Cloud-Native Security Information and Event Management System on Aws

Keerthan[1], Uday Karthik[2], Sonu Singh[3], Kshitiz[4]
[1,2,3,4]*Keerthan, Parul Institute of Engineering and Technology*

***Abstract*—The rise of cloud-first architectures has transformed enterprise computing but simultaneously expanded the attack surface for security threats. Traditional Security Information and Event Management (SIEM) solutions, while effective for on-premises systems, struggle with scalability, cost, and integration challenges in dynamic cloud environments. This research presents a Cloud-Native SIEM architecture designed and implemented using Amazon Web Services (AWS). The system leverages AWS Lambda, DynamoDB, S3, EventBridge, CloudTrail, and API Gateway to deliver real-time threat detection, alerting, and visualization within a fully serverless ecosystem. The architecture enables centralized log ingestion, normalization, and analysis from synthetic and real-world AWS CloudTrail events. Implemented under AWS Free Tier constraints, the solution demonstrates how serverless paradigms can achieve cost-efficient, scalable, and resilient SIEM operations. Testing indicates ingestion throughput exceeding 1000 events/minute with latency below 200 ms, proving its viability for academic, small business, and low-cost enterprise deployments. Future improvements include integration with Amazon GuardDuty, machine learning-based anomaly detection, and multi-source event correlation.**

***Index Terms*—SIEM, Cloud Security, AWS, Serverless, EventBridge**

I. INTRODUCTION

Cloud computing has rapidly become the foundational backbone of modern enterprise IT infrastructure. The inherent flexibility, elasticity, and global reach offered by platforms such as Amazon Web Services (AWS) have significantly accelerated the migration of mission-critical workloads from traditional on-premises data centers to the cloud. However, this transformative shift has concurrently introduced new and evolving vectors of security risk, encompassing issues such as misconfigurations, insider threats, API abuse, and unauthorized access [1]. In response to these challenges, Security Information and Event Management (SIEM) systems are crucial for aggregating and analyzing logs to identify suspicious activities and policy violations. While established SIEM platforms like Splunk or IBM QRadar provide robust capabilities for on-premises environments, they often incur substantial costs and demand significant maintenance overhead [2]. Cloud-native environments, in stark contrast, necessitate lightweight, serverless, and real-time monitoring solutions that can dynamically adapt to their inherently event-driven infrastructures [3]. This paper introduces a novel Cloud-Native SIEM System on AWS, a serverless solution specifically engineered to provide centralized logging, alerting, and visualization capabilities tailored for cloud environments. The system efficiently ingests logs via AWS EventBridge, processes them using AWS Lambda, and stores normalized records in Amazon DynamoDB and S3 for comprehensive analysis. Critical alerts are propagated through Amazon Simple Notification Service (SNS), while a React-based dashboard hosted on AWS Amplify offers real-time metric visualization.

1.1 The contributions of this research are threefold

A. Design and Development

A comprehensive, fully serverless SIEM architecture leveraging a suite of AWS managed services [4].

B. Real-time Analytics Implementation

The successful implementation of real-time log analytics and alert generation mechanisms utilizing AWS EventBridge and Lambda.

C. Performance Evaluation
An empirical evaluation of the system's performance, scalability, and usability within a cost-constrained (AWS Free Tier) environment.

The remainder of this paper is structured as follows: Section II provides a review of related literature and background information. Section III details the problem statement and research gap. Section IV outlines the threat model and risk assessment methodology. Section V describes the system architecture and design. Sections VI and VII present the methodology and implementation details, respectively. Results and discussion are covered in Sections VIII through XII. Finally, Section XIII concludes the paper and Section XIV discusses future work directions.

