SOC Dashboard for Threat Intelligence and Log Correlation

Bhuvanyaa S[1], Dr. Angel S[2]
[1]Student, Department of Cyber Security, PSGR Krishnammal college for women
[2]Assistant Professor, Department of Cyber Security, PSGR Krishnammal college for women

***Abstract*—This chapter presents a practical study on the design and implementation of a Security Operations Center (SOC) dashboard for threat intelligence, log analysis, and event correlation using open-source technologies. The system was developed to emulate real-world SOC operations by enabling continuous monitoring, enrichment, and visualization of security events within a controlled research environment. Wazuh SIEM was utilized for centralized log collection and rule-based detection, MongoDB for structured data storage and historical analysis, and FastAPI for exposing security data through RESTful services. Interactive visualization was achieved using Dash and Plotly to provide analysts with clear insight into alerts and Indicators of Compromise (IOCs).**

To strengthen detection capabilities, the platform integrates external threat intelligence feeds such as AlienVault OTX and AbuseIPDB. These sources provide reputation-based context that enhances situational awareness and supports faster identification of suspicious network entities. A custom-built correlation engine processes incoming logs against known threat indicators, generating prioritized alerts that assist in proactive incident response while reducing manual investigation effort.

Automation plays a significant role in the proposed architecture. Scheduled ingestion pipelines, Bash-based execution scripts, and periodic threat intelligence updates ensure continuous system operation without requiring constant human supervision. The visualization layer further supports analytical decision-making by presenting attack patterns, severity levels, and event distributions in an intuitive dashboard format.

Overall, this work demonstrates that SOC-grade defensive capabilities can be reproduced using cost-effective and accessible open-source components. By combining log analytics, threat intelligence enrichment, and automated correlation, the proposed system provides a practical foundation for cybersecurity research, academic experimentation, and future development of adaptive security monitoring frameworks.

I. INTRODUCTION

The rapid expansion of digital infrastructure has transformed how organizations communicate, operate, and manage critical information. However, this technological growth has simultaneously increased exposure to sophisticated cyber threats targeting authentication systems, operating services, and network communication channels. Modern attacks often involve multi-stage techniques such as brute-force attempts, credential theft, privilege escalation, and lateral movement, making them difficult to detect using traditional security controls alone.

Conventional defense mechanisms, including firewalls and antivirus software, primarily focus on perimeter protection and signature-based detection. While effective against known threats, these tools frequently lack the contextual awareness required to identify complex or coordinated intrusions. As a result, Security Operations Centers (SOCs) have become an essential component of modern cybersecurity strategy, providing centralized monitoring, real-time analysis, and structured incident response capabilities.

Enterprise-grade SOC platforms powered by Security Information and Event Management (SIEM) technologies—such as Splunk Enterprise Security, IBM QRadar, and ArcSight—offer advanced detection and analytics features. However, their high licensing costs, infrastructure requirements, and operational complexity often make them impractical for academic institutions and smaller organizations seeking to study SOC workflows or prototype defensive architectures.

To address this challenge, the present work implements a fully functional SOC-style environment using open-source technologies. The system integrates Wazuh SIEM for log collection, MongoDB for scalable storage, FastAPI for backend communication,

and Dash with Plotly for visualization. External threat intelligence feeds, including AlienVault OTX and AbuseIPDB, are incorporated to enrich local telemetry with reputation-based insights. Deployed on a native Ubuntu platform, the proposed architecture closely mirrors real-world SOC operations by enabling automated ingestion, event correlation, and dashboard-driven threat visibility. This approach contributes to cybersecurity education and research by demonstrating that effective security monitoring frameworks can be developed without reliance on costly commercial solutions.

II. LITERATURE SURVEY

1. Log Correlation for Intrusion Detection (2008)

- Lin et al. investigated the effectiveness of correlating security events collected from multiple system sources to identify coordinated cyberattacks. Their research demonstrated that analyzing authentication records alongside kernel and network logs enables security teams to reconstruct attacker behavior more accurately.
- The study highlighted that correlation-based detection significantly improves the identification of threats such as brute-force attempts and unauthorized privilege escalation.

2. Rule-Based Intrusion Detection Using OSSEC (2010)

- Guzman and associates conducted an evaluation of OSSEC as a host-based intrusion detection platform for monitoring system activities. Their findings showed that rule-driven analysis could successfully detect suspicious SSH access patterns and unauthorized administrative actions.
- The research established that open-source intrusion detection solutions are capable of delivering reliable performance comparable to certain commercial security platforms.

