Explainable and Tamper-Resistant Real-Time Web Attack Detection System

Ramya Shree M
Computer Science and Engineering
Christ University
Bengaluru, India
ramyashree.m@mtech.christuniversity.in

Baburaj E
Computer Science and Engineering
Christ University
Bengaluru, India
baburaj.e@christuniversity.in

Addapalli Krishna
Computer Science and Engineering
Christ University
Bengaluru, India
adapalli.krishna@christuniversity.in

***Abstract*—The modern web systems generate large amounts of security logs that are generally verbose, semi-structured and not readily readable during response to an incident or when exploring the legal side of an incident. Traditional Web Application Firewalls (WAFs) and Security Information and Event Management (SIEM) engines are largely preoccupied with the blocking of traffic or an aggregate in a centralized manner, yet, they do not often offer deterministic explainability and real-time cryptographic integrity verification of individual log records. To close this gap, this essay presents a lightweight, real-time, web attack detection and explainable log forensics system that is named, SecureLog. The deterministic rule based signature engine that is used by SecureLog is achieved through optimized regular expression matching to make decisions on classification that are transparent and reproducible. Once every malicious request is identified it is categorized, summarized in plaintext and hashed with the SHA-256 cryptographic hash at ingestion in order to guarantee verifiable forensic integrity. This was experimented using the official test set of the CSIC 2010 HTTP Dataset that had 61,065 HTTP request(s) (36,000 benign, 25,065 anomalous). The accuracy of detection (91.67), FalsePositiveRate (0) and processing latency (roughly 0.0113 ms per request) are ultra-low as shown experimentally. SecureLog offers a legal and computationally-efficient substitute to the probabilistic generalization, which might be applicable in the low-scale web infrastructure environments by prioritizing the features of availability, interpretability, and deterministic behavior.**

***Index Terms*—Web Application Security, real-time Intrusion Detection, Rule-based Signature Detection, explainable security analytics, digital forensics, log analysis, SHA-256 Hashing, determinate detection architecture, WebSocket-based monitoring, CSIC 2010 Dataset.**

I. Introduction

The web applications are currently the primary evolution of the digital services, and all of them are based on the banking platform to academic portal and electronic commerce systems. The freer an individual is, the more he or she is subjected to cyber threats. Some of the most common destructive vectors of attacks are SQL Injection (SQLi), Cross-site Scripting (XSS) as well as Path Traversal attacks. Such attacks exploit impairments in input validation and application logic that would allow the attacks to alter databases, inject malicious code, or access restricted files in a system.

To counter any threats of this nature, organizations tend to deploy Web Application Firewalls (WAFs) such as the ModSecurity which heavily rely on known rule sets which they apply to block malicious HTTP requests. Despite being useful in the process of eliminating familiar patterns of attack, such systems may equally generate rule-based logs that are difficult to read unless one has special expertise. They mostly focus on prevention not on structural forensic elucidation. Network-based intrusion detection tools also offer defensive actions to the usage of packet-level traffic. These systems however focus on low level network signatures and in most cases require additional tools to reconstruct application-layer attack narratives. As a result, the gap between the time of detection and the definite forensic reporting remains big. Enterprise level SIEM systems, including those developed by Elastic NV, attempt to visualise and centralise large volumes of log data. Nevertheless, these platforms are influential, however, they demand a great deal of computing power, complex installation, and maintenance. Better still, they do not always provide per-request cryptographic verification when creating logs, which is especially critical in an environment where the evidence integrity is the need.

Over the past several years, a tendency has been observed in the body of work of an increasing number of studies focused on machine learning-based models of intrusion detection [6], [10]. These will increase the flexibility and detection of obfuscated or zero-day attacks [3], [8], [9]. They do introduce probabilistic uncertainty, high cost of computing and false positives that may disrupt genuine users [1], [2], [12]. Recent advancements also explore LLM-based collaborative rule generation and heterogeneous log analysis [4], [11]. Furthermore, human-centered explainable AI frameworks emphasize interpretability and trust in IDS decision-making [5], [7]. Adaptive and deception-based cybersecurity defense mechanisms have also been proposed to strengthen security architectures [13], [14].

The adoption of adaptive complexity can be less applicable and less beneficial than deterministic behavior and explainability in the academic evaluation setting and legal review settings. To address these flaws, this paper provides a novel lightweight and real time attack detection system and explainable log forensics system known as SecureLog. Unlike large SIEM systems or probabilistic ML pipelines, SecureLog is also based on a deterministic and rule-based signature engineering, offering transparent and reproducible choices of

classification. The system tracks web servers, shows the web server logs in real time, and recognizes malicious patterns by regular expression matching best with regular expressions and generates plain-English summaries of the threats each of which is identified. Notably, every flagged request is hashed when it is ingested with SHA-256 that develops a forensic ledger that is tamper evident. SecureLog is a solution to the lack of protection between real-time identification and legally admissible forensic reporting by concentrating on the availability.

