Event Log Analysis Framework based on the ATT&CK Matrix in Cloud Environment

Yeeun Kim

Department of Future Convergence Technology Engineering

The Graduate School of Sungshin Women's University

Event Log Analysis Framework based on the ATT&CK Matrix in Cloud Environment

A Master's Thesis
Submitted to the
Graduate School of Sungshin Women's University
in partial fulfillment of the requirements
for the degree of
Master of Future Convergence Technology
Engineering

Yeeun Kim

NOV, 2023

This is to certify that we have examined the
Master's Thesis of
Yeeun Kim
Submitted to Department of Future Convergence
Technology Engineering

Approved as to style and content:

Thesis Advisor ..... Seongmin Kim

Committee Chairman ..... Wonhyung Park

Committee Member ..... Il-Gu Lee

The Graduate School of Sungshin Women's University

Abstract

Event Log Analysis Framework based on the ATT&CK Matrix in Cloud Environment

Yeeun Kim
Department of Future Convergence Technology Engineering
Graduate School of
Sungshin Women's University

With the increasing trend of Cloud migration, security threats in the Cloud computing environment have also experienced a significant increase. Consequently, the importance of efficient incident investigation through log data analysis is being emphasized. In Cloud environments, the diversity of services and ease of resource creation generate a large volume of log data. This results in difficulties determining which events to investigate when an incident occurs, and examining all the extensive log data requires considerable time and effort. Therefore, a systematic approach for efficient data investigation is necessary.

CloudTrail, the Amazon Web Services(AWS) logging service, collects logs of all API call events occurring in an account. However, CloudTrail lacks insights on which logs to analyze in the event of an incident. This paper proposes an automated analysis framework that integrates Cloud Matrix and event information for efficient incident investigation, enabling

simultaneous examination of user behavior log events frequency and attack information. This is expected to contribute to Cloud incident investigations by efficiently identifying critical events based on the ATT&CK Framework.

Contents

Table Contents

Figure Contents

Ⅰ. Introduction

As the demand for non-contact services rapidly increases, companies change how they manage their IT infrastructure. Moving away from On-Premise, where network and server assets are physically configured, there is an accelerating shift towards Cloud, where IT infrastructure is obtained as needed and set up in a virtual environment. According to Gartner research, spending on public Cloud services is projected to increase from $491 billion in 2022 to $597.3 billion in 2023[1]. Due to the high cost-efficiency of flexible operation, many corporate workloads are migrating to the Cloud. However, in the Cloud environment, security incidents frequently occur due to environmental configuration and setting errors caused by user mistakes. In fact, in October 2021, a Cloud configuration error by Facebook led to a complete shutdown for over six hours for 2.9 billion users of Facebook, WhatsApp, and Instagram[2]. To analyze such incidents in the Cloud, it is essential to analyze the logs containing events and related details that occurred at the time of the incident[3].

However, In the Cloud environment, resource creation, such as S3 for data storage and WAF for web traffic detection and blocking, is more straightforward than on-premise, leading to a larger volume of logs being generated quickly. This necessitates substantial storage resources if these logs are to be stored long-term.

Also, Unlike the On-Premises environment, where incident investigation methodologies are well-established, Cloud environments lack guidelines on which logs to analyze during a specific incident. This makes identifying related incidents within the vast volume of logs difficult. Therefore, the classification system for log events is required to efficiently investigate the increasing security incidents in cloud environments. This will enable rapid identification of events necessary for incident analysis and investigation, reducing time and procedures.

Commercial Cloud service providers offer logging services for incident response and detection of security threats to Cloud instances, with notable examples including Amazon's CloudTrail[4] and Microsoft's Activity Log[5]. Among them, AWS CloudTrail, which holds the highest market share in the Cloud industry, can track resource and service actions related to API calls in an account and store log records in the S3(Simple Storage Service) bucket. However, while CloudTrail records account and service activities, it does not provide insights for determining key events in specific incidents. Therefore, additional tools or methods are needed for incident analysis. AWS CloudWatch is a service for monitoring various applications in the Cloud environment[6], collecting events occurring in AWS, and providing alert functions for set-value violations. For instance, it can detect anomalies and generate alerts when the frequency of certain API call events exceeds normal ranges or unusual patterns are observed. However, setting threshold values can be challenging without sufficient information on crucial eventName, making it difficult for non-experts to determine alert criteria.

This paper aims to establish a log event classification system for incident analysis in Cloud environments. eventName in CloudTrail's logging records indicates the events that occurred and is crucial in identifying the operations. Accordingly, we propose a visualization Event Log analysis framework that links eventName with the ATT&CK Matrix. The process follows: First, an environment is set up to establish the classification system to collect logs from various sources. Subsequently, the identified eventName and related service information from the logs are linked to the Cloud Matrix, and all data is stored in an analysis database. This approach makes it possible to intuitively assess the volume of event occurrences and identify event information based on Tactics. The proposed log analysis framework demonstrates the ease of identifying and extracting necessary event and resource-related log data from a large volume of log data in the event of a specific incident. Through this, The aim is to reduce the time for incident log analysis and investigation processes in Cloud environments.

The remainder of this paper is organized as follows. Section Ⅱ explains the security threats in Cloud environments and describes CloudTrail, which is a primary topic of the research. Section Ⅲ describes previous studies about Cloud Log Analysis. Section Ⅳ explains our goal and describes the structure and operational flow of the proposed event log analysis framework. Section Ⅴ, The proposed framework's performance is evaluated to demonstrate its usability and to verify its detection coverage. Finally, Section Ⅵ presents the discussion and future research direction and concludes the paper.

Ⅱ. Background

This section examines the major security threats in Cloud environments, emphasizing the importance of log analysis in addressing these threats. Additionally, it explains the concept and principles of CloudTrail, a vital topic of this research.

1. Security Threats in Cloud Environments

Cloud Computing is a service that allows users to utilize computing resources such as storage and applications on-demand[7]. This enables efficient and flexible use of IT resources, providing numerous benefits to businesses and individuals. With no need for physical installation space and direct infrastructure setup, it allows for adjusting required resources and operational hours, increasing flexibility, cost-effectiveness, and organizational efficiency. However, with increased convenience and accessibility comes a rise in security threats, leading to frequent security incidents in virtual environments. In contrast to technical threats like DDoS in On-Premises environments, many administrative security threats occur due to credentials, access, and management issues in Cloud environments, where such attacks are more prevalent. According to the 'Future of Cloud and Security(2020)' report by the National Information Society Agency(NIA) and the 'Top Threats to Cloud Computing(2022)' report by the Cloud Security Alliance(CSA), it can be observed that most recent incidents in Cloud environments are attributed not to

external attackers, but rather to administrative security threats resulting from mistakes made by Cloud users or operators[8,9]. Table 1 presents the primary security threats in a Cloud environment as provided by CSA. It can be observed that administrative threats, such as inadequate ID, access, key management, and insecure interfaces and APIs, are ranked higher than technical causes like DDoS and system vulnerabilities. Administrative security threats in a Cloud environment manifest as issues related to authorization, service and application configuration, resource changes, and more. To analyze and investigate such incidents, it is crucial to analyze logs generated by API calls within the Cloud. AWS CloudTrail is a service that records all log data related to API calls, enabling the collection of essential log data for addressing security incidents within the Cloud. However, the service does not provide the capability to identify critical logs, necessitating additional means for prompt event identification. For example, a classification system is required to swiftly identify events that require analysis, such as when an unauthorized user attempts unauthorized access due to insufficient permissions or when incidents occur due to configuration errors.

The Proposed classification system significantly improves incident analysis and investigation efficiency and speed. It facilitates swift assessments of behavioral events, like changes in resources and instances of unauthorized user access. As a result, this framework enhances the effectiveness of analyzing and investigating security incidents.

TABLE 1. Top 3 Cloud Threats

Rank	Average Score	Issue Name
1	7.729927	Insufficient ID, Credential, Access and Key Mgt, Privileged Accounts
2	7.592701	Insecure Interfaces and APIs
3	7.424818	Misconfiguration and Inadequate Change Control

2. AWS CloudTrail

CloudTrail is one of the critical tools for incident response in the AWS environment, recording and monitoring all API activities originating from an account[10]. It supports event-based logging for the AWS Management Console, AWS SDK(Software Development Kit), AWS CLI(Command Line Interface), and account activities, categorized into three types: Management event, Data event, and Insight event[4]. Management events refer to operations that involve managing or configuring resources and services originating from users or roles with AWS IAM(Identity and Access Management) permissions. For example, this includes actions like creating EC2(Elastic Compute Cloud) Instances and S3(Simple Storage Service) buckets. Data events record operations performed on resources and within resources[10]. These events are beneficial for tracking activities related to data changes or access within resources, such as deleting or modifying files stored in S3 buckets. Data events are initially disabled by default, and resources or types for which logs need to be collected must be added. Insight events provide information about abnormal activities and are logged when there is a deviation in API usage from typical patterns. In this paper, we focus on management events and data events, as we analyze log data through regular logging rather than logs detected as signs of anomalies.

The information available in log data generated by CloudTrail is diverse. It consists of various fields, including ‘eventTime’, which indicates the date and time(UTC) when the API call occurred;

‘userIdentity’, which allows verification of the IAM credentials of the entity that initiated the event; ‘eventSource’, identifying the service where the request was made; ‘eventName’, specifying the name of the requested API operation; ‘awsRegion’, indicating the region where the request occurred; and error-related details such as ‘errorCode’ and ‘errorMessage’[11]. An efficient log analysis can be performed using these field values to filter and analyze the data. For instance, using ‘userIdentity’ and ‘eventName’, one can determine the entity that triggered the event and the specific API call that generated the log event. A field such as ‘eventTime’ and ‘awsRegion’ also identify when the event occurred and in which region the respective API was invoked.