3. Adoption of Open-Source SIEM in Security Operations (2014)

- Ramos et al. analyzed the growing adoption of open-source SIEM technologies within enterprise security infrastructures. Tools such as OSSEC, Snort, and Bro were assessed based on scalability, alert management, and cross-platform visibility.
- The authors concluded that open-source ecosystems provide a practical alternative for organizations seeking cost-efficient security monitoring without sacrificing analytical capability.

4. Threat Intelligence Enrichment in SOC Environments (2015)

- Ahmed et al. explored the integration of external threat intelligence feeds into SOC workflows to enhance incident analysis. Their framework enriched raw log data with contextual attributes such as IP reputation and domain history, allowing analysts to make faster and more informed decisions.
- The study demonstrated that intelligence-driven enrichment reduces response time and strengthens investigative accuracy.

5. Visualization Techniques for Security Log Analysis (2016)

- Li and Xu proposed visualization strategies aimed at simplifying the interpretation of large-scale security datasets. By employing timeline graphs and statistical dashboards, their approach enabled analysts to identify anomalies and behavioral trends more efficiently.
- The research emphasized that visual analytics plays a critical role in accelerating incident investigation and improving situational awareness.

6. Open-Source Intelligence for Threat Attribution (2016)

- Kwon and colleagues examined the role of publicly available intelligence sources in attributing cyber threats. Their work focused on identifying malicious domains, phishing infrastructure, and abusive IP addresses through community-driven datasets.
- The authors concluded that combining OSINT with internal telemetry enhances the reliability of threat attribution processes.

7. Real-Time Monitoring Through SIEM Pipelines (2017)

- Alomar et al. developed a streaming-based SIEM architecture designed to support real-time event ingestion and alert generation.

- Leveraging distributed technologies, their model demonstrated the ability to process high volumes of security data while maintaining analytical responsiveness. This work reinforced the importance of scalable pipelines in modern SOC operations.

8. Automated Correlation Using Threat Intelligence APIs (2017)

- Hossain and team investigated the use of reputation-based APIs to classify suspicious network entities automatically. Their system correlated incoming logs with external intelligence sources to prioritize high-risk events.
- The findings indicated that automated enrichment improves alert prioritization while minimizing false positives.

9. Dashboard-Centric Security Monitoring (2019)

- Kim et al. introduced an analyst-focused dashboard designed to support security monitoring through interactive visual components. Cluster maps and temporal charts enabled faster recognition of attack patterns, thereby reducing cognitive workload for SOC personnel.
- The research confirmed that visualization-driven interfaces enhance operational efficiency during threat analysis.

10. Intelligent Alert Correlation in SOC Platforms (2020)

- Singh et al. proposed a machine-assisted correlation mechanism that groups related alerts based on behavioral similarity. Their approach reduced alert fatigue by filtering redundant notifications and presenting consolidated threat views.
- The study suggested that semi-automated correlation techniques can significantly improve SOC productivity.

11. FastAPI for Security Microservices (2020)

- Pereira et al. evaluated modern Python frameworks for building lightweight security services. FastAPI demonstrated strong performance in handling asynchronous requests and delivering structured JSON responses with minimal latency.
- The authors recommended microservice-based architectures to support modular and scalable SOC deployments.

12. Cyber Threat Intelligence Sharing Platforms (2021)

- Garba and collaborators analyzed collaborative intelligence-sharing ecosystems such as AlienVault OTX and MISP. Their research showed that shared threat indicators enable organizations to detect emerging attack campaigns more effectively.
- The paper emphasized that cooperative defense strategies are essential for countering globally distributed cyber threats.

13. Reputation-Based IP Analysis for Network Protection (2021)

- Huang examined the effectiveness of crowd-sourced reputation databases in identifying malicious IP addresses. By leveraging community-reported abuse data, the study successfully detected hosts involved in brute-force activities.
- The results supported the integration of reputation scoring mechanisms into automated defense frameworks.

14. SOC Automation Using Python-Based Pipelines (2022)

- Lopez et al. designed an automated pipeline that transforms raw security logs into actionable alerts using Python-driven workflows. Their implementation streamlined ingestion, enrichment, and correlation tasks while reducing manual workload.
- The research highlighted automation as a key factor in modernizing SOC infrastructure in a cost-effective manner.

15. AI-Driven SIEM for Multi-Stage Attack Detection (2023)

- Murthy et al. explored the application of artificial intelligence within SIEM platforms to detect complex attack sequences. Their model analyzed relationships across diverse log sources to uncover patterns associated with lateral movement and persistent threats.

