204bf2bf4f8e4be8f9b60878bcce1c01d406 plainbit-s3 [01/Oct/2025:12:41:02 +0000] 85.203.21.21
arn:aws:iam::231307122651:user/acces.admin 35X1GPG4ANQWJP15 REST.GET.OBJECT Secret/Top_Secret "GET /Secret/Top_Secret
HTTP/1.1" 200 - 15 15 33 32 "-" "aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7
md/pyimpl#CPython m/E,n,b,Z cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.get-object" -
Z4w4W5wqIDEZTJg4jhdxYBjLAydgpMIuh5bNgkENtllRTvSjMbRScCxg4fdGqEXQ6kVs2oUsmfRyQ+br6rdDFlcsRNXV4qx84zUpwSfQIpU= SigV4
TLS_AES_128_GCM_SHA256 AuthHeader plainbit-s3.s3.ap-northeast-2.amazonaws.com TLSv1.3 - -
```

[Figure68] Event of downloading an object from an S3 bucket via an AWS CLI command, as seen in the S3 Access Log

[Table172] Key field details for the object download event in an S3 bucket via AWS CLI command

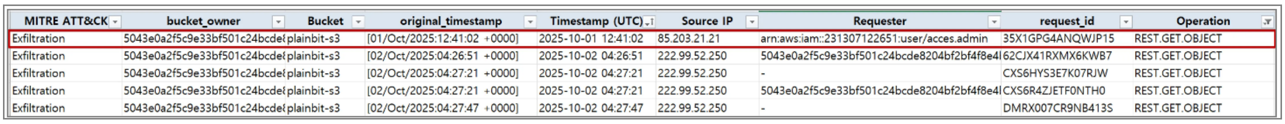
Field	Key Field Details
(CloudTrail) Object download events in an S3 bucket via AWS CLI commands Object Download in S3 Bucket	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-01T12:41:02Z • eventSource: s3.amazonaws.com • eventName: GetObject • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.21 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/E,n,b,Z cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.get-object • requestParameters - bucketName: plainbit-s3 - key: Secret/Top_Secret
(S3 Server Access Log) Downloading objects from an S3 bucket Downloading objects from an S3 bucket	• bucketName: plainbit-s3 • eventTime: [01/Oct/2025:12:41:02 +0000] • sourceIPAddress: 85.203.21.21 • arn: arn:aws:iam::231307122651:user/acces.admin • task: REST.GET.OBJECT • targetObject(Key): Secret/Top_Secret • request: GET /Secret/Top_Secret HTTP/1.1 • statusCode: 200 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/E,n,b,Z cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3api.get-object

The bitParser analysis results file also confirms this behavior as follows.



Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Exfiltration	2025-10-01 12:41:02+00:00	2025-10-01 21:41:02	ap-northeast-2	GetObject	s3.amazonaws.com	85.203.21.21	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin

[Figure69] Object download events (CloudTrail) within the S3 bucket confirmed in the bitParser analysis results file.

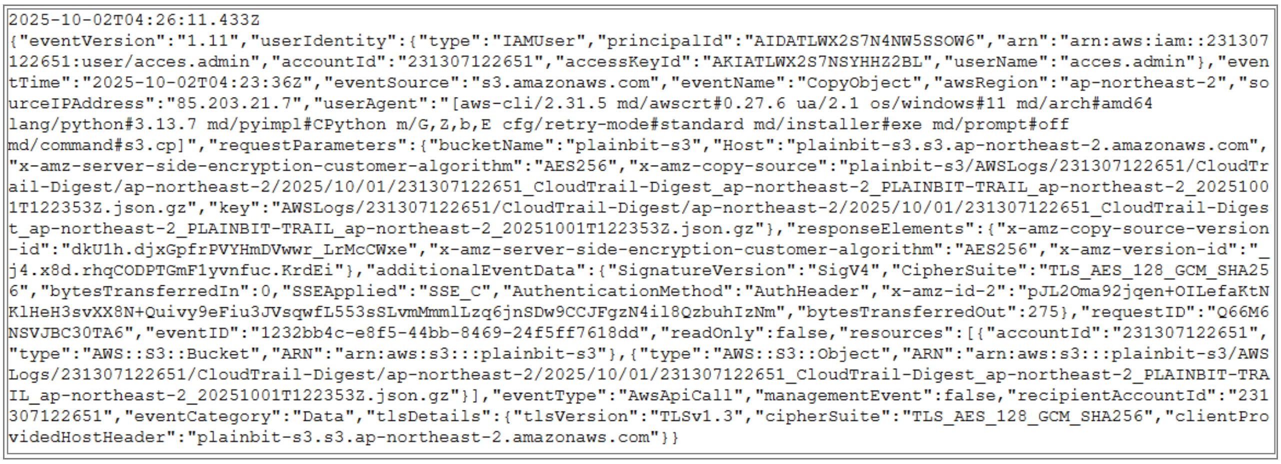


MITRE ATT&CK	bucket_owner	Bucket	original_timestamp	Timestamp (UTC)	Source IP	Requester	request_id	Operation