In practice, the logs recorded in CloudTrail are structured as depicted in Figures 1 and 2. These logs are part of the event logs related to the termination of EC2 instances and can be identified by the ‘TerminateInstances’ value in the ‘eventName’. Figure 1 represents an instance termination event initiated through an IAM account, while Figure 2 illustrates instance termination via the Root account. Therefore, it can be observed that the ‘type’ value in ‘userIdentity’ is different from ‘IAMUser’ and ‘Root’.

Understanding the necessary fields and data types for incident investigation reduces excessive noise and enables quick data filtering and analysis. ‘eventName’, which represents API call information originating from resources and services, is crucial in security monitoring and event analysis. This is because it facilitates the identification of the type of operation performed. For example, by using eventName such as

'ConsoleLogin', 'StartInstance' or 'CreateUser', one can retrieve API call information related to specific activities, including login history, service activity time, accessed resource types, and the creation of users by non-administrator accounts.

This paper aims to analyze 'eventName', considered a key field for incident investigation among various fields, and demonstrate an automated event analysis framework that correlates with the ATT&CK Matrix, providing utility for Cloud incident investigations.

FIGURE 1. TerminateInstances Event Log(IAM)

FIGURE 2. TerminateInstances Event Log(Root)

Ⅲ. Related Works

1. Cloud Log Analysis

The widespread adoption of Cloud computing has led to increased cybersecurity incidents within virtual environments. Consequently, there is a growing emphasis on the need for research in Cloud log analysis to understand the specifics of these incidents. However, managing and analyzing it has become increasingly challenging with the increasing volume and diversity of Cloud log data[12]. While research in Cloud log analysis is ongoing, numerous issues still need to be addressed. Considering the challenges mentioned in prior research regarding Cloud log analysis, this study aims to propose an analysis framework to address these challenges.

Saad Alqahtany and three others identified challenges when performing forensics in a Cloud-based environment. They emphasized the dependence on Cloud Service Providers(CSPs), data collection and integrity issues when investigating Cloud log data in Infrastructure as a Service(IaaS) environments, and the lack of adequate Cloud forensic tools. In response, they proposed an independent Forensic Acquisition and Analysis System(FAAS) to automate and simplify digital forensics in the Cloud without needing involvement or cooperation from Cloud providers. However, this proposal remains at a conceptual level, and its practical viability, as well as its effectiveness and efficiency without relying on

CSPs, has yet to be experimentally validated[13]. The authors described the AWS logging service, CloudTrail, which can provide forensic sources to users. However, they argued that additional methods are necessary to analyze logs effectively.

Suleman Khan and seven others reviewed Cloud Log Forensic(CLF) technology and highlighted various Cloud log data analysis issues. They emphasized the dependence on Cloud Service Providers(CSPs) and the diversity and complexity of log data. They underscored the importance of CLF in effectively collecting and managing log data generated in Cloud environments[14]. However, they did not provide specific guidelines on which log types to include for analysis and how to collect and analyze various logs effectively. Due to the considerable amount and complexity of logs generated in Cloud environments, this study aims to provide criteria for efficiently storing, analyzing, and categorizing them.

Kenny Awuson-David and four others proposed the Blockchain Cloud Forensic Logging(BCFL) framework using the Design Science Research Methodological(DSRM) approach for evidence acquisition in Cloud environments[15]. However, introducing blockchain technology may complicate the analysis process due to factors like maintaining distributed ledgers and transaction verification, potentially impacting the performance and efficiency of Cloud environments. To address this, we propose a straightforward event analysis method that can be easily applied. This aims to enhance the accessibility of log analysis, allowing for effective utilization in various scenarios while minimizing technological complexity and enabling straightforward log analysis. This study proposes an event

classification system and log analysis framework based on the AWS logging service, CloudTrail. CloudTrail serves as a means to address CSP dependencies, log data collection, and access issues. It records crucial information about API calls, including the time, name, and request parameters of activities performed within an account. This enables access to log data without CSP support and maintains data integrity through encryption to prevent unauthorized modifications and reads. Therefore, it can be effectively utilized as valuable data for incident investigations[16].

The Proposed framework introduces a method to utilize AWS CloudTrail for collecting event logs and linking them with the ATT&CK Matrix for visual analysis. MITRE has systematized the techniques used in attacks as TTP(Tactics, Techniques, and Procedures) and developed the ATT&CK Matrix[17]. This matrix summarizes attackers' tactics, techniques, and procedures based on actual attack cases, making it helpful in identifying and detecting malicious activities. We have designed an event analysis framework based on the ATT&CK Matrix for accurate and rapid log event analysis within the AWS environment. This framework will provide vital resources for future incident investigations in the Cloud, contributing to identifying vast log data rapidly. In addition, streamlining the analysis process of log events is expected to improve Cloud log analysis and investigation methods.

Ⅳ. Cloud Event Log Analysis Framework

Efforts are being made to classify threats from the perspective of security incidents. However, the lack of established criteria for event mapping from generated log data makes determining an EntryPoint for analysis challenging. As the volume of analysis data increases, this issue becomes more pronounced, highlighting the need for an efficient solution. To address this, we propose an event analysis framework based on Cloud Matrix[18], a standard framework for threat models in security incidents. This framework is designed to rapidly identify the objectives of attacks and key events[19]. MITRE has proposed the ATT&CK(Adversarial Tactics, Techniques, and Common Knowledge) Framework[20], which defines the behaviors and methodologies of actual attackers. This facilitates establishing a reliable, standardized system for classifying event logs[21]. Through this, it is possible to enhance the understanding of incidents that have occurred and improve the response capabilities in preparation for potential future threats. Figure 3 illustrates the system overview and data flow of the proposed Cloud log analysis framework.

The framework is composed of three main modules. The first module establishes an event classification system based on the ATT&CK Matrix. The second module is where user behavior log data is stored. The last is an event visualization analysis module that can visually check the analysis results through the two modules.

Figure 4 shows the DB schema that can analyze data information stored through each module. The first module, designed for storing and analyzing event classification system data, saves data in the 'eventName_attck'. The second module, intended for inputting and analyzing extracted user behavior log data, stores this data in the 'log' table. Users can extract data from the S3 bucket where logs are stored, save it in 'log', and then input this into the third module's framework for visible event analysis.

This process performs event log analysis automatically, allowing for intuitive identification of eventNames from complex log data. This will enable investigators to avoid manually identifying events, enhancing efficiency and saving time.

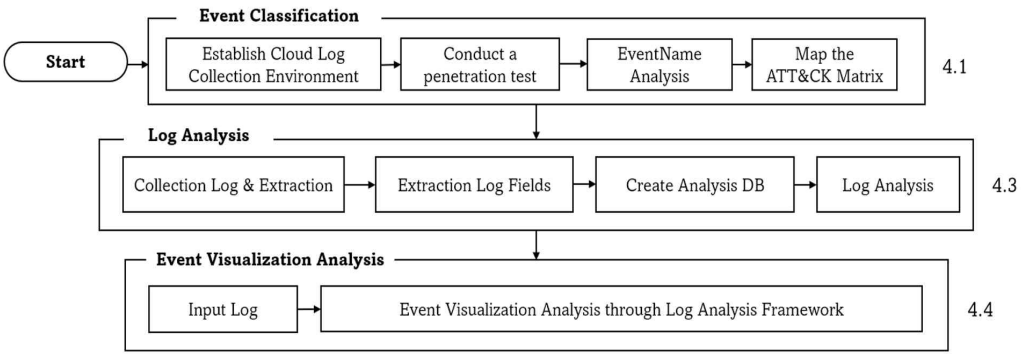

FIGURE 3. Event Log Analysis Framework Overview

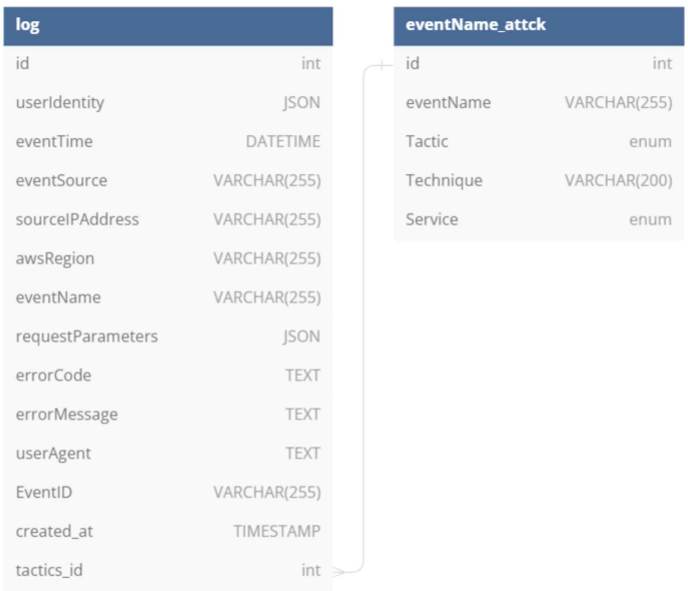

FIGURE 4. Analysis DB Schema

1. Event Classification System Module

1) Establish Cloud Log Collection Environment

This section describes the process of setting up an analysis environment for identifying the eventName of events that occur in the initial execution phase of the proposed framework. Logs are generated from various sources in a Cloud environment to track and monitor system, application, and network activities. The necessary environment setup to collect logs generated across the entire system, including user(IAM), network, and applications, is depicted in Figure 5[22].

