

Intelligent Cloud Based Log Analyzer for Security Monitoring Using AWS

B. Harish Goud¹, Ambeer Shravan Kumar², Bandari Lakshmana Prasad³

¹Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

²Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

³Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

Abstract — Log data is a critical component of understanding system behavior and identifying security threats in modern cloud computing environments [10]. However, log analysis systems are often expensive, complicated, and difficult to scale. This paper proposes the design of an intelligent log analysis system using Amazon Web Services. The proposed log analysis system is designed using a unified data model that combines rule-based and machine learning-based techniques for identifying suspicious patterns. The proposed system is designed to scale using a serverless architecture. The design of the log analysis system using a serverless architecture is considered to be beneficial for simplifying log analysis.

Key Words: Cloud Computing, Log Analysis, Cybersecurity, AWS, SIEM, Anomaly Detection

1. INTRODUCTION

In today's cloud computing and distributed applications, which are rapidly evolving, systems are producing enormous amounts of log data in real-time. Log data is highly valuable, as it can provide valuable information on the behavior of the system, how users interact with it, and potential security threats [10]. Effective analysis of log data is critical in keeping the system in good health and fending off potential cyber threats.

Traditional log monitoring tools were originally designed for centralized systems and do not map well onto today's cloud-native environments [2]. These tools often require heavy hardware, complex setup, and their rule-based approach makes it difficult to identify unknown threats.

In this work, we propose a intelligent cloud-based log analyzer tool that can address the challenges.

2. RELATED WORK

Log analysis has improved greatly with with the emergence of cloud computing and machine learning. Nowadays, SIEM systems gather logs from various and detect anomalies indicative of potential risks [1]. Studies on log systems point out the need that categorizing logs into distinct classes is important. The When, What, Who and Where model can be used to organize log data [2].

This makes it easier and clearer to analyze efficient. Machine learning methods, particularly anomaly algorithms such as Isolation Forest, have been highly effectiveness in detecting unusual behavior [6,7]. In comparison to server-based architectures, processing events via cloud services is a cost-efficient and scalable way of establishing order [4]. It offers benefits in terms of cost and scalability.

3. LIMITATIONS OF EXISTING SYSTEMS

With all technological improvements in log monitoring, there challenges still remain.

The logs are dispersed across different systems and are in different formats. This leads to atomized data which is hard to get a holistic picture.

Rule-based systems are not very effective in identifying new patterns in attacks. These systems can be circumvented by adhering to certain limits [9].

The infrastructure is complex. In traditional systems, re ource provisioning is an issue because it is costly.

Furthermore, they may not be user-friendly. non-technical users.

4. PROPOSED SYSTEM

The system offers a simplest and scalable system to analyze logs, supporting massive data of data. Our system discovers risks that may damage our systems.

The system takes logs from the applications and safely stores them in the cloud on S3 AWS. This configuration enables the system to run without continual intervention while ensuring data availability. The system is also able to process logs as soon as logs as soon as they are uploaded. [4].

The analysis of real-time log data can be done using AWS Lambda Functions. Logs will be manipulated by extracting helpful information from them and then converting them to a desired format. Due to not having to have a server running, costs will be reduced.

The system splits log files into four major components to ease the analysis: the date of the event (when the event occurred), the type of action (what was done), the user (who did something), and the location to perform the event (where the event was done) [2].

This format allows us to easily identify suspicious events and to find patterns of threats.

The proposed system is very effective in detecting threats. It does this in two ways: by identifying known patterns of attacks and machine learning to detect unusual patterns. For example, if a user logs in many times the system will trigger the alert. The machine learning system looks at patterns of human behavior and finds anomalies [6,9].

Once it has checked all the logs it keeps the information it has found. If it finds something it will alert us immediately. It also offers a simple way to visualize the data to see what is going on with our systems.

Our new system is different from other log analysis systems. It's easy to use and processes a lot of data. Is good at finding threats. So it's a good way to improve security [5].

4.1 Architecture Overview

The solution is built from scratch with a cloud event-driven and cloud native system [3,4] to facilitate management of large amounts of log data. Logs from various such as operating systems, applications and so on, are gathered and stored centrally with AWS S3. When there is a flow of new log information, the system uses AWS Lambda functions to process the new real-time. You don't have to keep servers running to process the information in real-time. The second step is to use an engine to detect information using a detection engine. The detection engine uses a combination of rules and machine learning to detect known threats and unknown threats.

