RESEARCH ARTICLE OPEN ACCESS

Automated Soc Log Analysis Using Machine Learning

Mr.J.Madhukarthick ,AP*,G.Vengatraman**,R.Vishwa**, R.Vignesh**,V.Nithish**

*Assistant Professor Computer Science and Engineering, A.V. C. College of Engineering, Mannampandal, Mayiladuthurai,

**UG Students -Computer Science and Engineering, A.V. C. College of Engineering, Mannampandal, Mayiladuthurai,

India

ABSTRACT:

The rapid evolution of cyber threats necessitates intelligent and proactive security monitoring systems capable of detecting both known and unknown attack patterns. Traditional Security Operations Center (SOC) systems rely heavily on rule-based detection and static log analysis, which limits their ability to identify advanced threats. This paper presents an AI-driven SOC framework that integrates machine learning–based anomaly detection, deception-based honeytoken mechanisms, event-driven processing, and Retrieval-Augmented Generation (RAG)–based intelligent investigation. The proposed system consists of a simulated victim application, a centralized SOC backend, and a real-time monitoring dashboard. Logs are generated dynamically by the victim application and ingested via HTTP APIs. The backend processes logs using a multi-layered detection pipeline, including rule-based detection, enumeration detection, User and Entity Behavior Analytics (UEBA), and machine learning–based anomaly detection using Isolation Forest and ensemble techniques trained on the CICIDS2017 dataset. A dynamic honeytoken-based deception system is implemented, where fake endpoints, files, and credentials are generated and rotated periodically to detect malicious interactions. The system also incorporates an event-driven architecture for scalable processing and integrates SOAR (Security Orchestration, Automation, and Response) principles to automate alert handling and response workflows. Furthermore, a RAG-based investigation layer powered by FAISS enables semantic querying and contextual analysis of security events. An Explainable AI (XAI) module provides human-readable explanations for detected threats. Experimental evaluation demonstrates that the proposed system effectively detects reconnaissance, credential abuse, and anomalous behavior with high accuracy and reduced false positives. The framework provides a scalable and intelligent solution for modern cybersecurity operations.

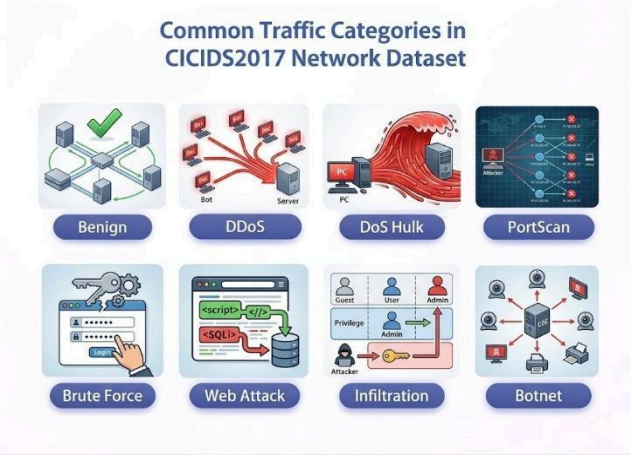
Keywords: SOC, Honeytokens, Isolation Forest, RAG, FAISS, SOAR, UEBA, XAI, Cybersecurity, CICIDS2017.

I.INTRODUCTION

The increasing dependence on digital systems has significantly expanded the attack surface of modern organizations. Cyber attackers employ sophisticated techniques such as reconnaissance scanning, lateral movement, credential abuse, and privilege escalation to compromise systems. Traditional security solutions, including firewalls and signature-based intrusion detection systems, are often reactive and ineffective against evolving threats. Security Operations Centers (SOCs) are responsible for monitoring and responding to security incidents. However, conventional SOC implementations rely heavily on rule-based detection and manual analysis, which leads to delayed responses and high false-positive rates. These limitations highlight the need for intelligent and automated security monitoring solutions.

This research proposes an AI-driven SOC framework that combines multiple advanced technologies, including machine learning, deception-based security, event-driven processing, and intelligent investigation mechanisms. The system introduces dynamic honeytokens to detect malicious interactions, integrates anomaly detection using

Fig 1



Isolation Forest models trained on CICIDS2017, and employs UEBA techniques to analyze user behavior patterns.