The Amazon OpenSearch Service, used for collecting all log data in one place, aggregates logs from various sources to enable integrated analysis and provides the functionality to visualize them according to user needs. This is useful for real-time monitoring of events and intuitive classification of data.

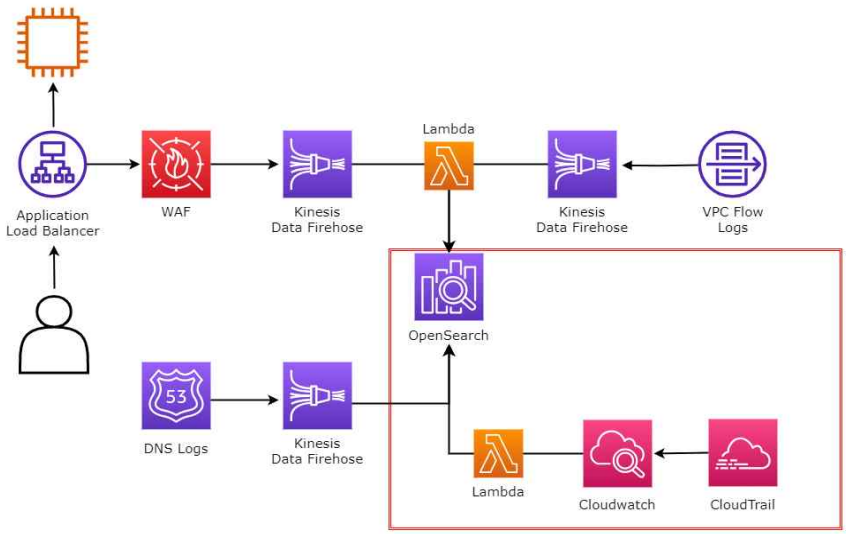

FIGURE 5. Log Collection/Analysis Environment

The first collection pathway is log collection through CloudTrail. This involves a process where log data is collected through CloudTrail, CloudWatch, Lambda, and then OpenSearch. CloudTrail monitors activities in an AWS account and records them as events, allowing for the collection of logs such as API calls related to activities and events and resource access policies based on IAM. In AWS, it is often necessary to provide authentication and authorization for resources and services to control access to resources effectively. The invoked Lambda must have permissions for OpenSearch to enable the transmission and integration of log data. The second is for logging information about public DNS queries received by Route 53, AWS's managed DNS(Domain Name System) service. This enables the identification of domain, date and time, IP, and other details. The data is collected through Route53

DNS Log and Kinesis Data Firehose into OpenSearch. Kinesis Data Firehose is utilized for real-time processing and storage of log data[23].

Additionally, it is possible to configure a collection path for logs generated by the WAF(Web Application Firewall), which detects and blocks web application attacks at Layer 7, and for VPC Flow Logs, which record network traffic log information concerning various packets transmitted and received within a VPC. Such environment configurations allow collecting and visualizing log events from multiple sources.

This paper focuses on collecting all API call records of an AWS account and analyzing log data, focusing on log collection based on CloudTrail.

2) Analysis Attack Time

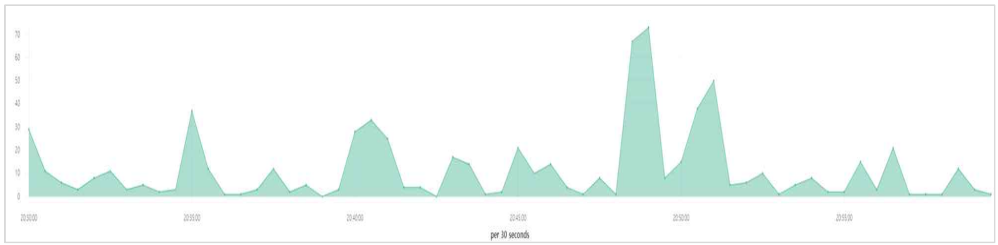

FIGURE 6. Analysis Attack Time

OpenSearch was configured to aggregate log collection and ensure visibility when establishing the environment. Through this, it becomes possible to detect surges in specific events, such as creating EC2 instances or console logins, and filter by the relevant time range. Figure 6 shows the visualization of Count values for eventName in OpenSearch using TSVB(Time Series Visual Builder) after a penetration test. TSVB is a valuable metric when visualizing activities over a specific period[27]. It was observed that most events occurred between 20:47 and 20:53, which was the actual time of the penetration test. This range was filtered to analyze based on the occurred eventNames. Such metrics can be effectively used in penetration tests and real incident scenarios to identify specific events that surge or occur at unexpected times.

Table 2 shows some events that can occur according to Tactics as part of the conducted attack actions[24,25,26]. The Stratus Red Team open-source project was utilized to classify these events. This is based on the MITRE ATT&CK framework and is a test library focused on simulating standard attack techniques in Cloud environments[24,25].

By utilizing this, it is possible to establish event analysis criteria based on various tactics. The simulatable Techniques are shown in Figure 7.

TABLE 2. Events of the attack stage

Tactic	Event
Initial Access (TA0001)	Console Login without MFA
	Suspicious SAML Activity
Execution (TA0002)	Launch Unusual EC2 instances
	Change EC2 Startup Shell Script
Persistence (TA0003)	Create an Access Key on an existing IAM User
	Create a IAM User with administrative permissions
	Add a Malicious Lambda Extension
	Overwrite Lambda Function's Code

TECHNIQUE ID	TECHNIQUE NAME	PLATFORM	MITRE ATT&CK TACTIC
aws.credential-access.ec2-get-password-data	Retrieve EC2 Password Data	AWS	Credential Access
aws.credential-access.ec2-steal-instance-credentials	Steal EC2 Instance Credentials	AWS	Credential Access
aws.credential-access.secretsmanager-retrieve-secrets	Retrieve a High Number of Secrets Manager secrets	AWS	Credential Access
aws.credential-access.ssm-retrieve-securestring-parameters	Retrieve And Decrypt SSM Parameters	AWS	Credential Access
aws.defense-evasion.cloudtrail-delete	Delete CloudTrail Trail	AWS	Defense Evasion
aws.defense-evasion.cloudtrail-event-selectors	Disable CloudTrail Logging Through Event Selectors	AWS	Defense Evasion
aws.defense-evasion.cloudtrail-lifecycle-rule	CloudTrail Logs Impairment Through S3 Lifecycle Rule	AWS	Defense Evasion
aws.defense-evasion.cloudtrail-stop	Stop CloudTrail Trail	AWS	Defense Evasion
aws.defense-evasion.organizations-leave	Attempt to Leave the AWS Organization	AWS	Defense Evasion
aws.defense-evasion.vpc-remove-flow-logs	Remove VPC Flow Logs	AWS	Defense Evasion
aws.discovery.ec2-enumerate-from-instance	Execute Discovery Commands on an EC2 Instance	AWS	Discovery
aws.discovery.ec2-download-user-data	Download EC2 Instance User Data	AWS	Discovery
aws.execution.ec2-launch-unusual-instances	Launch Unusual EC2 instances	AWS	Execution
aws.execution.ec2-user-data	Execute Commands on EC2 Instance via User Data	AWS	Execution
			Privilege Escalation
aws.exfiltration.ec2-security-group-open-port-22-ingress	Open Ingress Port 22 on a Security Group	AWS	Exfiltration
aws.exfiltration.ec2-share-ami	Exfiltrate an AMI by Sharing It	AWS	Exfiltration
aws.exfiltration.ec2-share-ebs-snapshot	Exfiltrate EBS Snapshot by Sharing It	AWS	Exfiltration
aws.exfiltration.rds-share-snapshot	Exfiltrate RDS Snapshot by Sharing	AWS	Exfiltration
aws.exfiltration.s3-backdoor-bucket-policy	Backdoor an S3 Bucket via its Bucket Policy	AWS	Exfiltration
aws.impact.s3-ransomware-batch-deletion	S3 Ransomware through batch file deletion	AWS	Impact
aws.impact.s3-ransomware-client-side-encryption	S3 Ransomware through client-side encryption	AWS	Impact
aws.impact.s3-ransomware-individual-deletion	S3 Ransomware through individual file deletion	AWS	Impact
aws.initial-access.console-login-without-mfa	Console Login without MFA	AWS	Initial Access
aws.persistence.iam-backdoor-role	Backdoor an IAM Role	AWS	Persistence
aws.persistence.iam-backdoor-user	Create an Access Key on an IAM User	AWS	Persistence
			Privilege Escalation
aws.persistence.iam-create-admin-user	Create an administrative IAM User	AWS	Persistence
			Privilege Escalation
aws.persistence.iam-create-user-login-profile	Create a Login Profile on an IAM User	AWS	Persistence
			Privilege Escalation
aws.persistence.lambda-backdoor-function	Backdoor Lambda Function Through Resource-Based Policy	AWS	Persistence
aws.persistence.lambda-overwrite-code	Overwrite Lambda Function Code	AWS	Persistence
aws.persistence.rolesanywhere-create-trust-anchor	Create an IAM Roles Anywhere trust anchor	AWS	Persistence
			Privilege Escalation

FIGURE 7. Stratus - Attack Techniques[24]

3) Analysis eventName

While AWS CloudTrail provides logging capabilities, it is challenging to determine which logs to analyze in the event of a specific incident. To address this, we aim to establish a classification system that can identify the attack techniques or patterns related to the events that occurred by linking eventName, which can determine what actions were performed with the ATT&CK Matrix. To achieve this, we conducted attack techniques categorized under each Tactic, such as Initial Access(TA0001) and Execution(TA0002) through Cloud-focused penetration test projects and publicly available rule sets, and classified the logged eventNames. So, We can establish a checklist-based logging review and framework by identifying security incident events based on attack tactics.