II. Related Work

Explainable artificial intelligence techniques have been integrated into intrusion detection systems to improve model interpretability and transparency. The L-XAIDS framework utilizes LIME explanations to provide human-understandable reasoning behind attack detection decisions [1].

A lightweight ensemble learning based intrusion detection framework has been proposed to improve detection accuracy while maintaining computational efficiency. The system combines multiple machine learning models with explainable AI techniques to provide interpretable cybersecurity analytics [2].

Deep learning based intrusion detection models using convolutional neural networks have been explored for Internet of Things environments. The approach integrates explainable AI mechanisms to provide clear insights into detected network anomalies and cyber threats [3].

Large language models have recently been applied to generate and repair intrusion detection rules through collaborative AI techniques. The GRIDAI framework demonstrates how LLMs can automatically improve rule-based security systems and enhance threat detection capabilities [4]. Human-centered explainable AI techniques have been explored to improve transparency in intrusion detection systems. A deep learning based framework was proposed to enhance security decision making while providing interpretable attack explanations [5].

Recent surveys highlight the rapid evolution of intrusion detection systems integrating machine learning, deep learning, and hybrid approaches. These studies also identify challenges such as scalability, interpretability, and detection of sophisticated attacks [6].

Explainable AI has been widely adopted to improve trust and interpretability in machine learning based intrusion detection systems. Evaluation frameworks demonstrate that XAI techniques can effectively clarify model decisions for cybersecurity analysts [7].

Explainable intrusion detection approaches for IoT environments focus on identifying network anomalies while providing interpretable insights. Such systems improve transparency and help administrators understand attack patterns in resource-constrained IoT networks [8].

An intelligent and explainable intrusion detection framework has been proposed for Internet of Sensor Things environments. The system integrates machine learning models with explainability techniques to enhance attack detection and system transparency [9].

Deep learning has become a dominant approach for intrusion detection in emerging technologies such as IoT, cloud, and edge computing. Comprehensive surveys highlight the strengths of deep neural architectures while discussing challenges like computational cost and explainability [10].

Large language models have recently been investigated for cyberattack detection through anomaly analysis across heterogeneous system logs. The approach demonstrates the ability of LLMs to correlate complex log patterns and detect sophisticated cyber threats [11].

The LENS-XAI framework introduces a lightweight and explainable intrusion detection model using knowledge distillation and variational autoencoders. This method aims to achieve scalable cybersecurity protection while maintaining model transparency and efficiency [12].

Adaptive deception frameworks have been proposed to strengthen cybersecurity defense mechanisms through behavioral analysis. Such systems dynamically respond to attacker activities and improve threat intelligence by misleading adversaries [13].

FLARE presents a lightweight aggregation mechanism for evaluating intrusion detection models in IoT environments. The framework focuses on feature-based evaluation to enhance robustness and efficiency in resource-constrained systems [14].

An ensemble deep learning approach combined with explainable AI techniques has been proposed for cybersecurity intrusion detection. The model improves classification accuracy while providing interpretable insights into attack detection decisions [15].

Explainable AI based intrusion detection methods for IoT systems aim to improve transparency and trust in automated security systems. These approaches allow security analysts to understand the reasoning behind attack detection and system alerts [16].

III. Methodology

The operational methodology of SecureLog would have been in such a way that it becomes responsive to real time, classification deterministic and cryptographic integrity of all security event throughout its life cycle. It is founded on formal processing pipeline and beginning with log ingestion step and ending in safe forensic recording and visualization.

A. Real Time Log Ingestion

SecureLog continuously reads the web server access log capturing the events as they occur. No periodic polling is used and asynchronous input output mechanisms using Python asyncio are used to track the log file in real time at the backend. The web server writes a new entry to the server; the ingestion engine will immediately read the line that has just been added to it without blocking or interrupting the file. It implies that the surveillance will not interfere with normal processes of the servers and will ensure the provision of their services.

B. Threat Classification

After being consumed, every one of the log entries is read and decoded to extract helpful attributes, such as, but not limited to, source IP address, method of the HTTP request, requested resource, and the content of the payload. The extracted payload is analyzed with the aid of a deterministic rule based signature engine. The system compares the request to attack patterns, which have been known to be in relation to SQL Injection, Cross Site Scripting and Path Traversal. In the event of a match, the request can be classified according to the nature of attack. This is then assigned a score of severity based on the perceived impact and the degree of risk of the identified vector. The default allow policy considers requests whose signature is not known as a benign request.