II. LITERATURE REVIEW AND BACKGROUND

A. Evolutions of SIEM Systems
The concept of Security Information and Event Management (SIEM) originated from the convergence of Security Information Management (SIM) and Security Event Management (SEM). Early SIEM architectures primarily relied on centralized servers for log collection and correlation, which often led to significant scalability and latency limitations. As the field evolved, researchers like Roschke [5] proposed modular, virtualized SIEM architectures aimed at improving extensibility and interoperability. More recently, research has shifted towards cloud-native SIEMs, which leverage distributed and event-driven models to inherently enhance scalability and adaptability [6].

B. Cloud Security and Event-Driven Design
Cloud-native SIEMs inherently benefit from event-driven architectures, where security triggers such as API calls or user activities automatically invoke processing functions. For instance, Vignesh and Agarwal [3] demonstrated that event-driven designs utilizing message queues significantly enhance real-time responsiveness in distributed systems. Similarly, Moustafa [6] introduced a lightweight SIEM specifically optimized for zero-day attack detection within cloud contexts. These foundational works collectively reinforce the feasibility and advantages of implementing SIEM functionality through serverless components like AWS Lambda and EventBridge, which form the core of this study's architecture.

C. Machine Learning and Anomaly Detection
Recent advancements in cybersecurity research increasingly emphasize the integration of machine learning (ML) to improve the accuracy and efficiency of anomaly detection. Zhao et al. [7] showcased the effectiveness of LSTM-based log analysis in achieving higher detection precision compared to traditional rule-based systems. Furthermore, reviews by Hagemann and Katsarou [8] highlight that hybrid ML-rule approaches often outperform standalone ML models in cloud anomaly detection. While our current prototype employs a simpler z-score analysis for anomaly detection, its architecture is deliberately designed for seamless future integration with more advanced ML capabilities.

D. Cloud-Native Security Practices
Official AWS documentation and whitepapers consistently recommend the use of services such as CloudTrail, CloudWatch, and EventBridge for robust monitoring of cloud environments [9]. These services are instrumental in providing continuous audit trails and enabling automated responses to identified anomalies. Moreover, Amazon GuardDuty, which integrates ML-driven detection, presents a clear pathway for incorporating advanced analytics into subsequent versions of this SIEM system [10].

E. Gaps in Current Research
Despite significant progress in both commercial and open-source SIEM solutions (e.g., Wazuh, ELK Stack), many existing systems remain resource-intensive and complex to manage, particularly for smaller organizations or academic settings [11]. This study directly addresses this critical gap by proposing and implementing a lightweight, cost-efficient, and fully serverless SIEM. Our solution is specifically tailored for educational institutions, startups, and small-to-medium enterprise (SME) use cases, providing an accessible and scalable security monitoring platform.

III. PROBLEM STATEMENT AND RESEARCH GAP

Modern enterprises generate millions of security events daily across cloud, hybrid, and on-premises environments. Traditional Security Information and Event Management (SIEM) systems, while capable of log correlation and threat detection, often struggle in serverless and dynamic cloud environments. These struggles are primarily due to high data ingestion costs, rigid scaling limitations, and a general lack of contextual awareness for cloud-specific telemetry [12]. Cloud-native systems, such as those offered by AWS, Azure, and Google Cloud Platform (GCP), introduce new forms of telemetry (e.g., API calls, service-level logs, Identity and Access Management (IAM) policy changes). Existing SIEM solutions frequently cannot process this data efficiently without complex configurations or prohibitively expensive licenses. The core problem addressed by this research is, How can a cost-efficient, fully serverless SIEM be designed to provide real-time detection, high scalability, and broad accessibility for academic, startup, and small-to-medium enterprise (SME) use within cloud-native architectures? This study aims to bridge the gap between theoretical SIEM frameworks and practical, deployable cloud-native systems. It emphasizes modularity, automation, and affordability as key design principles. Furthermore, this work contributes to the academic exploration of serverless cybersecurity engineering, a field that remains in its early adoption phases within research communities [13].