Exfiltration	5043e0a2f5c9e33bf501c24bcde8	plainbit-s3	[01/Oct/2025:12:41:02 +0000]	2025-10-01 12:41:02	85.203.21.21	arn:aws:iam::231307122651:user/acces.admin	35X1GPG4ANQWJP15	REST.GET.OBJECT
Exfiltration	5043e0a2f5c9e33bf501c24bcde8	plainbit-s3	[02/Oct/2025:04:26:51 +0000]	2025-10-02 04:26:51	222.99.52.250	5043e0a2f5c9e33bf501c24bcde8204bf2bf4f8e4	62CJX41RXMX6KWB7	REST.GET.OBJECT
Exfiltration	5043e0a2f5c9e33bf501c24bcde8	plainbit-s3	[02/Oct/2025:04:27:21 +0000]	2025-10-02 04:27:21	222.99.52.250	-	CXS6HYS3E7K07RJW	REST.GET.OBJECT
Exfiltration	5043e0a2f5c9e33bf501c24bcde8	plainbit-s3	[02/Oct/2025:04:27:21 +0000]	2025-10-02 04:27:21	222.99.52.250	5043e0a2f5c9e33bf501c24bcde8204bf2bf4f8e4	CXS6R4ZJETF0NTH0	REST.GET.OBJECT
Exfiltration	5043e0a2f5c9e33bf501c24bcde8	plainbit-s3	[02/Oct/2025:04:27:47 +0000]	2025-10-02 04:27:47	222.99.52.250	-	DMRX007CR9NB413S	REST.GET.OBJECT

[Figure70] Object download events (S3 Access Log) identified in the file analyzed by bitParser within the S3 bucket

24) Object encryption within the S3 bucket via AWS CLI commands

CloudTrail shows that the IAM user (acces.admin) encrypted (SSE_C) and then copied objects from the S3 bucket (plainbit-s3) using the 's3 cp' command via AWS CLI from the IP address 85.203.21.7 (Singapore).



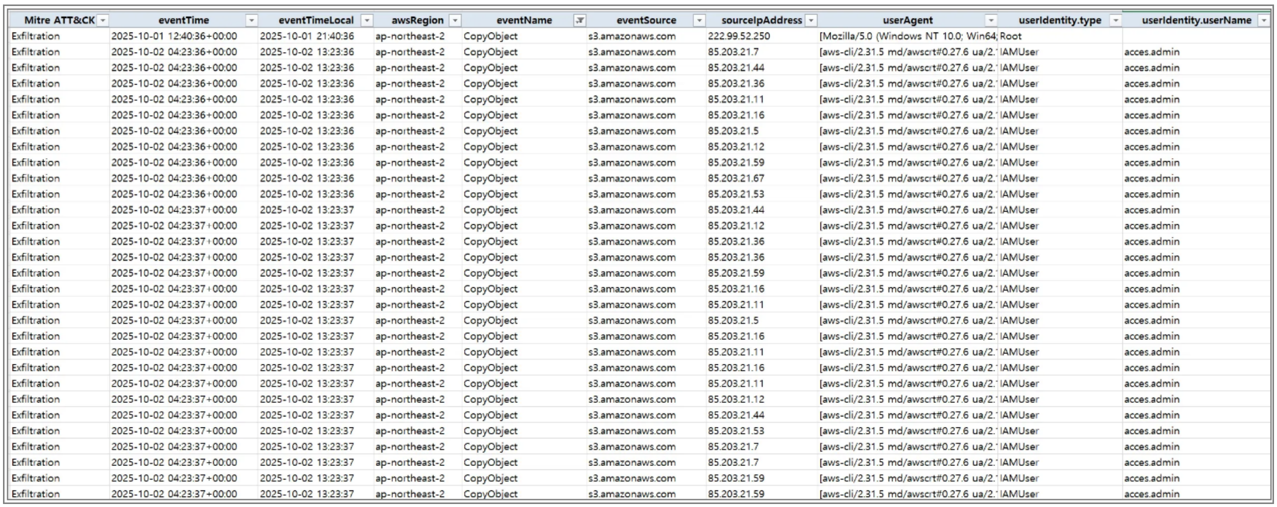
```
2025-10-02T04:26:11.433Z
{"eventVersion":"1.11","userIdentity":{"type":"IAMUser","principalId":"AIDATLWX2S7N4NW5SSOW6","arn":"arn:aws:iam::231307
122651:user/acces.admin","accountId":"231307122651","accessKeyId":"AKIATLWX2S7NSYHHZ2BL","userName":"acces.admin"},"even
tTime":"2025-10-02T04:23:36Z","eventSource":"s3.amazonaws.com","eventName":"CopyObject","awsRegion":"ap-northeast-2","so
urceIPAddress":"85.203.21.7","userAgent":"[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64
lang/python#3.13.7 md/pyimpl#CPython m/G,Z,b,E cfg/retry-mode#standard md/installer#exe md/prompt#off
md/command#s3.cp]","requestParameters":{"bucketName":"plainbit-s3","Host":"plainbit-s3.s3.ap-northeast-2.amazonaws.com",
"x-amz-server-side-encryption-customer-algorithm":"AES256","x-amz-copy-source":"plainbit-s3/AWSLogs/231307122651/CloudTr
ail-Digest/ap-northeast-2/2025/10/01/231307122651_CloudTrail-Digest_ap-northeast-2_PLAINBIT-TRAIL_ap-northeast-2_2025100
1T122353Z.json.gz","key":"AWSLogs/231307122651/CloudTrail-Digest/ap-northeast-2/2025/10/01/231307122651_CloudTrail-Diges