C. Cryptographic Hashing

In order to retain its forensic integrity, every malicious event that is identified will be hashed with a cryptographic hash after it has been classified. The system combines the raw log entry, the time at which the attack took place, the attack category, and the severity score into a structured one. Then a digital fingerprint of this record is computed using an SHA 256 hash. This ensures that any manipulation made on the evidence stored can be detected in this way boosting the tamper evident documentation and a good chain of custody.

D. Storage and Transmission

Once processed and hashed the final forensic record is stored in SQLite database locally as a long term archival and audit database. At the same time, this system also transmits the alert to the frontend WebSocket dashboard through a persistent WebSocket connection. This two-way communication process allows the security personnel to view the identified threats in almost real time that by default, are milliseconds after the original log has been created.

IV. Proposed System Architecture

SecureLog architecture has embraced the layered system design in order to offer modularity, real time processing and forensic integrity. The system will have three primary layers namely Data Ingestion Layer, the Analytical Processing Layer and the Presentation and Ledger Layer. Each of the layers possesses a specific activity within the security monitoring pipeline.

A. Data Ingestion Layer

At the first level, the raw web traffic data is gathered. Apache web server also logs access to the HTTP each time a user accesses the web application. These logs are monitored by the Real Time Log Monitoring module all the time. The system is an asynchronous processing that reads in the new entries that are being added to the log in real time and does not interfere with the normal operations of the server. This will ensure availability of real time responsiveness and non-existence of delay in threat detection.

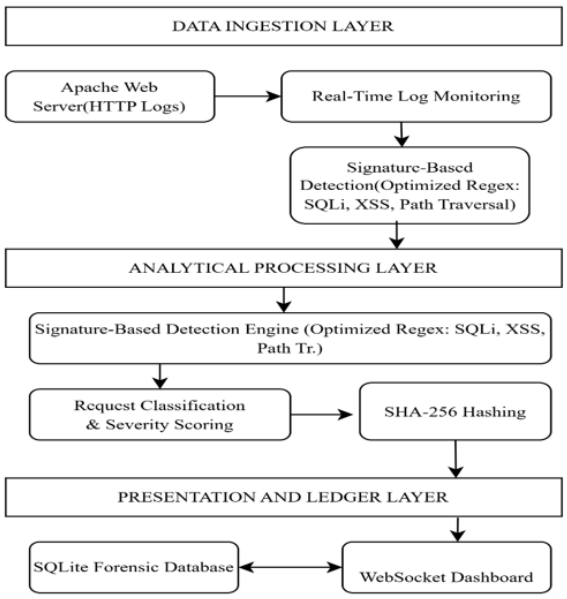

Fig. 1. SecureLog follows a three-layer modular architecture

B. Processing Layer Analysis

Once a log record is taken, it gets transferred to Analytical Processing Layer, where security analysis is done. A detection engine that is used to match each request with optimized regular expressions is known as the Signature Based Detection Engine. The engine is matched with the malicious patterns that have been stored in the malicious SQL Injection, Cross Site Scripting and Path Traversal attacks. The request is then classified according to the type of the attack in case a malicious signature is detected. The severity score is then provided to demonstrate the potential effects of the threat. After classifying the event, the whole event record is hashed using SHA 256. The reason is to generate a special cryptographic fingerprint, thus, such that the forensic evidence data are not in a readable format that can be changed without it being detected.

C. Presentation and Ledger Layer

The final level is storage and visualization. The processed and hashed security event is stored in a SQLite forensic database that can be used in long term archiving and as audit trail. In the meantime, the system sends the alert to a Web-Socket driven dashboard. This enables security administrators to view detected threats in a couple of milliseconds after they are created by a log. The bifurcated storage and visualizing system ensures the evidentiary preservation and consciousness of functioning.

V. Experimental Evaluation

A. Dataset Description

Table 3 gives the distribution of the 2010 HTTP testing data provided by the CSIC that was to be evaluated.

TABLE I
PERFORMANCE EVALUATION METRICS

Metric	Value
Accuracy	91.67%
Precision	100%
Recall	92.82%
F1 Score	96.25%
False Positive Rate	0%

TABLE II
PERFORMANCE METRICS OF ATTACK DETECTION SYSTEM

Metric	Count
True Positives	23,265
True Negatives	36,000
False Positives	0
False Negatives	1,800

TABLE III
DISTRIBUTION OF WEB SERVER REQUESTS

Category	Number of Requests
Normal Requests	36,000
Anomalous Requests	25,065
Total	**61,065**

TABLE IV
SYSTEM PERFORMANCE AND COMPUTATIONAL OVERHEAD

Metric	Result
Detection Latency	$\sim$0.0113 ms
SHA-256 Hashing Overhead	$\sim$0.0025 ms
Total Execution Time	$<$ 0.02 ms

The dataset is a mix of legitimate and malicious HTTP requests so that the detection capability can be evaluated equally.