Figures 8 and 9 show the results of identifying the eventNames that occurred after the tests of the Execution and Credential Access stages, respectively, viewed through CloudTag. In Figure 8, Execution results, events such as creating an IAM without permissions to start EC2 instances, running instances, or inserting malicious scripts to execute instances with administrator privileges can be observed. Specifically, 'ModifyInstanceAttribute', which manages and updates the properties of an EC2 instance, and 'StopInstances' and 'StartInstances', which signify the termination and initiation of an instance, were identified. By filtering based on the time when a specific attack occurred and then reviewing the logged eventName, it is possible to identify the events necessary for Matrix mapping quickly.

FIGURE 8. eventName Analysis of Execution Events

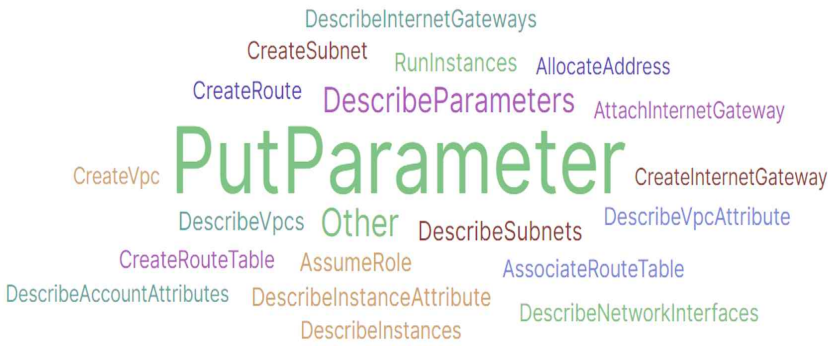

FIGURE 9. eventName Analysis of Credential Access Events

4) Map the ATT&CK Matrix

One of the essential steps in Cloud incident analysis is the rapid selection of necessary log events from a vast amount of data. This is crucial because it enables the use of accurate data for incident investigation and saves both data processing time and effort[28]. So, after conducting a penetration test in the Cloud environment and examining the recorded events along with rule sets, the eventName was classified based on the Cloud Matrix.

The MITRE ATT&CK Cloud Matrix systematically describes Tactics, which explain 11 types of attack techniques and behaviors, and it provides a categorized list of Techniques, which are methods of attacking to achieve each Tactic[29]. In this study, eventNames were were classified according to the tactics, and information about the Technique and the source service from which it originated was recorded.

Along with linking the ATT&CK Matrix and eventName, it is also essential to identify specific resources and service information where the event occurred. Identifying this allows for interpreting the cause of incidents through service-based searches. For example, if inappropriate access or changes to a specific resource are detected, these can be verified and investigated based on this information. By categorizing the eventName with the related resource and service event information, it becomes possible to conduct investigations and analyses according to the type of Cloud resource. Table 3 presents a classification system showing the resources and services linked to the ATT&CK Matrix and eventName mapping results, representing a portion of the total 483

events. This makes it easy to discern which invoked API information quickly relates to which attacks, tactics, resources, and services at a glance.

TABLE 3. Results of eventName and Matrix Mapping

eventName	Tactic	Technique	Service
Describe Task Definition	Persistence (TA0003)	Implant Internal Image (T1525)	ECS
CreateUser		Valid Accounts (T1078.004)	IAM
CreateAccessKey		Account Manipulation (T1098)	
ListBuckets	Discovery (TA0007)	Cloud Infrastructure Discovery(T1580)	S3

2. Statistical Analysis of the eventName

This is the result of statistical analysis that maps the 483 eventNames analyzed based on the ATT&CK Matrix. As evident from Figure 10, eventNames related to the Discovery(TA0007) and Persistence(TA0003) phases frequently occurred. The Discovery Tactic involves activities where attackers gather potentially exploitable information within the system. One of the primary activities of penetration testing tools is simulating the reconnaissance to uncover information about the target system's services, accounts, and infrastructure. Especially considering the emphasis of penetration testing tools on gathering various information to identify vulnerabilities in the target system and expand the scope of attacks, the increase in events related to these activities aligns with the objectives of such tools.

Next, the Persistence Tactic, which occurs frequently, involves the attacker employing various means and methods to maintain access to the system and evade detection. Examples of events in this phase include installing backdoors and escalating privileges. This can also be seen as a result of the essence of penetration testing. Penetration testing is intentionally performed to intrude into a system to assess its security strength or uncover potential vulnerabilities. Standard techniques such as simulating 'Persistence' or elevating privileges are frequently employed during this process.

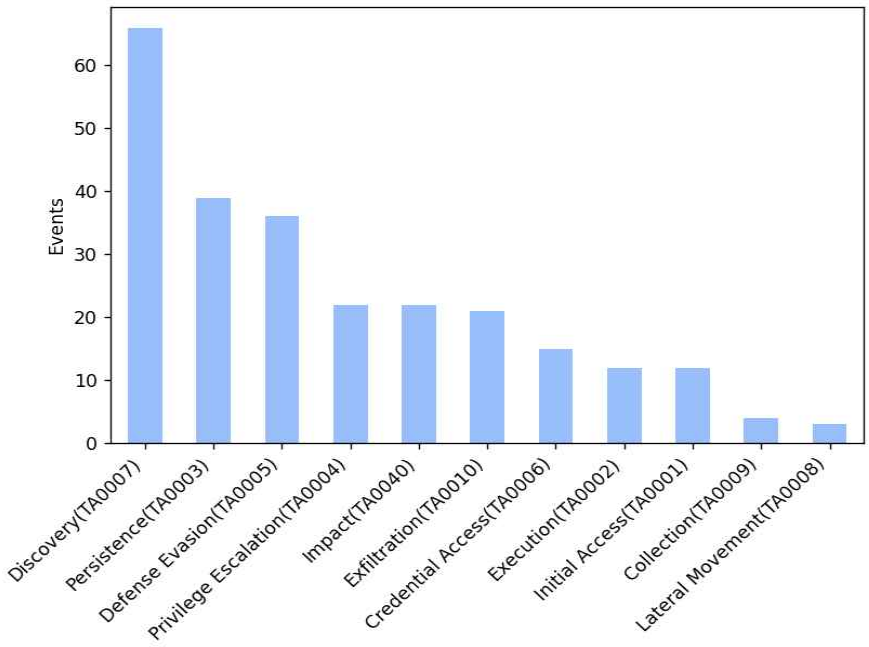

FIGURE 10. Results of eventName Occurrence

Figure 11 displays the frequency of services associated with mapped eventNames. The analysis reveals that EC2, IAM, and S3 domains exhibited the highest frequencies. This is attributed to the critical and vital nature of these services within AWS[30], as many applications and systems operate based on these services. The penetration testing tools employed also simulate vulnerability analysis and attacks on critical services, resulting in a high frequency of log events for these services.

These results are also related to the most frequently occurring threat category among Cloud security incident threat classifications, 'resource configuration management permissions and errors', as shown in Table 1.

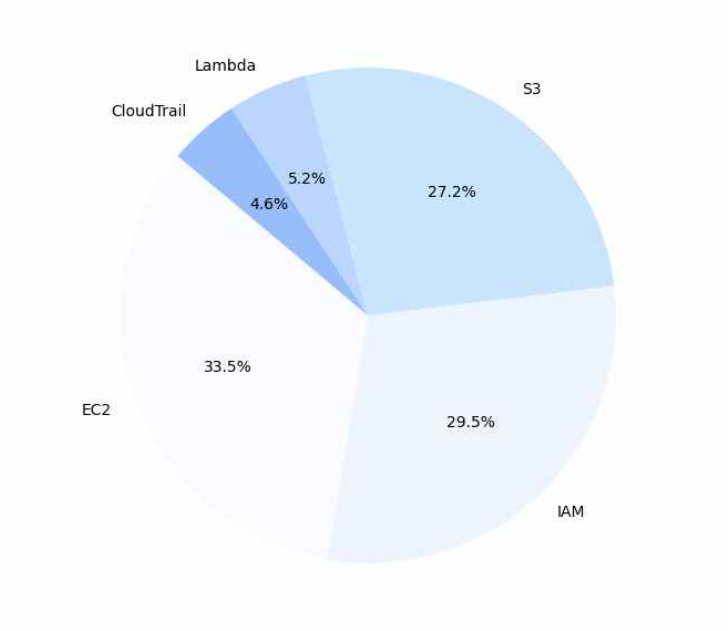

FIGURE 11. Results of Service Occurrence

Figure 12 represents the mapping results between services and Tactics in a heatmap, highlighting the most frequent combinations. The analysis reveals that Discovery(TA0007) with EC2 and Discovery(TA0007) with S3 are the most prevalent. Discovery represents activities where attackers collect information within the system, and EC2 and S3 are essential and critical services in AWS. Therefore, the frequent connection between these two services and the Discovery in the heatmap results indicate that the simulation primarily involved information-gathering activities using critical services. In other words, this shows that scenarios in which attackers access vital services within the system, such as EC2 and S3, to gather information for carrying out attacks have frequently occurred. These statistical results will provide valuable insights for security analysts to understand the correlation between events from penetration testing tools and actual threat activities. The statistical analysis of events and the Matrix can vary depending on the tools and methods used for identification. As additional tools and mapping efforts are conducted, the coverage of various event analyses based on Tactics is expected to increase.