Additionally, the system incorporates aRAG-based investigation layer using FAISS for semantic retrieval of security events and an Explainable AI module to provide insights into detected threats. SOAR principles are integrated to automate alert handling and streamline incident response workflows. The proposed framework aims to enhance detection accuracy, reduce response time, and provide actionable intelligence for security analysts..

II.RELATED WORK

A. Security Information and Event Management (SIEM) Systems

Security Information and Event Management (SIEM) systems have been widely adopted for centralized collection and analysis of logs generated from applications, systems, and user activities. These systems provide visibility into system behavior by aggregating logs and applying correlation rules to detect potential security incidents. However, traditional SIEM solutions rely heavily on predefined rules and static detection logic, which limits their ability to identify unknown or evolving threats, particularly in application-level environments.

B. Intrusion Detection and Anomaly Detection

Intrusion Detection Systems (IDS) and anomaly detection mechanisms have been used to monitor system activities and identify suspicious behavior. While signature-based approaches are effective for detecting known threats, they fail to identify novel attack patterns. Anomaly-based detection techniques aim to address this limitation by identifying deviations from normal behavior. However, these approaches often require careful tuning and may generate false positives if not properly calibrated.

C. Deception-Based Security and Honeytokens

Deception-based security techniques, including honeytokens, have gained attention as proactive mechanisms for detecting malicious activity. Honeytokens are lightweight deceptive artifacts such as fake endpoints, configuration files, and credentials that are embedded within applications. Any interaction with these resources is considered highly suspicious, making honeytokens an effective method for detecting unauthorized access and attacker exploration. Recent research highlights the effectiveness of dynamic honeytoken generation and rotation in preventing attackers from identifying and bypassing these traps.

D. Machine Learning for Log Analysis

Machine learning approaches have been increasingly applied to application-level log analysis for anomaly detection and behavioral modeling. Techniques such as Isolation Forest and ensemble learning are used to identify abnormal patterns in user interactions and system events. Datasets such as CICIDS2017 are

commonly used for training and evaluating these models, although adaptation to application-level logs requires appropriate feature engineering and preprocessing.

E. User and Entity Behavior Analytics (UEBA)

User and Entity Behavior Analytics (UEBA) focuses on analyzing behavioral patterns of users within applications to detect anomalies such as unusual access patterns or repeated login failures. This approach is particularly useful for identifying insider threats and compromised accounts, as it captures deviations in normal user behavior over time.

F. Security Orchestration, Automation, and Response (SOAR)

Security Orchestration, Automation, and Response (SOAR) systems have been introduced to automate the handling of security alerts and streamline incident response workflows. By integrating detection mechanisms with automated response processes, SOAR reduces manual effort and improves response efficiency in SOC environments.

G. Retrieval-Augmented Generation (RAG) for Investigation

Recent developments in artificial intelligence have introduced Retrieval-Augmented Generation (RAG) techniques for intelligent investigation. By combining vector-based search methods such as FAISS with language models, RAG systems enable semantic querying of stored logs and security events. This approach enhances the ability of analysts to retrieve relevant information and gain contextual insights during investigations.

III. PROCEDURE

A. Log Generation in Victim Application

The system utilizes a simulated victim application that generates structured logs for every user interaction. These logs include essential metadata such as IP address, endpoint accessed, request method, timestamp, and user-related attributes. This approach ensures continuous monitoring of application-level activities and provides a realistic data source for security analysis.

B. Log Ingestion and Event Processing

Generated logs are transmitted to the SOC backend through HTTP-based APIs, enabling real-time ingestion. An event-driven architecture is implemented using an EventBus mechanism, which distributes incoming log events across multiple detection engines asynchronously. This design improves scalability and ensures efficient processing under high-load conditions.

C. Rule-Based and Enumeration Detection

The system incorporates rule-based detection mechanisms to identify known suspicious patterns, such as repeated access to sensitive endpoints. Additionally, an enumeration detection module identifies reconnaissance behavior by analyzing frequent probing of application routes, indicating potential attacker exploration.

D. Machine Learning–Based Anomaly Detection

A machine learning module based on Isolation Forest and ensemble techniques is integrated into the detection pipeline. The model is trained using the CICIDS2017 dataset and adapted for application-level log analysis. It identifies deviations from normal behavior and detects previously unseen attack patterns, enhancing the system's ability to handle zero-day threats.