IV. THREAT MODEL AND RISK ASSESSMENT

The proposed SIEM is designed to detect, visualize, and mitigate various threat categories commonly observed in AWS environments. These threats and their corresponding mitigation mechanisms are detailed below.

A. Threats and its Mitigations

1. Unauthorized Access: This category includes threats such as compromised credentials or brute-force login attempts. An example log pattern for this would be a "ConsoleLogin" event with a failed status. The system mitigates this by triggering a Lambda alert and sending an SNS notification.
2. Insider Threats: These involve malicious IAM users altering privileges. Log patterns like "PutUserPolicy" or "AttachRolePolicy" indicate such threats. Mitigation involves generating a high-severity alert and correlating it with the source IP address.
3. Data Exfiltration: This threat occurs when S3 bucket access originates from unusual or foreign IP addresses. A "GetObject" event with an unusual geolocation is an example log pattern. An EventBridge Rule triggers an alert to address this.
4. Configuration Drift: This refers to unauthorized or unexpected changes in critical AWS service settings, such as CloudTrail or VPC configurations. Log patterns like "StopLogging" or "DeleteTrail" are indicative. The system responds with an auto-alert and suggests lockdown actions.
5. API Abuse: Characterized by repeated unusual API calls, such as a high frequency of "Invoke" events. Mitigation involves rate-limiting and anomaly tagging to identify and control such activities.

This threat model adheres to the STRIDE methodology (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege). Each identified event type is correlated with one or more STRIDE categories, thereby enhancing the system's ability to accurately assess risk severity.

B. Risk Quantification

Each log event is assigned a risk score, ranging from 0 to 10, derived from a set of weighted parameters.

1. Event Frequency (EF): The number of occurrences per hour.
2. Impact Potential (IP): The potential for data exposure or privilege escalation.
3. Confidence (C): Derived from the certainty of anomaly detection.

The risk score is calculated using the following formula

$$\text{Risk Score} = (0.4 * EF) + (0.4 * IP) + (0.2 * C)$$

Events exceeding a predefined threshold of 7.0 automatically trigger Critical Alerts. These alerts are

then routed to SOC dashboards and distributed via email notifications. This semi-quantitative scoring approach is aligned with the risk management guidelines outlined in NIST 800-30.

V. SYSTEM ARCHITECTURE AND DESIGN

The proposed system adopts a four-layer architecture encompassing log sources, ingestion, storage, and visualization. Each layer leverages AWS-managed services to ensure scalability, fault tolerance, and minimal operational overhead.

A. Log Sources

Logs are collected from two primary categories:

1. Synthetic logs generated for testing purposes (e.g., simulated login attempts, access policy changes).
2. AWS CloudTrail events, which represent real user activities and API interactions occurring within an AWS account.

B. Ingestion Layer

AWS EventBridge functions as the central event bus, capturing CloudTrail logs in real time and routing them to an Ingest Lambda function. This Lambda function is responsible for normalizing the logs into a common schema (including timestamp, event type, user, source IP, severity, and message) and performing enrichment tasks such as severity tagging and alert classification.

C. Storage Layer

Structured data is stored in Amazon DynamoDB, which is optimized for low-latency queries. Raw logs are archived in Amazon S3, partitioned by date for compliance and long-term retention [14]. The use of secondary indexes in DynamoDB allows for efficient filtering of data by severity, event type, or user ID.

D. Visualization Layer

The frontend dashboard, developed using React.js with Tailwind CSS and hosted on AWS Amplify, provides a visual interface for alerts, Key Performance Indicators (KPIs), and detected anomalies. APIs exposed through AWS API Gateway enable secure communication with the backend Lambda functions. Additionally, high-severity alerts are propagated via Amazon Simple Notification Service (SNS) for immediate email or SMS notifications.