The proposed system contains three primary components: cloud storage, serverless processing, and detection engines.

Fig. 1 shows the overall workflow of the proposed system, illustrating how cloud storage, serverless processing, and detection mechanisms works together as part of a unified pipeline.

4.2 Database Schema

As shown in tables I and II, the system maintains separate schema for processed log data and security alerts. This separation helps in efficient data management and allows faster querying for both regular log analysis and anomaly detection tasks.

Table 1: Processed Logs Table Schema

Attribute	Type	Description
user_id	String (PK)	Unique user identifier
timestamp	String (Sort Key)	Time of login event
ip	String	IP address of user
status	String	Login status (success/failure)

Table 2: Security Alerts Table Schema

Attribute	Type	Description
user_id	String (PK)	Identifier of affected user
timestamp	String (Sort Key)	Time of detected anomaly
ip	String	Source IP address
threat_flag	String	Type of detected anomaly

4.3 Unified Data Model

To maintain consistency, the system structures log data using the previously defined four-dimensional model, ensuring uniform representation across all log sources.

- When: Timestamp of the event
- What: Type of activity performed
- Who: User or system responsible
- Where: Source location or IP address

To formally represent the structure of log data, each log entry can be modeled as:

$$L = \{W_n, W_t, W_u, W_l\} \tag{1}$$

where L represents a structured log event, W_n denotes *When*, W_t denotes *What*, W_u denotes *Who*, and W_l denotes *Where*. This structure makes the data easier to analyze and helps in identifying relationships between different events more effectively [2].

4.4 Hybrid Detection Mechanism

The system is based on a combination of rule-based and machine learning to detect threats. Rule-based methods are good at detecting common patterns, such as multiple failed login attempts [1,9]. However, threats are not always predictable. To handle this, machine learning is applied to user behavior and identifies anomalous behavior [9], therefore, providing the ability to detect potential threats that would not be identified by traditional rule-based detection. The rule-based method of detecting threats may be described as follows:

$$S_r = \begin{cases} 1, & \text{if } N_f > T \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where s_r represents the rule-based suspicion flag, N_f is the number of failed login attempts, and T is the predefined threshold.

To capture deviations from normal behavior, an anomaly score is computed as:

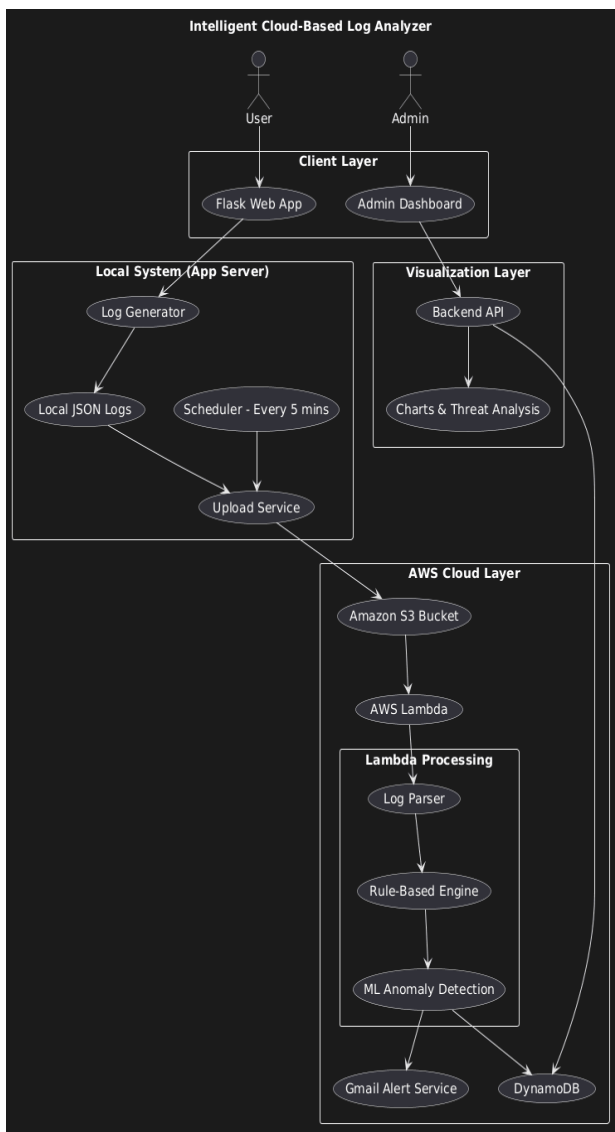


Figure 1: Architecture of the Proposed Cloud Based Log Analyzer

$$S_a = \frac{|x-\mu|}{\sigma} \tag{3}$$

where x is the observed feature value, μ is the mean behavior, and σ is the standard deviation.