E. User and Entity Behavior Analytics (UEBA)

The UEBA module analyzes user behavior patterns to detect anomalies such as unusual access frequency, abnormal resource usage, and suspicious login attempts. This component plays

a crucial role in identifying insider threats and compromised user accounts.

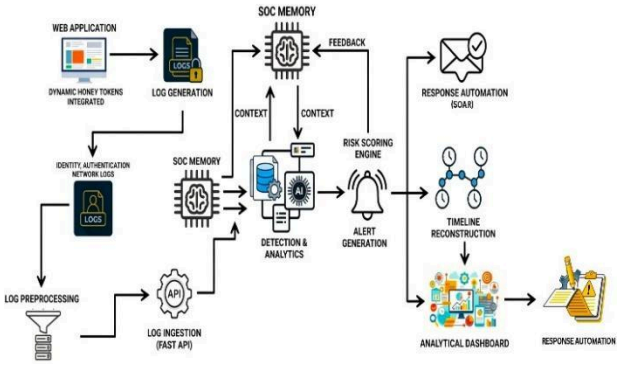


Fig 2

F. Honeytoken-Based Deception Detection

The system implements a dynamic honeytoken framework, where fake endpoints, configuration files, and credentials are generated and rotated periodically. These deceptive elements are exposed within the victim application. Any interaction with honeytokens is treated as a high-confidence indicator of malicious intent.

G. Risk Scoring and Alert Generation

Detected events are evaluated using a risk scoring mechanism that assigns severity levels based on multiple factors, including anomaly scores, behavior patterns, and deception triggers. Alerts are generated and stored for further analysis and visualization.

H. Attack Correlation and Timeline Analysis

An attack correlation engine links related events to form attack chains, providing a comprehensive view of attacker behavior. Timeline analysis organizes events chronologically, enabling analysts to understand the sequence and progression of attacks.

I. RAG-Based Intelligent Investigation

The system integrates a Retrieval-Augmented Generation (RAG) module powered by FAISS. Logs and alerts are converted into vector embeddings and stored in a vector database. Analysts can perform semantic queries, and the system retrieves relevant events to provide contextual insights.

J. Explainable AI (XAI) Module

An Explainable AI module generates human-readable explanations for detected threats. It combines outputs from machine learning models, behavioral analysis, and rule-based detection to provide interpretable insights for security analysts.

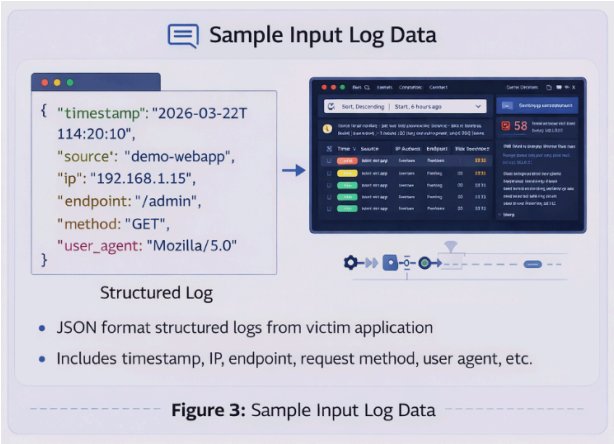


Figure 3: Sample Input Log Data

K. SOAR-Based Automation

The system incorporates Security Orchestration, Automation, and Response (SOAR) principles to automate alert handling and investigation workflows. This reduces manual intervention and improves response efficiency within the SOC.