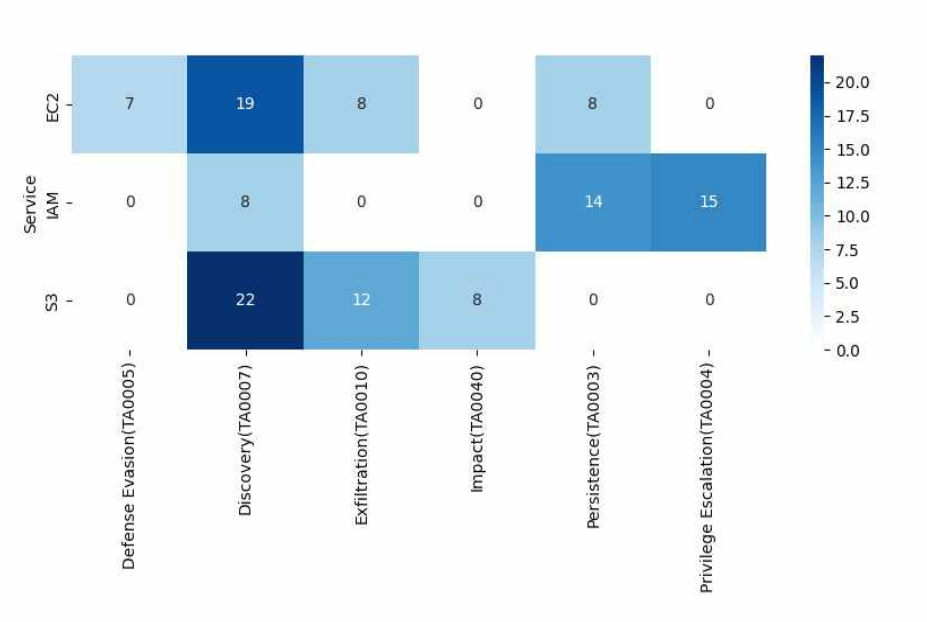

FIGURE 12. Combination Results of Service&Tactic

3. Log Analysis Module

This section describes the log storage and analysis module required for analyzing actual user behavior data in conjunction with the Module in section Ⅳ.1, which is classified based on the ATT&CK Matrix. This Module goes through the process, which entails collecting and extracting logs from the S3 log data storage, selecting log fields for analysis, and the database creation stage.

1) Collection/Extraction Log & Extraction Log Fields

The first step in the log analysis module is to collect and extract the data to be analyzed. CloudTrail can store events related to changes and

access to AWS resources in Amazon S3, which can be extracted in CSV or JSON format. In this study, JSON data is analyzed and stored in the analysis database for flexibility and scalability considerations. Next, it is necessary to identify the log fields for behavior analysis. In the proposed framework, the analysis fields are selected based on 4W1H(Who, When, Where, What, How) criteria to extract information efficiently. This allows for event representation of 'Who, When, Where, What, How' for behavior analysis while considering storage efficiency. This approach allows for efficient storage of log data. Table 4 describes the fields extracted for analysis. While a lightweight database was created by selecting log fields, accessing the original data for detailed analysis may be necessary. To facilitate this, an eventID field is added, representing a unique identifier for events, allowing the retrieval of the original log data by its location. Therefore, Even if not all fields are stored in the database, the eventID allows access to the log data.

However, a limitation of these log fields is the difficulty in integrating log analysis across various Cloud platforms like Microsoft Azure and GCP(Google Cloud Platform). This challenge arises due to differences in format and schema for logs in each Cloud platform. To collectively gather and analyze logs generated from AWS and Azure, we used a toolset based on the ELK(Elasticsearch, Logstash, Kibana) Stack to review the log data. It was found that, instead of normalizing the log data into a single format, they are collected in different formats according to each Cloud environment. This demonstrates that even with establishing an environment for managing logs in a unified manner,

practical incident analysis becomes challenging due to the different types and formats of logs occurring in heterogeneous environments. Therefore, standardizing log field names and applying a consistent format across platforms is crucial. This will enable effective handling of diversity and efficient analysis and management of log data.

We analyzed the OCSF(Open Cyber Security Schema Framework) framework to overcome these limitations. Table 4 shows the results matching the analyzed log fields from CloudTrail with equivalent fields in OCSF. OCSF provides a vendor-agnostic classification method through open-source frameworks, enabling improved data collection and analysis without the time-consuming normalization process[31]. As a result of the review, it was confirmed that 'eventName' corresponds to 'operation' and 'eventTime' corresponds to 'time'.

If such normalization standards are actively established for log analysis, it can save the time and effort required for data normalization when performing integrated analysis across various Cloud platforms.

TABLE 4. Analysis Log Table Fields

	Log Field	Description	OCSF
Who	user Identity	User information (user identifier, name, MFA authentication status, etc.)	unmapped [userIdentity.~]
When	event Time	Time of occurrence of an event or task	time
Where	event Source	Source or type of the event that occurred, or the type of task	service.name
	source IP Address	IP address from which the request was made	src_ endpoint.ip
	awsRegion	Region where the event or task occurred	cloud.region
What	eventName	Actual name of the event or task	operation
	request Parameters	Parameters of the API request performed in the respective event	unmapped. [requestParameters.~]
How	user Agent	Agent from which the request was made	http_request. user_agent
	error Code	Error code if the request returns an error	api.response.error
	error Message	Error description if the request returns an error	api.response. message
ID	eventID	Unique identifier for the event	-

2) Create Log Analysis DB

This is the process of storing and analyzing the analyzed information in a database. When configuring the database, establishing connections between tables through Foreign Keys(FK) allows for the effective integration of behavioral data. Figures 13 and 14 provide detailed information about the configured tables. The 'log_analysis' table stores the data extracted from the log fields mentioned in Table 4, while the 'eventname_attck' table stores the Tactic, Technique, and associated services matching the eventName.

The ATT&CK Matrix provides information on various attack techniques and tactics, enabling the identification of behavioral patterns and assessing the risks associated with them through the integration and in-depth analysis of log data. Therefore, by integrating the eventName-Matrix mapping information table with the log analysis table using the proposed framework, it becomes possible to obtain information that may be challenging to extract from the logs themselves, facilitating more effective security responses.

id	userIdentity	eventTime	eventSource	sourceIPAddress	awsRegion	eventName	requestParameters	errorCode	errorMessage	userAgent	EventID	created_at	tactics_id
25	{"arn": "arn...	2023-12...	s3.amazona...	210.125.93.124	us-east-1	ListObjects	{"Host": "s3.amazo...	NULL	NULL	[S3Consol...	fce76a...	2023-12-2...	450
26	{"type": "A...	2023-12...	s3.amazona...	cloudtrail.amaz...	us-east-1	PutObject	{"key": "AWSLogs/...	NULL	NULL	cloudtrail....	061a9c...	2023-12-2...	439
27	{"arn": "arn...	2023-12...	s3.amazona...	210.125.93.124	us-east-1	ListObjects	{"Host": "s3.amazo...	NULL	NULL	[S3Consol...	2a8b4e...	2023-12-2...	450
28	{"type": "A...	2023-12...	kms.amazon...	cloudtrail.amaz...	us-east-1	GenerateD...	{"keyId": "arn:aws:...	NULL	NULL	cloudtrail....	f0914a...	2023-12-2...	411
29	{"arn": "arn...	2023-12...	cloudtrail.a...	210.125.93.124	us-east-1	LookupEvents	{"maxResults": 50, ...	NULL	NULL	Mozilla/5.0....	20d047...	2023-12-2...	220
30	{"arn": "arn...	2023-12...	health.amaz...	210.125.93.124	us-east-1	DescribeEv...	{"filter": {"startTim...	NULL	NULL	AWS Inter...	ea1772...	2023-12-2...	451
31	{"arn": "arn...	2023-12...	s3.amazona...	210.125.93.124	us-east-1	ListBuckets	{"Host": "s3.amazo...	NULL	NULL	[S3Consol...	1243fd...	2023-12-2...	136
32	{"arn": "arn...	2023-12...	ec2.amazon...	210.125.93.124	us-east-1	DescribeRe...	{"regionSet": {}, "al...	NULL	NULL	Mozilla/5.0....	18abf5...	2023-12-2...	200

FIGURE 13. Log_analysis Table

id	eventName	Tactic	Technique	Service
4	ConsoleLogin	Initial Access(TA0001)	Valid Accounts(T1078)	IAM
470	UpdateSAMLProvider	Initial Access(TA0001)	Valid Accounts(T1078)	IAM
471	UpdateSAMLProvider	Defense Evasion(TA0005)	Use Alternate Authentication Material(T1550)	IAM
473	AssumeRoleWithSAML	Initial Access(TA0001)	Valid Accounts(T1078)	STS
474	AssumeRoleWithSAML	Defense Evasion(TA0005)	Use Alternate Authentication Material(T1550)	STS

FIGURE 14. eventName_attck Table

4. Event Log Analysis Framework

Figures 15 and 16 are GUI screens of the self-implemented ATT&CK Matrix-eventName-based log analysis framework. The GUI implemented using Tkinter provides an intuitive and user-friendly environment for users. Tkinter is Python's standard GUI library widely used for its ease of use and effectiveness in developing various GUI applications[32]. Furthermore, the heatmaps created using Matplotlib and Seaborn provide users with a visual event analysis tool, allowing them to visualize the frequency of events and tactics intuitively. The analysis framework helps gain insights into various events occurring in the Cloud environment and identify the entry points for investigation in diverse log data. This allows users to grasp event distributions and effectively conduct analyses intuitively. When the framework is executed, the x-axis displays the events that have occurred, while the y-axis represents the corresponding tactics for each event. Analyzing information from both axes allows users to quantitatively assess the frequency of event occurrences.