E. Security Controls

The system's design strictly adheres to least privilege and secure-by-design principles:

1. IAM roles are implemented to enforce strict access segregation between Lambda functions, DynamoDB, and S3 buckets.
2. All traffic between the frontend and backend components is encrypted using HTTPS/TLS 1.3 to ensure data confidentiality and integrity.
3. Logs are sanitized to prevent common vulnerabilities such as injection attacks, aligning with OWASP standards.

VI. METHODOLOGY

A. Development Model

The Agile Scrum methodology guided the development process, emphasizing iterative sprints for modular implementation. Each sprint specifically targeted a distinct subsystem, including the ingestion pipeline, storage layer, API layer, and frontend visualization. Continuous testing was performed after each sprint to ensure reliability.

B. Security-by-Design Workflow

Security was integrated throughout the entire development lifecycle, following a security-by-design workflow:

1. Planning: Defining clear security and compliance objectives.
2. Implementation: Applying IAM policies and encryption for all data flows.
3. Verification: Conducting continuous monitoring using AWS CloudWatch metrics.
4. Testing: Performing synthetic attack simulations to validate alert accuracy.

C. Tools and Technologies

The key components and technologies utilized for the development of the Cloud-Native SIEM system are summarized in Table 1.

Table 1. Technologies used for Development

Component	Technology
Cloud Provider	Amazon Web Services (AWS)
Backend	AWS Lambda (Python 3.12, boto3)
Storage	DynamoDB, S3
Event Processing	EventBridge
Frontend	React.js, Tailwind CSS, AWS Amplify
Notifications	SNS
APIs	API Gateway, log generator
Testing	

D. Workflow Overview

The system operates based on an event-driven workflow, ensuring near-real-time monitoring and elasticity under varying load conditions:

1. A log event is triggered (either from CloudTrail or a synthetic generator).
2. EventBridge captures this event and routes it to the Ingest Lambda function.
3. The Ingest Lambda normalizes the event data and stores it in DynamoDB and S3.
4. A Query Lambda function provides APIs for dashboard functionality and Key Performance Indicator (KPI) computation.
5. Alerts are sent via SNS for critical events.

VII. IMPLEMENTATION AND TESTING

A. Backend Implementation

The system's backend is constituted by two primary Lambda functions:

1. Ingest Lambda: This function is responsible for parsing incoming logs, classifying their severity, writing the processed data to DynamoDB and S3, and triggering alerts via SNS when necessary.
2. Query Lambda: This function fetches filtered data, calculates Key Performance Indicators (KPIs), and detects anomalies using z-score-based statistical thresholds. These functions were individually tested through unit tests utilizing synthetic event payloads.

B. Frontend Development

The dashboard, serving as the frontend, presents information through three main panels:

1. KPI Metrics: Displays aggregated metrics such as total, critical, and resolved alerts.
2. Alerts Table: Provides a paginated table of alerts, which can be filtered by type, severity, or source.
3. Charts and Anomaly Graphs: Visualizes hourly alert distributions using Chart.js to highlight trends and anomalies.

C. Testing Phases

The system underwent a comprehensive testing process, including the following phases:

1. Unit Testing: Verified the functionality of individual Lambda functions, specifically log ingestion and normalization, using mock payloads.
2. Integration Testing: Ensured the seamless end-to-end flow of data, from CloudTrail log capture to dashboard visualization.
3. Performance Testing: A synthetic log generator was used to produce up to 1000 events per minute, with the system consistently sustaining latency below 200 m/s.
4. User Acceptance Testing (UAT): SOC analysts evaluated the dashboard's usability and the clarity of alert interpretation to ensure it met operational requirements.

D. Results

The testing results confirmed that the system performs efficiently even under moderate load conditions, thereby validating the scalability and effectiveness of its serverless design.