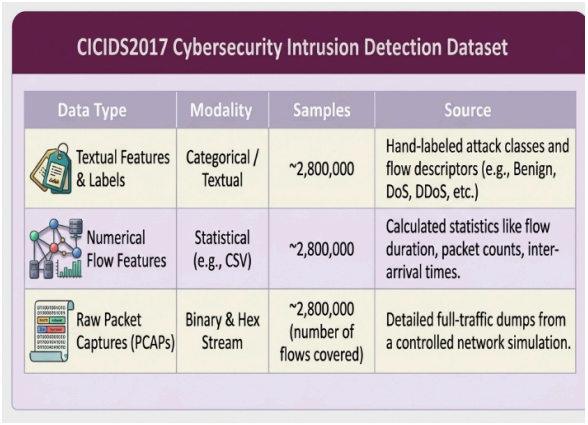
V.DATASETS

The machine learning component of the proposed system is trained using the CICIDS2017 dataset, a widely recognized benchmark dataset for intrusion detection research. This dataset was developed by the Canadian Institute for Cybersecurity and contains realistic benign and malicious traffic scenarios captured in a controlled environment.. The dataset provides labeled records that are useful for training and evaluating anomaly detection models. This dataset contains labeled

traffic data representing both benign and malicious activities, including attack categories such as brute-force attacks, denial-of-service, infiltration, and web-based attacks. The dataset provides a diverse range of features that enable effective modeling of normal and abnormal behavior.

In this work, the dataset is utilized to train an Isolation Forest–based anomaly detection model. Preprocessing steps such as feature selection, normalization, and removal of redundant attributes are performed to ensure data consistency and improve model performance. Although CICIDS2017 primarily contains network-level features, the learned patterns are adapted to application-level log analysis through feature abstraction and mapping techniques.

The trained model establishes a baseline of normal behavior and identifies deviations during real-time log processing. This approach allows the system to detect previously unseen anomalies in application interactions. The combination of simulated application logs and trained models provides a practical and scalable data-driven foundation for the proposed SOC framework.



The combination of benchmark training data and live application logs creates a hybrid data environment that supports both learned anomaly detection and real-time operational monitoring. Furthermore, alerts and processed logs are stored in **PostgreSQL**, while selected contextual events are indexed in the **SOC memory and vector retrieval layer** for intelligent investigation using FAISS-based semantic search.

VI.DISCUSSION

The proposed system demonstrates the effectiveness of integrating multiple advanced security mechanisms within a unified SOC framework. By combining rule-based detection, machine learning–based anomaly detection, and deception techniques, the system achieves comprehensive coverage of both known and unknown threats. Honeytokens serve as a highly reliable detection mechanism, as any interaction with deceptive resources directly indicates malicious intent.

The event-driven architecture significantly enhances system scalability and responsiveness. By distributing log events across multiple detection modules asynchronously, the system efficiently handles high volumes of data without performance degradation. This design ensures that detection processes operate in parallel, improving overall system throughput.

The integration of User and Entity Behavior Analytics (UEBA) adds an additional layer of intelligence by monitoring behavioral patterns over time. This enables the detection of subtle anomalies such as insider threats or compromised user accounts that may not be captured by traditional detection methods.

Furthermore, the RAG-based investigation layer introduces a novel approach to security analysis. By leveraging FAISS for semantic retrieval, the system enables analysts to perform contextual queries and obtain meaningful insights from historical logs. The inclusion of Explainable AI (XAI) enhances transparency by providing clear explanations for detected threats, improving analyst trust and decision-making.

Despite these advantages, the system relies on the quality of training data and appropriate

configuration of detection thresholds. Continuous monitoring and model updates are required to maintain effectiveness against evolving attack patterns.

VII.EVALUATION PARAMETERS

The performance of the proposed system is evaluated using multiple parameters that measure detection capability, efficiency, and analytical effectiveness.

Detection Accuracy:
Measures the system's ability to correctly identify malicious activities, including reconnaissance attempts, anomaly detection, and honeytoken interactions.

FalsePositiveRate:
Evaluates the frequency of incorrect alerts generated by the system. Maintaining a low false-positive rate is essential to avoid alert fatigue.

Response Time:
Represents the time taken from log ingestion to alert generation. The event-driven architecture contributes to reducing response latency.

Anomaly Detection Performance:

Assesses the effectiveness of the Isolation Forest and ensemble models in identifying abnormal behavior patterns.

UEBA Effectiveness:

Evaluates the system's ability to detect behavioral anomalies such as unusual user activity and login patterns.

RAG Query Effectiveness:

Measures the relevance and accuracy of responses generated by the RAG-based investigation module.

System Scalability:

Analyzes how efficiently the system processes increasing volumes of logs without performance degradation.