t_ap-northeast-2_PLAINBIT-TRAIL_ap-northeast-2_20251001T122353Z.json.gz"},"responseElements":{"x-amz-copy-source-version
-id":"dkU1h.djxGpfrPVYHmDVwwr_LrMcCWxe","x-amz-server-side-encryption-customer-algorithm":"AES256","x-amz-version-id":"_
j4.x8d.rhqCODPTGmF1yvnfuc.KrdEi"},"additionalEventData":{"SignatureVersion":"SigV4","CipherSuite":"TLS_AES_128_GCM_SHA25
6","bytesTransferredIn":0,"SSEApplied":"SSE_C","AuthenticationMethod":"AuthHeader","x-amz-id-2":"pJL2Oma92jqen+OILefaKtN
KlHeH3svXX8N+Quivy9eFiu3JVsqwfL553sSLvmMmmlLzq6jnSDw9CCJFgzN4il8QzbuhIzNm","bytesTransferredOut":275},"requestID":"Q66M6
NSVJBC30TA6","eventID":"1232bb4c-e8f5-44bb-8469-24f5ff7618dd","readOnly":false,"resources":[{"accountId":"231307122651",
"type":"AWS::S3::Bucket","ARN":"arn:aws:s3:::plainbit-s3"},{"type":"AWS::S3::Object","ARN":"arn:aws:s3:::plainbit-s3/AWS
Logs/231307122651/CloudTrail-Digest/ap-northeast-2/2025/10/01/231307122651_CloudTrail-Digest_ap-northeast-2_PLAINBIT-TRA
IL_ap-northeast-2_20251001T122353Z.json.gz"}],"eventType":"AwsApiCall","managementEvent":false,"recipientAccountId":"231
307122651","eventCategory":"Data","tlsDetails":{"tlsVersion":"TLSv1.3","cipherSuite":"TLS_AES_128_GCM_SHA256","clientPro
videdHostHeader":"plainbit-s3.s3.ap-northeast-2.amazonaws.com"}}
```

[Figure71] S3 bucket object encryption event via AWS CLI command observed in CloudTrail

[Table173] Key field details of the object encryption event in the S3 bucket via the AWS CLI command

Field	Key Field Details
Object encryption in S3 bucket via AWS CLI command Object encryption in S3 bucket	• userIdentity - type: IAMUser - arn: arn:aws:iam::231307122651:user/acces.admin - userName: acces.admin • eventTime: 2025-10-02T04:23:36Z • eventSource: s3.amazonaws.com • eventName: CopyObject • awsRegion: ap-northeast-2 • sourceIPAddress: 85.203.21.7 • userAgent: aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.1 os/windows#11 md/arch#amd64 lang/python#3.13.7 md/pyimpl#CPython m/G,Z,b,E cfg/retry-mode#standard md/installer#exe md/prompt#off md/command#s3.cp • requestParameters - bucketName: plainbit-s3 - x-amz-server-side-encryption-customer-algorithm: AES256 • additionalEventData - SSEApplied: SSE_C

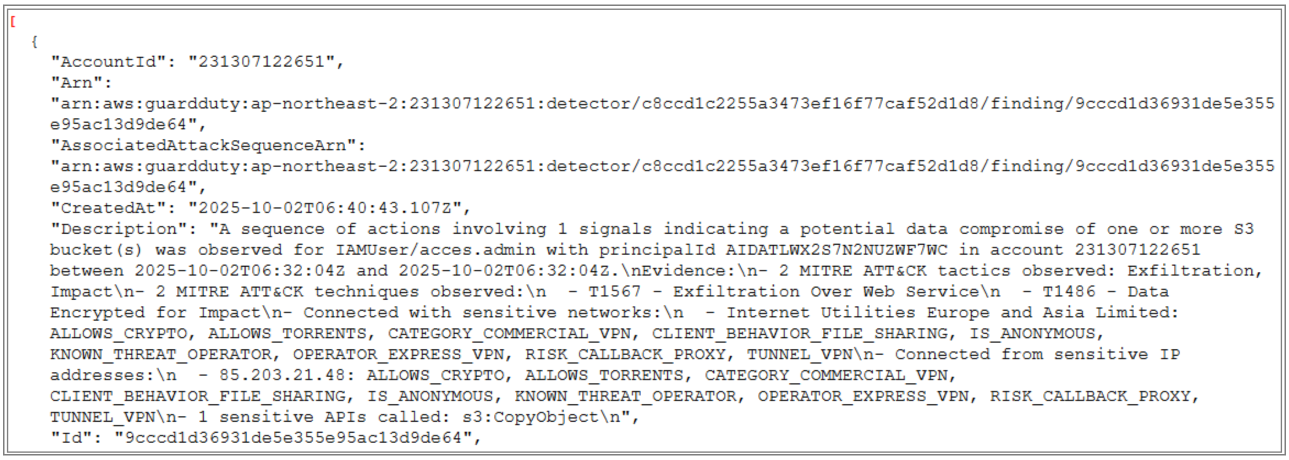
bitParser analysis results also confirm this behavior in the file as follows.



Mitre ATT&CK	eventTime	eventTimeLocal	awsRegion	eventName	eventSource	sourceIpAddress	userAgent	userIdentity.type	userIdentity.userName
Exfiltration	2025-10-01 12:40:36+00:00	2025-10-01 21:40:36	ap-northeast-2	CopyObject	s3.amazonaws.com	222.99.52.250	[Mozilla/5.0 (Windows NT 10.0; Win64;	Root	
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.7	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.44	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.36	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.11	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.16	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.5	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.12	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.59	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.67	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:36+00:00	2025-10-02 13:23:36	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.53	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.44	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.12	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.36	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.36	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.59	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.16	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.11	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.5	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.16	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.11	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.16	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.11	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.12	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.44	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.53	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.7	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.7	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.59	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin
Exfiltration	2025-10-02 04:23:37+00:00	2025-10-02 13:23:37	ap-northeast-2	CopyObject	s3.amazonaws.com	85.203.21.59	[aws-cli/2.31.5 md/awscrt#0.27.6 ua/2.	IAMUser	acces.admin

[Figure72] Object encryption events within the S3 bucket confirmed in the bitParser analysis results file

Additionally, when using GuardDuty, you can observe that it detects events when objects in an S3 bucket are encrypted.