Table 2. Performance Metrics

Metric	Expected	Observed
Ingestion latency	< 200 m/s	150 m/s
API Response	< 300 m/s	180 m/s
Dashboard Update Relay	< 5 seconds	2-3 seconds
Peak Throughput	1000 events/min	1000 +

VIII. RESULTS AND DISCUSSION

The project successfully validated that serverless architectures can effectively implement Security Information and Event Management (SIEM) functionalities. When compared with traditional SIEM solutions, the proposed system offers several significant advantages:

A. Cost Reduction

The system operates entirely within the AWS Free Tier, leading to a reduction in infrastructure expenses by over 90% compared to commercial tools.

B. Elastic Scalability
It auto-scales seamlessly with AWS Lambda and DynamoDB, thereby eliminating the need for manual resource provisioning.

C. Low Latency
A sub-200ms event processing time enables near real-time alerting capabilities.

D. Ease of Deployment
The use of infrastructure-as-code deployment via AWS CLI significantly simplifies the setup process.

E. Usability
The React-based dashboard provides intuitive interfaces, enhancing the experience for Security Operations Center (SOC) analysts.

However, the system currently has certain limitations. These include the absence of deep machine learning-based anomaly detection and a limited diversity of log sources (presently supporting only CloudTrail). Future enhancements, such as integration with VPC Flow Logs and Amazon GuardDuty, are expected to significantly enhance its detection breadth [15].

IX. EXPERIMENTAL EVALUATION AND PERFORMANCE TESTING

Beyond basic functional testing, a controlled experimental setup was conducted to thoroughly evaluate the system's performance metrics. For this evaluation, a synthetic log generator streamed 10,000 simulated CloudTrail-like events to the system over a period of 10 minutes.

A. Key Metrics Observed
The following key performance metrics were observed during the experimental evaluation

1. Average Ingestion Throughput: The system achieved an average ingestion throughput of 1,050 events per minute.
2. Mean API Response: The mean response time for API calls was measured at 180 ms.
3. Storage Cost: The daily storage cost was approximately $0.03, which remained well within the AWS Free Tier limits.
4. Memory Utilization: The average memory utilization per Lambda execution was 120 MB.
5. System Uptime: Under simulated load conditions, the system demonstrated a high uptime of 99.98%.

B. Scalability Test
A scalability test involved a linear increase in log volume, up to 10 times the baseline. The results showed proportional auto-scaling by AWS Lambda, confirming the system's ability to adapt dynamically without human intervention. This outcome validates AWS's horizontal scaling model and its "pay-per-execution" economic benefits.

C. Anomaly Detection Accuracy
To assess anomaly detection capabilities, 100 anomalous events were manually injected into the system. This resulted in a 91% detection accuracy with a 5% false positive rate. This confirms that statistical anomaly filtering, specifically using the z-score method, can achieve acceptable performance for environments with low event volumes.

X. CASE STUDY: SIMULATED SECURITY INCIDENT

To effectively demonstrate the real-world usability and capabilities of the system, a simulated insider attack scenario was executed. The incident unfolded as follows

A. Attack Vector
An IAM user attempted to disable CloudTrail logging, which was identified by a `StopLogging` event.

B. System Response
An EventBridge rule successfully captured this log event, and a Lambda function was triggered, generating an alert with a "Critical" severity level.

C. Alert Propagation
The Simple Notification Service (SNS) promptly notified the administrator via email within 2 seconds of the event.

D. Dashboard Update
The AWS Amplify dashboard immediately displayed an alert titled "CloudTrail Disabled Attempt," complete with relevant event metadata.

E. Post-Incident Analysis

The Query Lambda function correlated the user's current activity with their previous activities, revealing a clear pattern indicative of a privilege escalation attempt.

This experiment effectively highlights how a serverless SIEM can replicate core enterprise Security Operations Center (SOC) functionalities in a fully automated manner, encompassing incident capture, automated classification, notification, and visualization all achieved without the need for persistent servers.