B. Confusion Matrix and Derived Metrics

The confusion matrix presented in Table 2 represents the results of experimental assessment. According to the confusion matrix, the performance measures were estimated as indicated in Table 1.

The zero false positive indicates the deterministic rule based classification whereas the recall indicates the impact of missed.

C. Performance Overhead

The computational overhead that was measured during runtime evaluation is summarized in Table 4. The findings show ultra low latency which proves the appropriateness in real time application.

VI. RESULTS AND DISCUSSION

SecureLog signature based architecture is deterministic, and hence, the False Positive Rate is 0 i.e. not a single request by a legitimate user was reported as a malicious request. It particularly applies to the real web application where false alarms can disrupt the user experience, lock legitimate customers and reduce system availability. The system guarantees that attacks are only made with clearly known malicious patterns by following through on the known malicious patterns. However, the emergence of 1,800 False Negatives indicates that there were evil requests which went beyond unidentified. The reason of such missed detections may be mainly attributed to the use of the evolution of sophisticated evasion procedures such as payload obfuscation, and changes of encoding or even minor changes in attack signatures that do not specifically fit within the set of rules. It is a known vulnerability of deterministic rule-based systems: they are very sure and open, but may fail to notice novel or inventively concealed patterns of attack. This finding elicits the conventional trade off of intrusion detection system design. Deterministic approaches are accurate, explainable, and operationally safe and probabilistic/machine learning approaches are likely to offer far greater generalization at the cost of false positives and less explainability. Compared to the most recent machine learning-driven intrusion detection systems that have been reported in the 2025-2026 literature, SecureLog lacks adaptive learning, and explainability, zero service disruption, and ultra-low latency become the priorities. Even though, the ML models may slightly be more efficient insofar as they may be trained to detect other types of attacks, they may lead to an increment of additional costs of computations, complexity, and false-alarm risk. SecureLog on the contrary is predictable in its behavior, open decision logic and real time performance which can be implemented in resource constrained environment.

VII. SECURITY AND FORENSIC INTEGRITY ANALYSIS

SecureLog is intended to identify web attacks as well as provide forensic reliability. The integrity and the possibility to trace and protect against misuse or manipulation makes the evidence admissible and credible in digital investigations. To resolve these issues, the system employs both cryptographic and structural protection as soon as there is the threat detected. On identification of a malicious log entry, it is immediately run through the hash-256 hash algorithm. This algorithm creates a record of a digital fingerprint of each record. In case of any alteration to the initial log data, a small alteration would give a completely new hash value and hence tampering would be easy to detect. This process ensures the authenticity in evidence and maintains the originality of the data. All the events identified are persistently stored in a structured SQLite database. The logs are kept in tight chronological sequence and contextual details like timestamps, request information and attack typology. SecureLog improves the clarity of interpretation by converting real server logs into a technical yet structured forensic documentation. SecureLog manages to maintain the forensic records tamper-evident, time-ordered,

and audit-friendly through timestamped logging, cryptographic hashing, and a well-designed storage design, as well as to enable digital investigation using them.

VIII. Future Work

Even though SecureLog has a high precision, zero false positives and ultra-low latency, future improvements can be made. Another improvement that can be made is the inclusion of a lightweight anomaly detection[] mechanism that will be used to supplement the already existing signature-based rules. This may assist in determining a few changed or never been detected attack patterns and still be computationally efficient. The other direction that should be noted is the incorporation of input normalization methods to deal with the encoded payloads or obfuscated payloads that are usually used to circumvent the rule-based systems. More modern web applications coverage would also be enhanced by including the body of a REST API request in the analysis. Future studies can also aim at considering hybrid models of detection, the rule based transparency and the adaptive scoring techniques. Also, it would be beneficial to facilitate the system by making it run on larger and more diverse sets, to increase its generalization power. The aim of such improvements is to strengthen the strength and flexibility whilst maintaining the fundamental benefits of real-time performance and explainability.