Additionally, users can select specific cells to obtain detailed information about the desired events. When a cell is selected for analysis, as shown in Figure 16, an area outside the heatmap provides detailed information about the selected cell. In this area, users can access information about the tactics and techniques associated with the selected event. Additional information such as userIdentity, eventTime, eventSource, sourceIPAddress, awsRegion, eventName, requestParameters, userAgent, errorCode, errorMessage, eventID is also included. This lets

users quickly access log data for specific events through the heatmap.

This event log analysis framework helps visually represent analyzed data, allowing for the intuitive identification and swift detection of threat events in Cloud environments.

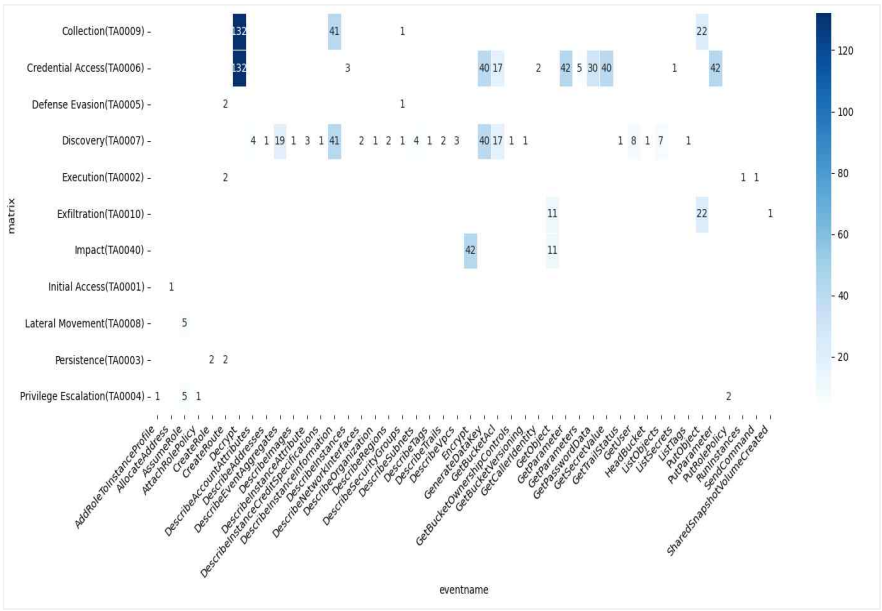

FIGURE 15. Automated Analysis Framework GUI

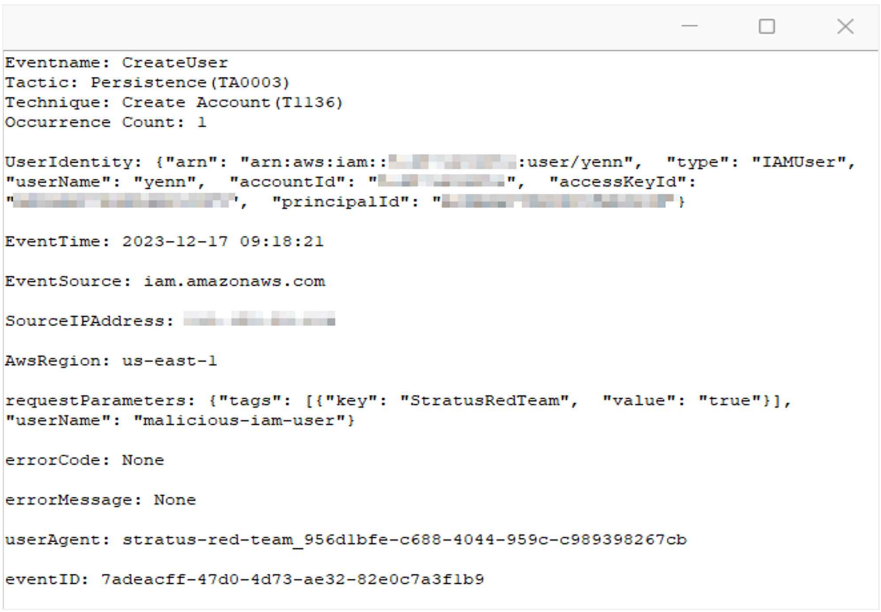

FIGURE 16. Cell Selection Screen

Ⅴ. Performance Evaluation

Analyzing AWS account API call information recorded by CloudTrail based solely on eventName is not intuitive for determining attack tactics or related resources. Therefore, we have implemented a log analysis framework that utilizes an event classification system to assist incident investigations. This section aims to demonstrate its value by analyzing logs generated through penetration testing and verifying the matching results with classified events.

1. ATT&CK Tactics Coverage Test

The log analysis framework is based on the ATT&CK Matrix, so we conducted tests to accurately assess its ability to identify eventNames corresponding to each Tactic. To do this, we utilized open-source test libraries mapped to the ATT&CK Framework and performed evaluations[24,25,33]. The critical evaluation criterion is how accurately the framework detects and classifies eventNames corresponding to each Tactic. This validates the framework's effectiveness in identifying and assessing significant events. For the experimental evaluation, we conducted coverage tests for the Discovery(TA0007) and Persistence(TA0003) tactics, representing most of the event analysis results. We also tested for the critical Credential Access(TA0006) tactic during the attack execution.

1) Analysis of Discovery(TA0007)

Table 5 describes the attack activities corresponding to the penetration process's Discovery(TA0007). This tactic primarily involves initial information gathering and reconnaissance by the attacker during penetration. The penetration test explores information for the attack on a compromised EC2.

Figure 17 visually represents the activity logs of the actions performed by the attacker to explore the environment and collect information, analyzed through the analysis framework. It can be seen that confirmed Discovery-related events perform steps to navigate the environment and collect information, such as 'GetCallerIdentity' and 'GetAccountSummary' to check the caller's account ID and IAM configuration by searching for information about users or roles, 'ListBuckets' to get a list of AWS S3 buckets, 'ListRoles' and 'ListUsers' to check all IAM roles and users in the account, and 'GetAccountAuthorizationDetails' to check detailed authority information about IAM entities.

TABLE 5. Discovery Techniques

Attack techniques	Description	Detection
ec2-enumerate-from-instance	Run the scan command after damaging the EC2	Detect instances of abnormal enumeration calls (e.g., ListBuckets)

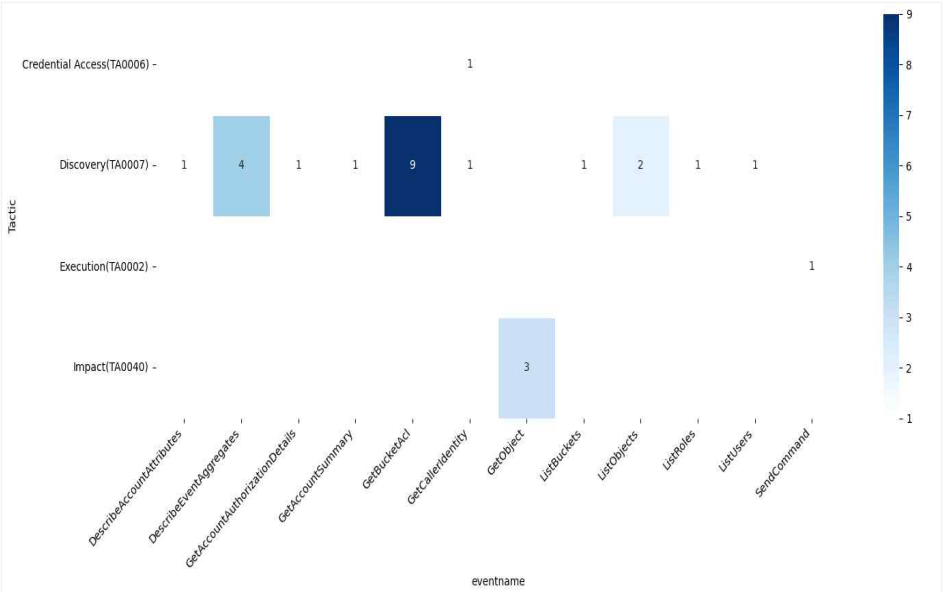

FIGURE 17. Results of Discovery Heatmap

2) Analysis of Persistence(TA0003)

Table 6 represents the attack techniques during the Persistence(TA0003). This tactic involves the attacker taking actions and employing techniques to maintain access to the system, ensuring persistent access rights.

Figure 18 is a visual analysis of the events performed by the attacker to maintain continuous access through the proposed framework. The analysis clearly shows the Persistence-related events performed by malicious users during the event of adding backdoors to IAM roles (‘UpdateAssumeRolePolicy’), adding access keys to existing IAM users (‘CreateAccessKey’), creating new IAM users with root privileges (‘CreateUser’, ‘AttachUserPolicy’, ‘createAccessKey’), generating profiles for existing IAM users (‘createLoginProfile’, ‘UpdateLoginProfile’), and overwriting Lambda function code (‘UpdateFunctionCode~*’) to establish account validity and persistence.