XI. LIMITATIONS AND CONSIDERATIONS

Despite its demonstrated success, the proposed system currently presents several limitations that warrant consideration

A. Limitations

1. System Uptime: Under simulated load conditions, the system demonstrated a high uptime of 99.98%.
2. Limited Log Sources: The system currently supports only AWS CloudTrail and synthetic logs, lacking comprehensive multi-cloud ingestion capabilities.
3. Basic Anomaly Logic: It relies on Z-score statistics for anomaly detection, rather than more advanced machine learning (ML) models or behavior baselining techniques.
4. No Real Incident Response Automation: While alerts notify users, the system does not yet trigger automated containment or remediation actions as part of incident response.
5. Privacy and Data Retention: Although logs are anonymized, future versions must ensure full compliance with data privacy regulations such as GDPR and ISO/IEC 27001, particularly concerning user data [16].

B. Ethical Considerations

When handling security event logs, it is paramount for researchers to ensure that no personally identifiable information (PII) is exposed. The deployment and experimentation were strictly confined to a sandbox AWS account, thereby ensuring ethical compliance and controlled experimentation. Future iterations of the system will incorporate privacy-preserving analytics, potentially utilizing techniques such as federated learning and differential privacy.

The results of this research validate that serverless computing is not merely a cost optimization tactic but represents a transformative approach for cybersecurity analytics. By leveraging managed services like EventBridge and DynamoDB, it becomes feasible to maintain both high performance and regulatory compliance, even in resource-constrained environments. Compared to traditional enterprise SIEMs that often necessitate dedicated Security Operations Center (SOC) teams, this serverless design enables automated, intelligent monitoring for smaller organizations, effectively democratizing access to advanced security analytics technology. Furthermore, the inherent modularity of AWS services offers significant opportunities for future integration, which include AWS SageMaker for advanced machine learning-based anomaly scoring, Amazon Detective for enhanced visual forensic exploration capabilities and Step Functions for orchestrating complex incident response playbooks.

XII. CONCLUSION

The Cloud-Native SIEM on AWS demonstrates a pioneering approach to cybersecurity for the cloud era, effectively combining affordability, scalability, and automation within a single event-driven pipeline. This project underscores that robust security monitoring no longer requires expensive infrastructure; instead, innovation in serverless architecture [17] can empower even small organizations to achieve enterprise-grade visibility.

This research significantly contributes to both academic and industrial domains by providing:

1. A comprehensive blueprint for serverless SIEM development that is adaptable to multiple cloud environments.
2. A valuable cost-performance benchmark for cloud-native cybersecurity tools.
3. A foundational framework for future exploration in AI-driven Security Operations Center (SOC) automation.

XIII. FUTURE ENHANCEMENTS

Building upon the current system, several key areas have been identified for future enhancements to further expand its capabilities and robustness.

A. Integration with Multi-Cloud Telemetry Sources

Future work will focus on integrating with telemetry sources from other cloud providers, such as Azure Sentinel and GCP Security Command Center, to enable multi-cloud monitoring.

B. Implementation of Unsupervised Deep Learning Models

To improve anomaly detection, the system will incorporate unsupervised deep learning models for more sophisticated log anomaly prediction.

C. Development of Self-Healing Response Systems

This involves creating self-healing response mechanisms through advanced event orchestration and predictive analytics to automate incident resolution.

D. Blockchain-Backed Log Integrity Verification

To ensure the immutability and non-repudiation of audit data, future iterations will explore blockchain-backed log integrity verification [18].

This research significantly contributes to the growing domain of cloud-native cybersecurity by providing a reproducible blueprint for scalable SIEM implementations, particularly under minimal cost constraints.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Computer Science & Engineering, Parul University, for their continuous guidance and unwavering support throughout this research endeavor. Special thanks are also extended to our esteemed project mentor, E. Kanniapan, and the faculty coordinators for their invaluable feedback, insightful suggestions, and consistent encouragement during the entire development and evaluation phases of the Cloud-Native SIEM System on AWS.