```
[
  {
    "AccountId": "231307122651",
    "Arn":
    "arn:aws:guardduty:ap-northeast-2:231307122651:detector/c8ccd1c2255a3473ef16f77caf52d1d8/finding/9cccd1d36931de5e355
    e95ac13d9de64",
    "AssociatedAttackSequenceArn":
    "arn:aws:guardduty:ap-northeast-2:231307122651:detector/c8ccd1c2255a3473ef16f77caf52d1d8/finding/9cccd1d36931de5e355
    e95ac13d9de64",
    "CreatedAt": "2025-10-02T06:40:43.107Z",
    "Description": "A sequence of actions involving 1 signals indicating a potential data compromise of one or more S3
    bucket(s) was observed for IAMUser/acces.admin with principalId AIDATLWX2S7N2NUZWF7WC in account 231307122651
    between 2025-10-02T06:32:04Z and 2025-10-02T06:32:04Z.\nEvidence:\n- 2 MITRE ATT&CK tactics observed: Exfiltration,
    Impact\n- 2 MITRE ATT&CK techniques observed:\n  - T1567 - Exfiltration Over Web Service\n  - T1486 - Data
    Encrypted for Impact\n- Connected with sensitive networks:\n  - Internet Utilities Europe and Asia Limited:
    ALLOWS_CRYPTO, ALLOWS_TORRENTS, CATEGORY_COMMERCIAL_VPN, CLIENT_BEHAVIOR_FILE_SHARING, IS_ANONYMOUS,
    KNOWN_THREAT_OPERATOR, OPERATOR_EXPRESS_VPN, RISK_CALLBACK_PROXY, TUNNEL_VPN\n- Connected from sensitive IP
    addresses:\n  - 85.203.21.48: ALLOWS_CRYPTO, ALLOWS_TORRENTS, CATEGORY_COMMERCIAL_VPN,
    CLIENT_BEHAVIOR_FILE_SHARING, IS_ANONYMOUS, KNOWN_THREAT_OPERATOR, OPERATOR_EXPRESS_VPN, RISK_CALLBACK_PROXY,
    TUNNEL_VPN\n- 1 sensitive APIs called: s3:CopyObject\n",
    "Id": "9cccd1d36931de5e355e95ac13d9de64",
```

[Figure73] Partial content of the S3 bucket object encryption event detected via GuardDuty

7. Research Findings

This study proposed and demonstrated a DFIR data collection and analysis framework to support efficient incident response and analysis in AWS cloud environments. Given the cloud architecture's inherent limitations on physical access and the distributed nature of data across services, this research focused on reproducible evidence collection and timeline-based behavioral identification as core objectives.

The collection procedures, analysis techniques, and tools derived from this research can significantly enhance the standardization and practical applicability of DFIR execution in AWS cloud environments. The integrated collection structure combining command-based, log-based, and forensic image data, tactical event mapping, and the combination of automated log parsing and visualization tools can provide tangible assistance to incident responders in ensuring incident reproducibility and analysis reliability. Key achievements are as follows.

1) Establishment of a DFIR Data Collection Framework

Considering the structural constraints of the AWS environment, collection types were categorized into the following three types, and collection items and methods for each category were systematized.

◉ Command-Based Collection

This method involves directly issuing commands to instances and server resources within the AWS environment to obtain system information, configuration settings, log files, etc. It serves as a procedure to rapidly assess the system status and security configuration during an incident. This research established a command-based collection framework utilizing AWS CLI, AWS Systems Manager (SSM), and Prowler.

◉ Log-based Collection

This method collects operational logs generated by AWS services, serving as key evidence for identifying indicators at each attack stage. This study defined collection paths, key log fields, and analysis points for CloudTrail, VPC Flow Logs, S3 Server Access Log, CloudWatch Logs, GuardDuty Findings, WAF Log, and other logs.

◉ Forensic Image Collection

This method involves securing snapshots of compromised instances. For EC2/EKS Worker Nodes, we established a plan for creating EBS snapshots.

2) DFIR Analysis Approach Derivation and Tool Development

During the analysis phase of cloud DFIR, the following steps were performed to investigate attack activities and reconstruct them in a timeline format based on the collected data:

● Development of Tactics-Based Analysis Framework and Cheat Sheet

We developed an AWS DFIR Cheat Sheet that maps events frequently observed in CloudTrail, VPC Flow, S3 Access Log, etc., to specific tactics referenced from MITRE ATT&CK. The Cheat Sheet consolidates the meaning of each event, types of exploitation in attacks, events likely to be associated during an incident, and key log columns, enabling its use as standardized interpretation guidelines for analysts.