TABLE 6. Persistence Techniques

Attack techniques	Description	Detection
iam-back door-role	Add a backdoor to an IAM role for persistent access	Detect 'UpdateAssumeRolePolicy' event
iam-back door-user	Add an access key to an existing IAM user for persistent access	Detect 'CreateAccessKey' event
iam-create-admin-user	Create a new IAM user(root) for persistent access	Detect 'CreateUser', 'AttachUserPolicy' and 'CreateAccessKey' events
iam-create-user-login-profile	Create a profile on an existing IAM user for persistent access	Detect 'CreateLoginProfile' or 'UpdateLoginProfile' event
lambda-overwrite -code	Overwrite a Lambda function's code for persistent access	Detect 'UpdateFunctionCode~' event

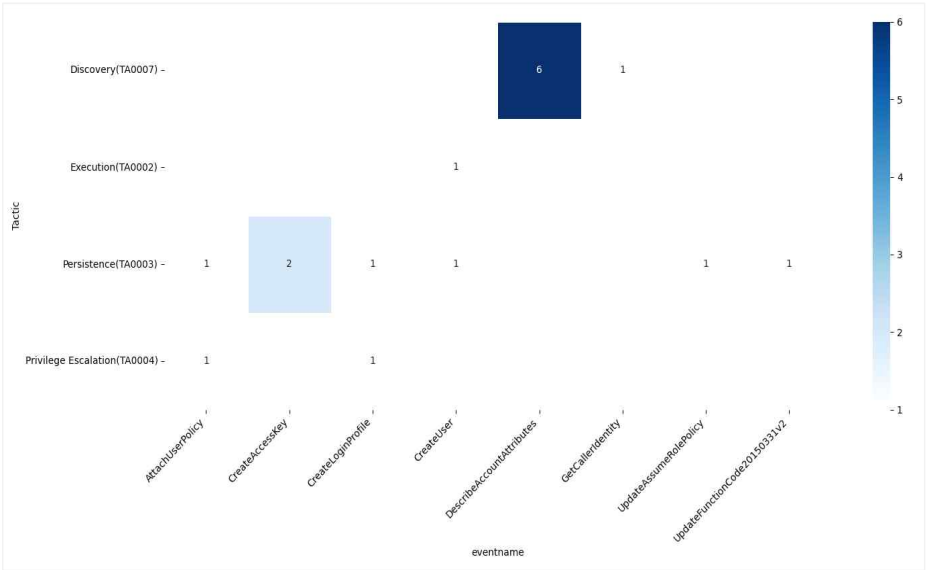

FIGURE 18. Results of Persistence Heatmap

3) Analysis of Credential Access(TA0006)

Table 7 describes the attack technology of Credential Access(TA0006), the attack technology at each stage, and the main eventName for detection that can be identified. Credential Access is a stage where an attacker attempts to obtain valid credentials within the system, aiming to gain authorization to log in to the system through various techniques and methods, such as trying to steal RDP(Remote Desktop Protocol) passwords within EC2 instances or credential searching.

Figure 19 depicts the results of analyzing actual attack logs conducted using various techniques with the analysis framework. It can visually confirm the handling of critical events during the Credential Access stage, such as searching for encrypted administrator passwords using 'GetPasswordData', retrieving detailed information about IAM users or roles with 'GetCallerIdentity', and outputting information about instances with 'DescribeInstances'. This provides a visual confirmation of the significant events during the Credential Access stage.

If attackers gain permissions in the Cloud environment (e.g., by compr omising service to obtain credentials with elevated privileges), they may request sensitive information from administrators[34]. To detect such typ es of attacks (as shown in Table 7, 'secretsmanager-retrieve-secrets'), it is essential to identify entities that retrieve many secrets by triggering t he CloudTrail 'GetSecretValue' event. In this case, additional log analysis can be performed, as shown in Figure 20, by selecting the 'GetSecretVal ue' event cell when utilizing the analysis framework. This allows you to examine event frequency, timing, MITRE ATT&CK Technique, and user

identification information related to the specific event. Consequently, it be comes possible to intuitively choose events for analysis from a large vol ume of log files and analyze the associated logs.

TABLE 7. Credential Access Attack Techniques

Attack techniques	Description	Detection
ec2-get -password-data	Steal RDP credentials within a running Windows EC2 instance	Identify the entity with a high frequency of 'GetPasswordData' events
ec2-steal-instance credentials	Unauthorized acquisition of EC2 instance credentials from the Instance Metadata Service	Identify the entity that attempted to call 'GetCallerIdentity'
secretsmanager retrieve-secrets	Search for passwords stored in Secrets Manager	Identify the subject that searches for the password and detect 'ListSecrets', 'GetSecretValue' events
ssm-retrieve securestring parameters	Generate a large number of Systems Manager parameters available in AWS regions	Identify that retrieves large numbers of SSM parameters and detect 'GetParameters' events

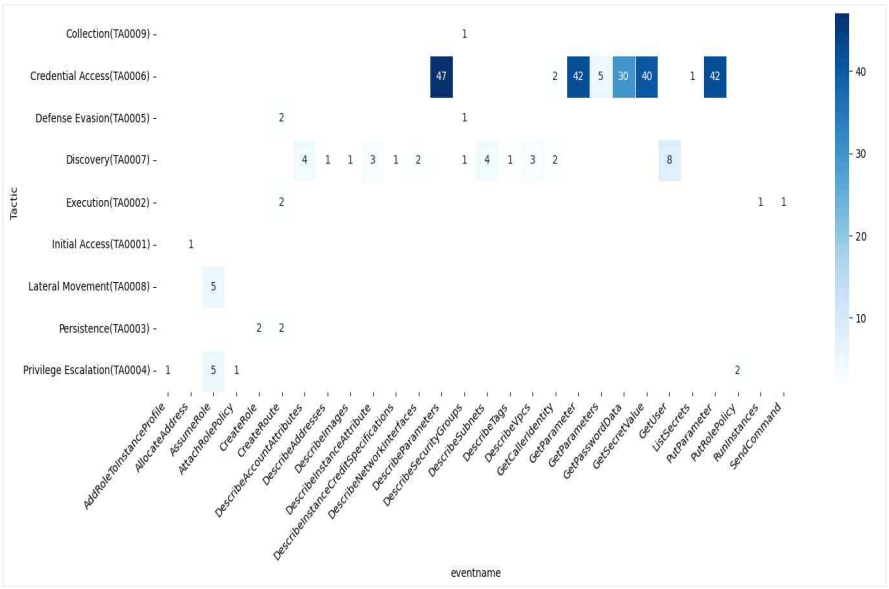

FIGURE 19. Results of Credential Access Heatmap

```
Eventname: GetSecretValue
Tactic: Credential Access(TA0006)
Technique: Cloud Secrets Management
Stores(T1555.006)
Occurrence Count: 40

UserIdentity: {"arn":
"arn:aws:iam:                    user/yenn",   "type":
"IAMUser",   "userName": "yenn",   "accountId":
                "accessKeyId":
                ,   "principalId":
                }

EventTime: 2023-11-25 08:29:40

EventSource: secretsmanager.amazonaws.com

SourceIPAddress:

AwsRegion: us-east-1

requestParameters: {"secretId":
"arn:aws:secretsmanager:us-east-1:          :secre
t:stratus-red-team-retrieve-secret-7-6oamN8",
"versionId":
"C8E5BAEF-F2F7-4267-A4CC-          "},

errorCode: None

errorMessage: None

userAgent: APN/1.0 HashiCorp/1.0 Terraform/1.1.2
(+https://www.terraform.io)
terraform-provider-aws/3.76.1
(+https://registry.terraform.io/providers/hashicorp/
aws) aws-sdk-go/1.44.157 (go1.19.3; linux; amd64)
stratus-red-team_edab5c3c-cbdc-49f1-9a4f-51306720a57
7 HashiCorp-terraform-exec/0.17.3

eventID: b82f716b-3007-4bc6-be02-675461da18cf
```

FIGURE 20. Results of Selected Event Cell

Even with large volumes of log data, the proposed visual analysis framework for events demonstrates rapid analysis and event assessment capability. Experimental evaluations confirm the enhancement of event analysis capabilities, which are expected to play a crucial role in effectively interpreting large-scale data, swiftly detecting security issues, and responding to them.

Ⅵ. Conclusion

The cloud computing environment generates a large volume of log data, making it essential to define events according to a classification system and analyze them effectively. We propose an event visualization and analysis framework that collects event logs through CloudTrail for incident investigations in AWS environments and correlates them with the ATT&CK Matrix. This makes it possible to quickly identify the necessary log events during a security incident and perform efficient log analysis. This framework provides the following advantages from an incident investigation perspective.

Firstly, in the event of an incident, it is possible to quickly determine what type of events to track among the vast amount of log data based on mapped information. Data collected from AWS CloudTrail can be systematically analyzed using the analysis framework to identify system events automatically. Visual representations assist in intuitively understanding various analysis results, enabling the rapid identification of events related to various threats in the cloud environment. This helps reduce the time and effort required for log analysis. Secondly, selecting the main eventName allows it to determine the progression and causes of incidents, enabling incident reconstruction. eventName helps track the steps or actions performed, allowing for the rapid identification of attacker pathways in the AWS environment. Furthermore, utilizing the correlation between events and the ATT&CK Matrix allows for

identifying event patterns related to specific incidents, enabling the formulation of investigative strategies based on this information. In the current situation where there is a lack of adequate research on guidelines for analyzing log events in the cloud environment, the proposed framework can help establish an effective incident investigation process. Research on analyzing behaviors using log data in cloud computing environments is continuously demanded, and it is essential to explore ways to respond to and investigate the increasing cases of cloud security threats.

In future work, we aim to implement processes to enhance the performance of the proposed framework, including memory and cost improvements. Additionally, we plan to consider expanding the framework beyond the AWS environment to include various cloud platforms like Azure and GCP. This expansion will involve establishing a standardized schema for log event normalization like OCSF.