The final hybrid suspicion score is computed by combining both approaches:

$$S_h = \alpha S_r + (1 - \alpha) S_a \tag{4}$$

where s_h is the final hybrid suspicion score, s_r is the rule-based score, s_a is the anomaly score, and α is the weighting factor between the two detection mechanisms.

5. METHODOLOGY

The system follows a structured multi-stage process for log analysis, where each stage handles a specific task from data collection to alert generation.

5.1 Data Collection

Logs are collected from the web application. These logs are continuously sent to cloud storage for further processing.

5.2 Preprocessing

In this stage, the collected logs are processed to extract useful information. The data is then cleaned and standardized to ensure consistency. This step is important because properly structured data improves the accuracy of further analysis. Once processed, the data is ready for the next stage.

5.3 Feature Extraction

The processed logs used to extract key attributes, including frequency of logins, pattern of IP addresses, and time of activity [12], are analyzed using both rule-based and machine-based detection methods.

5.4 Detection Process

Detection is done using two phases: Firstly, using predetermined regulations, any potentially large threats will be assessed and defined as either present or absent; secondly through Existing Data Analysis Processing/Files." This is where object anomaly detection systems take place creating various combinations of unusual behavior displaying what wouldn't ordinarily fall within normal limits. [6,7].

5.5 Alert Generation

When suspicious activity is detected, alerts are generated and stored. These alerts help administrators in further investigation and enable timely response to potential threats.

6. EVALUATION AND DISCUSSION

The system was assessed for its scalability, cost efficiency, and its threat detection capabilities. The serverless design of the system enables it to scale workloads automatically. In addition, the use of cloud services help reduce overall operational costs [4]. The combination of both rule-based and machine learning enhances the system's capability to identify and unknown threats. But the effectiveness of anomaly detection on the quality of the data and features selected for analysis [8,9]. The serverless architecture allows the system to automatically scale to accommodate different loads intervention.

Table 3: Comparison of Log Monitoring Solutions

Feature	Traditional Systems	SIEMs	Proposed
Architecture	Server / Containers	Hybrid	Serverless
Scalability	Manual / ASG	Enterprise	Automatic
Detection	Rule-Based	Advanced Rules	Hybrid ML
Overhead	High	Medium	Low

Cost	Fixed Infra	License Fee	Pay-per-use
------	-------------	-------------	-------------

As shown in Fig. 2, the proposed serverless system significantly reduces operational costs compared to traditional SIEM solutions, making it more suitable for scalable deployments.

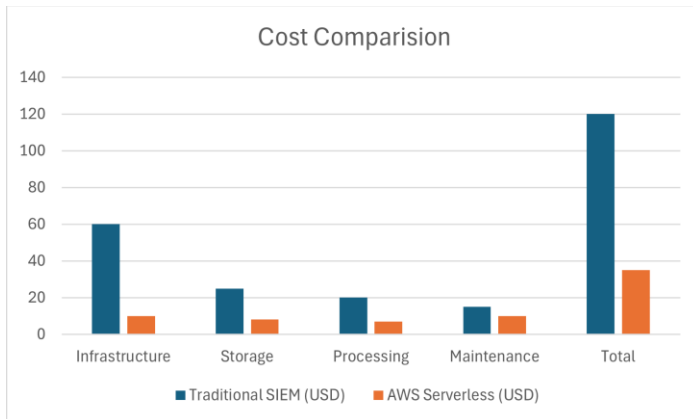


Figure 2: Cost Comparison Between Traditional SIEM and Proposed AWS Serverless System

As shown in Fig. 3 shows that the hybrid detection approach achieves higher accuracy compared to individual rule-based and machine learning methods, demonstrating the effectiveness of combining both techniques.