● Development of Analysis Tool (bitParser for AWS Log)

We implemented a tool that takes CloudTrail, VPC Flow, and S3 Access Log as input, normalizes (flattens) the logs, and automatically identifies and visualizes events by tactic. This tool is designed to enhance the efficiency of initial analysis by detecting key attack indicators and presenting events that analysts should prioritize reviewing, while also reducing discrepancies in the log interpretation process.

3) Scenario-Based Effectiveness Validation

We validated the proposed collection and analysis framework by constructing an AWS breach scenario simulating a ransomware attack. The overall attack behavior of the incident could be primarily analyzed from CloudTrail logs, while network activities such as internal movement could be identified through VPC Flow Logs. Furthermore, applying the bitParser tool developed in this study demonstrated a clear efficiency improvement compared to manual analysis during the process of automatically identifying and prioritizing tactics-based threat events and constructing timelines.

8. Conclusion and Future Research

This study proposes a DFIR data collection and analysis framework for incident response in AWS cloud environments and validated its effectiveness through a ransomware incident scenario. Its significance lies in establishing a reproducible data acquisition system and automated analysis foundation within AWS cloud environments.

Unlike existing on-premises-centric DFIR research, which was limited to individual logs or tools, this study distinguishes itself by presenting an integrated 'collection–analysis–tooling' framework that considers the structural constraints of cloud services. Systematizing collection procedures based on command-based, log-based, and forensic image-based approaches, and automating tactic-based log analysis, demonstrated the standardization potential and practical applicability for cloud incident response.

Furthermore, the research established a foundation enabling incident responders to consistently secure logs and system data within AWS environments and rapidly reconstruct attack sequences. This expands upon existing on-premises DFIR-centric research and presents the theoretical foundation for a DFIR framework specialized for cloud environments.

Conducted primarily within a single AWS environment, this research did not fully reflect the diverse operational models across the broader cloud landscape. Therefore, future studies require the following enhancements and expansions:

First, the need for expansion to multi-cloud environments

While this research was limited to the AWS environment, log formats and security architectures differ across heterogeneous clouds like Azure and GCP. Consequently, research is needed on a standardized DFIR framework capable of integrating and analyzing the log structures of each platform.

Second, Advancement of AI-Based Anomaly Detection

The current proposed system is configured to identify attack behaviors at the tactic-based event mapping level. However, future development should advance it into an intelligent analysis model that automatically detects and classifies attack behaviors by applying machine learning and artificial intelligence techniques.

Third, Expansion into a Real-Time Response System

Researching a real-time response orchestration structure that automates detection–isolation–evidence preservation using native services like AWS EventBridge, Step Functions, and Lambda will enhance the speed and consistency of incident response.

References

Number	Reference
1	Google Cloud, "M-Trends 2025 Report", https://cloud.google.com/security/resources/m-trends?hl=ko, May 27, 2025.
2	MITRE ATT&CK, “Enterprise - Cloud Matrix”, attack.mitre.org/matrices/enterprise/cloud/, 2025.04.25.
3	AWS, “Shared responsibility in the cloud”, learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility, 2024.09.29.
4	AWS, “The AWS Security Reference Architecture”, https://docs.aws.amazon.com/en_us/prescriptive-guidance/latest/security-reference-architecture/architecture.html, October 13, 2025.
5	Chris Champa, "What Is Cloud Incident Response?", https://www.wiz.io/academy/cloud-incident-response, July 14, 2025.
6	CSA, “Top Threats to Cloud Computing 2024”, https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-2024, August 5, 2024.
7	Google Cloud, “M-Trends 2023 Report”, https://services.google.com/fh/files/misc/m_trends_2023_report.pdf, April 18, 2023.
8	Daniel Leussink and Kantaro Komiya, “More than 2 million Toyota users face risk of vehicle data leak in Japan”, https://www.reuters.com/business/autos-transportation/toyota-flags-possible-leak-more-than-2-mln-users-vehicle-data-japan-2023-05-12/?ref=thestack.technology, May 12, 2023.