Hereby, The goal is to contribute to enabling rapid investigations and effective responses without environmental constraints in the event of security incidents in cloud computing environments.

References

[1] “Gartner, Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach Nearly $600 Billion in 2023”, last modified April. 19, 2023, accessed on Sep. 23, 2023, https://www.gartner.com/en/newsroom/press-releases/2023-04-19-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023.

[2] James Guffey, Yanyan Li, “Cloud Service Misconfigurations: Emerging Threats, Enterprise Data Breaches and Solutions”, 2023 IEEE 13th Annual Computing and Communication Workshop and Conference(CCWC), pp. 0806-0812, Mar. 2023.

[3] Pranitha Sanda, Digambar Pawar, and V. Radha, “An insight into cloud forensic readiness by leading cloud service providers: a survey” Computing 104(9), pp. 1-26, Apr. 2022.

[4] “AWS, What is AWS CloudTrail?”, last modified Sep. 1, 2023, accessed on Sep. 23, 2023, https://docs.aws.amazon.com/ko_kr/awscloudtrail/latest/userguide/cloudtrail-user-guide.html.

[5] “Microsoft, Azure Monitor activity log”, last modified Jun. 06, 2023, accessed on Oct. 3, 2023, https://learn.microsoft.com/en-us/azure/azure-monitor/essentials/activity-log?tabs=powershell.

[6] “AWS, What is Amazon CloudWatch?”, last modified Aug. 8, 2023, accessed on Sep. 23, 2023, https://docs.aws.amazon.com/ko_kr/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html.

[7] Jayachander Surbiryala, Chunming Rong, “Cloud Computing: History and Overview” , 2019 IEEE Cloud Summit, pp. 1-7, Aug. 2019.

[8] Kang Dong-seok, “The Future of Cloud and Security”, National Information Society Agency(NIA), Future 2030, 2(16), Dec. 2020.

[9] Jon-Michael Brook, Alexander Stone Getsin, Michael Roza, “Top Threats to Cloud Computing”, Cloud Security Alliance(CSA), Jun. 2022.

[10] “AWS. What is the difference between data events and management events in CloudTrail?”, last modified 2021, accessed on Dec. 31, 2023, https://repost.aws/ko/knowledge-center/cloudtrail-data-management-events.

[11] “AWS, CloudTrail record contents”, last modified Sep. 1, 2023, accessed on Sep. 23, 2023, https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-event-reference-record-contents.html.

[12] “Gartner. Gartner forecasts worldwide public cloud revenue to grow 17.5 percent in 2019”, last modified Apr. 2, 2019, accessed on Sep. 23, 2023, https://www.gartner.com/en/newsroom/press-releases/2019-04-02-gartner-forecasts-worldwide-public-cloud-revenue-to-g.

[13] Saad Alqahtany, Nathan Clarke, Steven Furnell and Christoph Reich, “A forensic acquisition and analysis system for IaaS”, Cluster Computing, 19(1), pp. 439 - 453, Nov. 25, 2015.

[14] Suleman Khan, Abdullah Gani, Ainuddin Wahid Abdul Wahab, Mustapha Aminu Bagiwa, Muhammad Shiraz, Samee U. Khan, Rajkumar Buyya and Albert Y. Zomaya, “Cloud Log Forensics: Foundations, State of the Art, and Future Directions“, ACM Computing Surveys(CSUR), 49(1), pp 1 - 42, May. 2016.

[15] Kenny Awuson-David, Tawfik Al-Hadhrami, Mamoun Alazab, Nazaraf Shah and Andrii Shalaginov, “BCFL logging: An approach to acquire and preserve a

dmissible digital forensics evidence in cloud ecosystem", Future Generation Co mputer Systems, 122, pp.1 - 13, Sep. 2021.

[16] Benjamin Yankson and Adam Davis, "Analysis of the Current State of Cloud Forensics: The Evolving Nature of Digital Forensics", 2019 IEEE/ACS 16th Int ernational Conference on Computer Systems and Applications (AICCSA), pp. 1 -8, Nov. 2019.

[17] "MITRE ATT&CK : MITRE ATT&CK®", accessed on Dec. 21, 2023, https:/ /attack.mitre.org.

[18] "Cloud Matrix", https://attack.mitre.org/matrices/enterprise/cloud/, accessed on Dec. 21, 2023.

[19] Chanho Shin and Changhee Choi, "Cyberattack Goal Classification Based on MITRE ATT&CK: CIA Labeling", Journal of Internet Computing and Services (JICS), 23(6), pp. 15-26, Dec. 2022.

[20] Simeen Sheikh, Ganesan Suganya and Premalatha Mariappan, "Automated Re source Management on AWS Cloud Platform", Proceedings of 6th International Conference on Big Data and Cloud Computing Challenges : ICBCC 2019, 164, pp. 133-147. Oct. 2019.

[21] P Rajesh, Mansoor Alam, Mansour Tahernezhadi, A Monika and Gm Chanak ya, "Analysis Of Cyber Threat Detection And Emulation Using MITRE Attack Framework", 2022 International Conference on Intelligent Data Science Technol ogies and Applications(IDSTA), pp. 4 - 12, Sep. 2022.

[22] "AWS. Building a SIEM with AWS services", accessed on Sep. 23, 2023, http s://catalog.us-east-1.prod.workshops.aws/workshops/2ff04db5-bb02-4208-b637-d 54a352f7bc6/ko-KR/10-siem.

[23] “AWS. Amazon Kinesis Data Firehose”, accessed on Sep. 23, 2023, https://aws.amazon.com/ko/kinesis/data-firehose/.

[24] “Stratus Red Team”, accessed on Dec. 23, 2023, https://stratus-red-team.cloud/.

[25] “DataDog, stratus-red-team”, accessed on Dec. 23, 2023, https://github.com/DataDog/stratus-red-team/tree/main.

[26] “SigmaHQ. Sigma”, accessed on Apr. 2023, https://github.com/SigmaHQ/sigma.

[27] Chen Qian, Yan Wang and Lei Guo, “A novel method based on data visual autoencoding for time-series classification”, Proceedings of the 2015 Chinese Intelligent Automation Conference, Lecture Notes in Electrical Engineering, 336. Springer, Mar. 2015.

[28] Prasad Purnaye and Vrushali Kulkarni, “BiSHM: Evidence detection and preservation model for cloud forensics”, Open Computer Science, 12(1), pp. 154-170, May. 2022.

[29] Chan-Woong Hwang, Sung-Ho Bae and Tae-Jin Lee, “MITRE ATT&CK and Anomaly detection based abnormal attack detection technology research”, Journal of Information and Security, 21(3), 13-23, Sep. 2021.

[30] Shilin He, Qingwei Lin, Jian-Guang Lou, Hongyu Zhang, Michael R. Lyu and Dongmei Zhang, “Identifying impactful service system problems via log analysis” Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 60 - 70, Oct. 2018.

[31] Paul Agbabian, “Understanding the Open Cybersecurity Schema Framework”,

Open Cybersecurity Schema Framework(OCSF), Aug. 2023.

[32] Alan D Moore, “Python GUI Programming with Tkinter: Design and build functional and user- friendly GUI applications”, Packt Publishing Ltd, Mar. 2022.

[33] Alexander M. Kirksharian, “Solving the Skills Gap: A Dynamic Approach to Cybersecurity Training”, Master of Science in Computer Science : San Diego State University, last modified May. 9, 2022, accessed on Dec. 28, 2023.

[34] MITRE ATT&CK, “Credentials from Password Stores: Cloud Secrets Manage ment Stores”, https://attack.mitre.org/techniques/T1555/006/, accessed on Dec. 3 1, 2023.

논문 개요

클라우드 환경에서의 ATT&CK Matrix 기반 이벤트 로그 분석 프레임워크

김예은
미래융합기술공학과
성신여자대학교 대학원

클라우드 마이그레이션에 대한 수요가 높아짐에 따라, 클라우드 컴퓨팅 환경에서의 보안 위협도 빠르게 증가하고 있다. 이로 인해 로그 데이터의 규모와 복잡성이 커지고 있어 침해사고 발생 시 어떤 이벤트를 분석해야 하는지 판단하는 것이 어렵다.

본 논문에서는 AWS 클라우드 환경에서 발생하는 이벤트 로그를 효율적으로 조사하기 위한 분석 프레임워크를 개발했다. 이를 위해 클라우드 보안 위협을 식별하는 데 필요한 데이터를 보다 세부적으로 분류하였으며, 로그 분석 과정의 효율성을 개선했다. 더불어, MITRE ATT&CK의 Cloud Matrix를 기반으로 한 로그 이벤트 분류 체계를 활용하여, 사용자 행위 로그와 관련된 공격 이벤트 정보를 직관적으로 이해할 수 있도록 했다.

현재 클라우드 환경에서 로그 이벤트 분석을 위한 명확한 가이드라인이 부족한 상황에서, 본 연구는 효과적인 사고 조사 프로세스를 구축하는 데 중요한 역할을 한다. 제안 프레임워크를 활용함으로써 클라우드 컴퓨팅 환경에서 발생하는 사고에 대해 신속하고 효과적으로 대응할 수 있으며, 이는 전반적인 클라우드 보안 위협의 사고 분석 능력을 향상시킬 것으로 기대한다.

Acknowledgements

본 논문을 지도해 주신 김성민 교수님과 박원형 교수님, 공저자로 함께 기여해 준 김정아, 홍지원, 채시윤 학생께 감사드립니다.