IX. Conclusion

SecureLog is an explanation-based, lightweight and deterministic web attack detection system that not only provides real-time detection of threats but also a forensic documentation of legal quality. In contrast to most intrusion detection systems that are blunt force techniques, which are often complex intrusion detection systems that extensively utilize computationally intensive machine learning models, SecureLog uses a simple rule-based system of intrusion detection that is more focused on clarity, predictability, and operational reliability. The CSIC 2010 HTTP dataset, which was used to test the system, has the result of 91.67 percent accuracy, one false positives, and a sub-milliseconds response time. The lack of false alarm provides the system with continuous service availability and the ultra-low latency makes the system suitable on a high traffic and resource-intensive environment. Though not all detached or well-structured attacks can be captured by deterministic signatures, the system is very precise and reliable in detecting well-defined web threats. SecureLog provides a bridge between digital forensic preparedness and cybersecurity monitoring through the combination of real-time detection, tamper-evident logging cryptographic hash, and structured forensic storage. Its focus on explainability, low-computational cost, and evidentiary integrity make it a viable substitute to the heavyweight SIEM platforms and diverse machine learning-based intrusion detection systems especially in the environment where transparency, cost-efficiency, and legal reliability matter a great deal.

References

[1] A. E. Muhammad, K. C. Yow, N. Bacanin-Džakula, *et al.*, "L-xaids: A LIME-based Explainable AI Framework for Intrusion Detection Systems," *Cluster Computing*, 2025.

[2] "Lightweight ensemble learning based intrusion detection framework with explainable artificial intelligence," *Engineering Applications of Artificial Intelligence*, vol. 112936, 2026.

[3] F. Ebrahimi, R. Javidan, R. Akbari, and Y. Hosseini, "Intrusion Detection in the Internet of Things Using Convolutional Neural Networks: An Explainable AI Approach," *Cybersecurity*, 2025.

[4] J. Li, Y. Chai, L. Du, *et al.*, "GRIDAI: Generating and Repairing Intrusion Detection Rules via LLM-Based Collaboration," *arXiv preprint*, 2025.

[5] M. M. Jahid Ayan, M. S. Rashid, T. A. Hassan, *et al.*, "Human-Centered Explainable AI for Security Enhancement: A Deep Intrusion Detection Framework," *arXiv preprint*, 2026.

[6] A. Hozouri, A. Mirzaei, and M. Effatparvar, "A Comprehensive Survey on Intrusion Detection Systems with Advances in Machine Learning and Emerging Challenges," *Discover Artificial Intelligence*, 2025.

[7] V. Z. Mohale and I. C. Obagbuwa, "Evaluating Machine Learning-Based Intrusion Detection Systems with Explainable AI: Enhancing Transparency and Interpretability," *Frontiers in Computer Science*, 2025.

[8] "Explainable AI-based intrusion detection in IoT systems," *IEEE Internet of Things Journal*, May 2025, p. 101589.

[9] "An Intelligent and Explainable Intrusion Detection Framework for Internet of Sensor Things," *Journal of Network and Computer Applications*, 2025.

[10] E. C. P. Neto, S. Iqbal, S. Buffett, M. Sultana, and A. Taylor, "Deep learning for intrusion detection in emerging technologies: A comprehensive survey and new perspectives," *Artificial Intelligence Review*, vol. 58, no. 11, p. 340, 2025.

[11] Y. Chagna and A. Goldschmidt, "Next-generation cyberattack detection with large language models: Anomaly analysis across heterogeneous logs," *arXiv preprint arXiv:2602.06777*, 2026. [Online]. Available: https://arxiv.org/abs/2602.06777

[12] M. A. Yagiz and P. Goktas, "LENS-XAI: Redefining lightweight and explainable network security through knowledge distillation and variational autoencoders for scalable intrusion detection in cybersecurity," *arXiv preprint arXiv:2501.00790*, 2025. [Online]. Available: https://arxiv.org/abs/2501.00790

[13] B. A. Al-Zahrani, "Adaptive deception framework with behavioral analysis for enhanced cybersecurity defense," *arXiv preprint arXiv:2510.02424*, 2025. [Online]. Available: https://arxiv.org/abs/2510.02424

[14] B. Boswell, S. Barrett, S. Rajaganapathy, and G. Dorai, "FLARE: Feature-based lightweight aggregation for robust evaluation of IoT intrusion detection," *arXiv preprint arXiv:2504.15375*, 2025. [Online]. Available: https://arxiv.org/abs/2504.15375

[15] A. Alabdulatif, "A novel ensemble of deep learning approach for cybersecurity intrusion detection with explainable artificial intelligence," *Applied Sciences*, vol. 15, no. 14, 2025, doi: 10.3390/app15147984.

[16] S. Li and N. Saxena, "Explainable AI-based intrusion detection in IoT systems," *Internet of Things*, vol. 31, 2025, doi: 10.1016/j.iot.2025.101589.