In real-world SOC environments, timely alert generation is essential for minimizing attack impact. The **risk score effectiveness** is another useful parameter, as it helps measure whether the assigned severity levels correctly reflect the seriousness of detected events.

Since the system incorporates a **RAG-based investigation module**, another evaluation parameter is the **quality of retrieval and query response relevance**. This can be assessed by checking whether the investigation layer retrieves the most relevant logs and provides contextually meaningful answers to analyst queries. Together, these evaluation parameters provide a comprehensive view of the framework's performance in detection, prioritization, and investigation support.

VIII.CHALLENGES AND FUTURE PREDICTIONS

Despite its advanced capabilities, the proposed system faces several challenges. One of the primary challenges is the adaptation of network-based datasets such as CICIDS2017 to application-level log analysis. This requires careful feature engineering to ensure meaningful representation of application behavior.

Another challenge is the tuning of machine learning models and detection thresholds to balance detection accuracy and false-positive rates. Improper configuration may lead to either missed detections or excessive alerts. Additionally, maintaining synchronization between dynamically generated honeytokens and the victim application is crucial for effective deception-based detection.

The computational overhead associated with embedding generation and vector search in the RAG module can also impact performance if not optimized properly. Efficient indexing and retrieval strategies are required to ensure scalability.

Future work can focus on integrating real-time data streams from production environments, enhancing anomaly detection using deep learning techniques, and expanding SOAR capabilities to include automated response actions such as blocking malicious users or isolating compromised components. Additionally, improving the RAG module with more advanced language models can further enhance investigation capabilities

IX.CONCLUSION

This paper presents a comprehensive AI-driven Security Operations Center framework that integrates machine learning, deception-based security, event-driven processing, and intelligent investigation techniques. The system addresses the limitations of traditional SOC solutions by enabling proactive detection, automated analysis, and efficient incident response.

The use of Isolation Forest and ensemble techniques enables effective detection of anomalous behavior, while honeytokens provide a reliable mechanism for identifying malicious interactions. The integration of UEBA enhances behavioral analysis, and the RAG-based investigation module improves the ability to analyze and interpret security events.

The implementation of SOAR principles further strengthens the system by automating alert handling and reducing manual workload. Overall, the proposed framework provides a scalable, intelligent, and efficient solution for modern cybersecurity challenges and demonstrates the potential of integrating multiple advanced technologies within a unified SOC architecture.

major contribution of the proposed system is the inclusion of a **RAG-based investigation layer** that allows analysts to query stored security events and receive contextual insights. Combined with the Explainable AI module, this makes the framework not only capable of detecting threats but also of helping analysts understand why alerts were raised. This is especially useful in modern SOC environments where decision support and alert interpretability are increasingly important.

REFERENCES

[1] G. Hinton, Y. Bengio, and Y. LeCun, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Vaswani et al., “Attention Is All You Need,” in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[4] Canadian Institute for Cybersecurity, “CICIDS2017 Dataset,” University of New Brunswick, 2017.

[5] M. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[6] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2008, pp. 413–422.

[7] K. Scarfone and P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS),” NIST Special Publication 800-94, 2007.

[8] MITRE Corporation, “MITRE ATT&CK Framework,” MITRE, McLean, VA, USA.

[9] L. Spitzner, *Honeypots: Tracking Hackers*. Boston, MA, USA: Addison-Wesley, 2003.

[10] N. Provos and T. Holz, *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Boston, MA, USA: Addison-Wesley, 2007.

[11] IBM Security, "Security Orchestration, Automation and Response (SOAR)," IBM Security Documentation, 2021.
[12] Meta AI, "FAISS: Facebook AI Similarity Search," Facebook AI Research.
[13] T. Brown et al., "Language Models are Few-Shot Learners," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
[14] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
[15] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2010, pp. 305–316.
[16] S. Axelsson, "The Base-Rate Fallacy and the Difficulty of Intrusion Detection," *ACM Transactions on Information and System Security*, vol. 3, no. 3, pp. 186–205, 2000.
[17] D. E. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.
[18] Elastic, "Security Information and Event Management (SIEM)," Elastic Security Documentation.
[19] Splunk Inc., "Security Operations and SIEM," Splunk Security Whitepaper.
[20] Gartner, "Security Orchestration, Automation and Response (SOAR) Market Guide," Gartner Research.