REFERENCES

[1] Aggrey, R., and Cudjoe, A., “Cloud Security Best Practices: Strategic Measures to Protect Digital Assets Within the Cloud,” International Journal of Future Management and Research (IJFMR), 2025. [Online]. Available: https://www.ijfmr.com/researchpaper.php?id=35156

[2] Bezas, K., and Filippidou, F., “Comparative Analysis of Open Source SIEMs,” Indonesian Journal of Computer Science, 2023. [Online]. Available: https://doi.org/10.33022/ijcs.v12i2.3182

[3] Natarajan, V., and Agarwal, R., “Event-Driven Architecture for Cloud Storage Systems,” Journal of Emerging Technologies and Innovative Research (JETIR), 2018. [Online]. Available: https://www.jetir.org/papers/JETIR2501189.pdf

[4] O’Meara, W., and Lennon, R., “Serverless Computing Security: Protecting Application Logic,” in Proc. International Symposium on Software Reliability Engineering, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9180214

[5] Roschke, S., Cheng, F., and Meinel, C., “An Extensible and Virtualized SIEM System Architecture,” in Proc. IEEE International Conference on Information Assurance and Security, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/5283248

[6] Moustafa, N., “A New Threat Intelligence Scheme for Safeguarding Cloud Environments from Zero-Day Cyber Attacks,” Future Generation Computer Systems, Elsevier, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8374422

[7] Zhao, Z., Xu, C., and Li, B., “A LSTM-Based Anomaly Detection Model for Log Analysis,” Journal of Network and Systems Management, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11265-021-01644-4

[8] Hagemann, T., and Katsarou, K., “A Systematic Review on Anomaly Detection for Cloud Computing,” in Proc. AICCC Conference, 2020. [Online]. Available:

https://dl.acm.org/doi/10.1145/3442536.3442550

[9] Amazon Web Services, "Security Monitoring and Logging in AWS," Whitepaper, 2023. [Online]. Available: https://aws.amazon.com/blogs/security/logging-strategies-for-security-incident-response/

[10] Amazon Web Services, "Amazon GuardDuty Threat Detection and Continuous Monitoring," Documentation, 2022. [Online]. Available: https://docs.aws.amazon.com/guardduty/

[11] Amami, R., Charfeddine, M., and Masmoudi, S., "Exploration of Open Source SIEM Tools and Deployment of an Appropriate Wazuh-Based Solution for Strengthening Cyber Defense," in Proc. International Conference on Digital Information and Communication Technology and its Applications (CODIT), 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10708476

[12] Naseer, A., Naseer, H., and Ahmad, A., "Real-time Analytics, Incident Response Process Agility and Enterprise Cybersecurity Performance: A Contingent Resource-Based Analysis," International Journal of Information Management, 2021. [Online]. Available: https://doi.org/10.1016/j.ijinfomgt.2021.102334

[13] Alshamrani, A., and Huang, D., "A Survey on Advanced Persistent Threats," IEEE Communications Surveys & Tutorials, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8606252

[14] Medeiros, A. S. I., and Ferreira, B., "A Cost-Effective Cloud Event Archival for SIEMs," in Proc. IEEE International Conference on Cloud Computing (CLOUD), 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8939948

[15] Amazon Web Services, "Analyzing VPC Flow Logs," AWS Security Blog, 2020. [Online]. Available: https://aws.amazon.com/blogs/security/investigate-vpc-flow-with-amazon-detective/

[16] Ponemon Institute, "The State of SOC in the Cloud Era," Report, 2020. [Online]. Available: https://www.devo.com/resources/analystresearch/devo-soc-performance-report/

[17] Flower, M., "Serverless Architectures," MartinFowler.com, 2018. [Online]. Available: https://martinfowler.com/articles/serverless.html

[18] Pahl, C., and Lee, B., "Containers and Cloud: From LXC to Docker to Kubernetes," IEEE Cloud Computing, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7158965