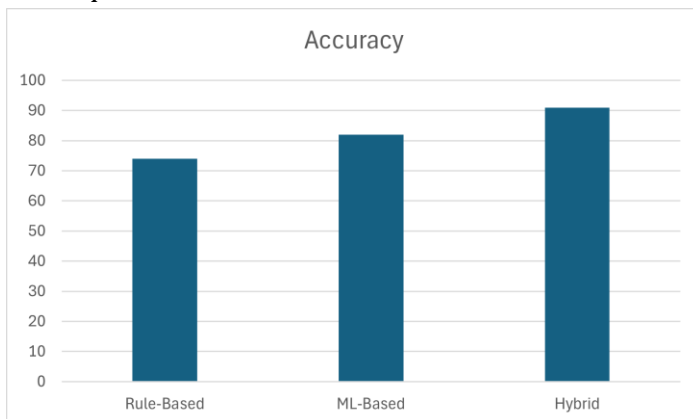


Figure 3: Detection Accuracy Comparison of Rule-Based, MLBased, and Hybrid Approaches

7. USE CASE SCENARIO

To illustrate how the proposed new system operates, let's assume an attacker is trying to gain unauthorized access by attempting to log in with several different short period of time.

Most log monitoring tools detect these types of attacks by monitoring failed attempts. However, attackers can avoid detection by these systems by using several using multiple usernames and IP addresses.

The new system does this by examining the login activity more effectively. Recognizes things like multiple from a single source and classifies them as suspicious. This enables the system to identify attacks and then and generate real-time alerts.

Meanwhile, the system monitors user behavior in varying criteria such as time of day, location and access system, Even if the hacker is trying to be stealthy, the system can identify unusual behavior.

For instance, if a user typically logs in from a single then tries to access the system from several locations at once, this is detected as suspicious and an alert is generated. Time, when this occurs, realtime notifications are sent to as it happens, so administrators can respond accordingly, such as blocking the IP address or launching an investigation. This method enhances the accuracy of detection and minimizes false positives by combining known patterns and behavioral anomalies.

8. PROJECT OUTCOMES

Table 4: Project Outcomes

Category	Outcome	Description
System Architecture	Scalable Serverless Pipeline	Utilizes AWS S3 and Lambda for event-driven log processing, eliminating dedicated servers and reducing operational overhead.
Data Standardization	Unified 4W Log Model	Organizes logs into When, What, Who, and Where categories, ensuring consistency and efficient analysis.
Security Detection	High-Fidelity Anomaly Alerts	Combines rule-based detection with machine learning to identify both known threats and anomalous behavior.
Operational Insight	Administrative Visibility	Enables monitoring of user activity, system health, and access patterns

		through structured logs.
Incident Response	Automated Threat Mitigation	Generates real-time alerts for suspicious activities, enabling timely response to security incidents.
Data Persistence	Optimized Log Retrieval	Stores processed logs in DynamoDB for efficient querying and fast retrieval.

The system has a number of benefits related to scalability, consistency and security.

It provides us with an economical approach to log analysis and can be scaled up if we need to do that, which is great useful for looking at logs. The system also makes use of a type of data that means we can get consistent results and that helps it easier to work with the data.

The system uses different methods to detect problems, makes the system more secure and this is because it can find threats we know about and threats we don't know about and the system can do this very effectively. Moreover, the hybrid detection strategy enhances system security by allowing for both known and unknown threats. It also enhances operational visibility by classifying log data into categories

9. FUTURE WORK

The system proposed in this paper lays the groundwork for enhanced security monitoring in the cloud. The following paths are outlined for further developing its functionality into a more proactive and autonomous platform.