9	jumpcloud, “[Security Update] June 20 Incident Details and Remediation”, https://jumpcloud.com/blog/security-update-june-20-incident-details-and-remediation, 2023.09.07.
10	Pierluigi Paganini, “DARKBEAM LEAKS BILLIONS OF EMAIL AND PASSWORD COMBINATIONS”, https://securityaffairs.com/151566/security/darkbeam-data-leak.html, September 27, 2023.
11	Ionut Arghire, “Mercedes Source Code Exposed by Leaked GitHub Token”, https://www.securityweek.com/leaked-github-token-exposed-mercedes-source-code/, January 31, 2024.
12	AWS, “AWS Security Incident Response Guide”, https://docs.aws.amazon.com/pdfs/whitepapers/latest/aws-security-incident-response-guide/aws-security-incident-response-guide.pdf, August 15, 2025.
13	AWS, "What is Amazon GuardDuty?", https://docs.aws.amazon.com/en_us/guardduty/latest/ug/what-is-guardduty.html, 2025.10.14.
14	AWS, "What is Amazon CloudWatch Logs?", https://docs.aws.amazon.com/en_us/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html, October 14, 2025.
15	AWS, "What is Amazon Detective?", https://docs.aws.amazon.com/en_us/detective/latest/userguide/what-is-detective.html, October 14, 2025.
16	AWS, "What is Amazon Athena?", https://docs.aws.amazon.com/en_us/athena/latest/ug/what-is.html, October 10, 2025.
17	AWS, “Introduction to AWS Security Hub”, https://docs.aws.amazon.com/en_us/securityhub/latest/userguide/what-is-securityhub-v2.html, October 14, 2025.
18	AWS, “What is AWS Systems Manager?”, https://docs.aws.amazon.com/en_us/systems-manager/latest/userguide/what-is-systems-manager.html, October 14, 2025.
19	AWS, “What is Amazon Macie?”, https://docs.aws.amazon.com/en_us/macie/latest/user/what-is-macie.html, October 14, 2025.
20	AWS, “What Is AWS Config?”, https://docs.aws.amazon.com/en_us/config/latest/developerguide/WhatIsConfig.html, 2025.10.14.
21	AWS, “What is Amazon Inspector?”, https://docs.aws.amazon.com/en_us/inspector/latest/user/what-is-inspector.html, October 14, 2025.
22	AWS, “What is AWS CloudFormation?”, https://docs.aws.amazon.com/en_us/AWSCloudFormation/latest/UserGuide/Welcome.html, October 14, 2025.
23	prowler-cloud, "prowler", https://github.com/prowler-cloud/prowler, 2025.09.30.
24	AWS, “AWS Shield”, https://docs.aws.amazon.com/en_us/waf/latest/developerguide/shield-chapter.html, 2025.10.14.
25	aws-samples, “aws-incident-response-playbooks”, https://github.com/aws-samples/aws-incident-response-playbooks, July 8, 2025.

Number	Reference
26	aws-samples, “aws-customer-playbook-framework”, https://github.com/aws-samples/aws-customer-playbook-framework/tree/main/docs, 2025.08.05.
27	aws-samples, “aws-incident-response-playbooks-workshop”, https://github.com/aws-samples/aws-incident-response-playbooks-workshop, 2024.02.20.
28	AWS, “What Is AWS CloudTrail?”, https://docs.aws.amazon.com/en_us/awscloudtrail/latest/userguide/cloudtrail-user-guide.html, 2025.10.14.
29	AWS, “Logging IP traffic using VPC Flow Logs”, https://docs.aws.amazon.com/en_us/vpc/latest/userguide/flow-logs.html, October 14, 2025.
30	AWS, “Enabling Amazon S3 server access logging”, https://docs.aws.amazon.com/en_us/AmazonS3/latest/userguide/enable-server-access-logging.html, October 14, 2025.
31	AWS, “Monitoring Amazon RDS log files”, https://docs.aws.amazon.com/en_us/AmazonRDS/latest/UserGuide/USER_LogAccess.html, October 14, 2025.
32	AWS, “Logging AWS WAF protection pack (web ACL) traffic”, https://docs.aws.amazon.com/en_us/waf/latest/developerguide/logging.html, October 14, 2025.