- 1. Learning Behavior with Autoencoders and Continual Learning:** The anomaly detection engine is currently based on thresholds and rule-based scoring. Future work will incorporate deep learning models like Autoencoders to develop a behavioral model of each user and system [11]. With the use of lifelong learning techniques, the system will to more examples of normal activity [9], and improve its to more data over time, evolving from reactive to behavior intelligence.
- 2. Proactive Defense through SOAR:** The existing system alerts of threats but does not act on those threats. Integrating Security Orchestration, Automation, and Response (SOAR) capabilities [1] will allow the system to perform actions automatically - like quarantining a suspicious IP address, authentication - as soon as suspicious conduct is confirmed. authentication - as soon as suspicious activity is detected, cutting the response time down from minutes to milliseconds.
- 3. Explainable AI (XAI) for Actionable Alerting:** Security analysts can best benefit from knowing *why* it was raised, not just *if* it was raised. Using explanation systems such as SHAP or LIME will enable the system to provide explanations with each alert [9,11] - for instance, listing that an attempt was made because the attempt was unknown location at an odd time and attempting trying to access a critical system. This builds analyst confidence and false positive investigation times.

4. Team Forensics with a Joint Investigation Workspace: Workspace: In security emergency situations, it is not uncommon may need to collaborate on a single alert. A common "War Room" area, where analysts can comment logs, investigate incidents, and make decisions investigations in real time would help speed up times and collaboration [10].

5. Integration with Threat Intelligence Feeds: Linking the system to world-wide threat databases such as AlienVault OTX or VirusTotal will allow known malicious IP addresses, domains and file hashes to be identified blocked as soon as they come into contact with the environment [5, 12]. This enables the system to take advantage of security knowledge of the entire community, rather experiences, rather than just those from its own logs.

6. Visualization Dashboards: Incorporating real-time visual dashboards will enable administrators a real-time picture of system events, threat timelines. These dashboards, along with the improvements above, will turn the envisioned system into a powerful, "smart" into an holistic, smart security operations system - not only detecting threats but understanding, and automatically reacts to them [1,9].

10. CONCLUSION

In this paper, a cloud-based log analysis system has been presented that focuses on scalability, efficiency and intelligent threat detection. The proposed approach addresses key limitations of traditional log analysis systems by leveraging AWS cloud services and a structured data model. The proposed system combines a hybrid detection approach with reliability and flexibility. The proposed system demonstrates the potential benefits of cloud technology in developing efficient security monitoring tools

ACKNOWLEDGEMENT

The authors wish to thank the Department of Information Technology at Chaitanya Bharathi Institute of Technology, Hyderabad, for providing the infrastructure and academic environment that made this work possible. Finally, we thank our peers and reviewers whose feedback helped sharpen the ideas presented in this paper.

REFERENCES

- [1] M. Khayat, E. Barka, M. A. Serhani, F. Sallabi, K. Shuaib, and H. M. Khater, "Advanced Techniques for Alert Management in Security Information and Event Management Systems With Ensembled Deep Learning, Hybrid Optimization, and Multi-Feature Extraction," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 7349–7368, 2025, doi: 10.1109/OJCOMS.2025.3603000, <https://ieeexplore.ieee.org/document/11142305>.
- [2] A. Oliner, A. Ganapathi, and W. Xu, "Designing a Unified Cloud Log Analytics Platform," in *Proc. IEEE/USENIX HotCloud*, 2012. [Online]. Available: <https://ieeexplore.ieee.org/document/7870995>
- [3] J. Roberts and C. Chapin, "Event-Driven Serverless Architectures on Cloud Platforms," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, 2018, pp. 1–8.
- [4] R. Poorvadevi, Surendar H and SriRamakrishnan S, "Serverless Data Processing Using AWS," in 2025 8th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2025, doi: 10.1109/ICOEI65986.2025.11013090.
- [5] A. B. Nassif, M. A. Talib, Q. Nasir, H. Albadani, and F. M. Dakalbab, "Machine learning for cloud security: A systematic review," *IEEE Access*, vol. 9, pp. 20717–20735, 2021, doi: 10.1109/ACCESS.2021.3054129.
- [6] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.
- [7] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-Based Anomaly Detection," *ACM Trans. Knowl. Discovery Data*, vol. 6, no. 1, 2012.
- [8] C. C. Aggarwal, *Outlier Analysis*, 2nd ed., Springer, 2017.
- [9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, 2009.
- [10] K. Kent and M. Souppaya, *Guide to Computer Security Log Management*, NIST Special Publication 800-92, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2006.
- [11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [12] W. Xu, L. Huang, A. Fox, D. Patterson, and M. Jordan, "Detecting Large-Scale System Problems by Mining Console Logs," in *Proc. ACM Symp. Oper. Syst. Princ. (SOSP)*, 2009, pp. 117–132.