- The study concluded that intelligence-enhanced SIEM architectures substantially strengthen organizational cyber defense.

III. PROPOSED METHODOLOGY

The methodology adopted in this project focuses on developing a realistic Security Operations Center (SOC) capable of monitoring, analyzing, and correlating security events in real time. Instead of relying on simulated data, the system captures actual operating system logs using Wazuh and enriches them with external threat intelligence to replicate practical SOC workflows. A native Ubuntu environment was selected to ensure system stability and compatibility with open-source security tools, while predefined detection rules help generate alerts whenever suspicious activity is identified.

All collected events are stored in MongoDB for centralized and structured data management, enabling efficient querying and investigation. A custom correlation engine compares logs with threat indicators to detect potential security violations and assign severity levels. Automation ensures continuous data ingestion and analysis, while the dashboard converts raw security data into visual insights, allowing analysts to quickly understand threat patterns and respond effectively.

3.1 Proposed System

The proposed SOC framework operates through five primary stages:

Step 1:
Log Collection - Wazuh agents gather system, authentication, and network logs, which are analyzed by the Wazuh Manager to detect abnormal activities.

Step 2:
Threat Intelligence Integration - Indicators of Compromise are retrieved from AlienVault OTX and AbuseIPDB to provide external context for evaluating suspicious events.

Step 3:
Event Correlation - A correlation engine compares log source IP addresses against threat indicators to identify potential malicious interactions.

Step 4:
Alert Generation and Storage - Detected threats are assigned severity levels and securely stored in MongoDB for historical tracking and forensic analysis.

Step 5:
Visualization and Monitoring - A Dash-based SOC dashboard displays alerts, attack patterns, and severity metrics, enabling real-time situational awareness.

3.2 Feature Extraction and Pre-processing

Feature extraction and pre-processing represent essential stages in the development of an effective Security Operations Center (SOC) architecture, as they convert raw security data into structured information suitable for analysis, correlation, and visualization. Logs generated from operating systems, authentication services, and security tools are typically heterogeneous and unstructured. Without proper refinement, such data can lead to inaccurate detection results and increased false positives. Therefore, a systematic preprocessing framework is implemented to enhance data quality and ensure reliable threat identification.

A. Data Sources

The proposed SOC system aggregates security data from multiple operational layers to provide comprehensive visibility into system activities. Primary data sources include authentication logs such as SSH and *auth.log*, system-level syslog records, and alerts generated by the Wazuh SIEM platform. Additionally, external threat intelligence feeds—including AlienVault OTX and AbuseIPDB—are integrated to supply contextual information regarding potentially malicious IP addresses and domains. These datasets contain critical attributes such as source IP, timestamps, rule descriptions, severity levels, and event categories, all of which contribute to effective security monitoring.

B. Data Preprocessing

Prior to analysis, the collected logs undergo several preprocessing steps designed to improve consistency and analytical accuracy.

Log Parsing:
Raw JSON alerts generated by Wazuh are parsed using Python-based ingestion scripts to extract meaningful security fields required for downstream processing.

Data Cleaning:
Incomplete records, corrupted entries, and null values are removed to maintain dataset reliability. Duplicate alerts are also filtered to prevent redundant storage and reduce analytical noise.

Normalization:
Since logs originate from diverse sources with varying formats, normalization converts them into a unified schema. This standardization enables efficient storage within MongoDB and supports seamless cross-source correlation.

Timestamp Standardization:
All event timestamps are transformed into a consistent format, allowing chronological ordering of security incidents and improving the accuracy of event reconstruction.

Deduplication:
Hash-based techniques are applied to identify repeated alerts before database insertion. This approach minimizes storage overhead while preventing analyst fatigue caused by excessive duplicate notifications.

C. Feature Extraction
Following preprocessing, the system extracts relevant attributes that support threat detection and behavioral analysis. Feature extraction focuses on identifying the most informative elements within each log entry to distinguish legitimate activity from suspicious behavior.

Key extracted features include:

Feature	Description
Source IP	Identifies the origin of the activity
Destination IP	Indicates the targeted system when available
Event Type	Specifies the nature of the activity (e.g., login failure, brute-force attempt)
Timestamp	Records the exact time of occurrence
Rule Description	Detection rule triggered by Wazuh
Severity Level	Represents the priority of the threat
IOC Match	Indicates whether the entity matches known malicious indicators

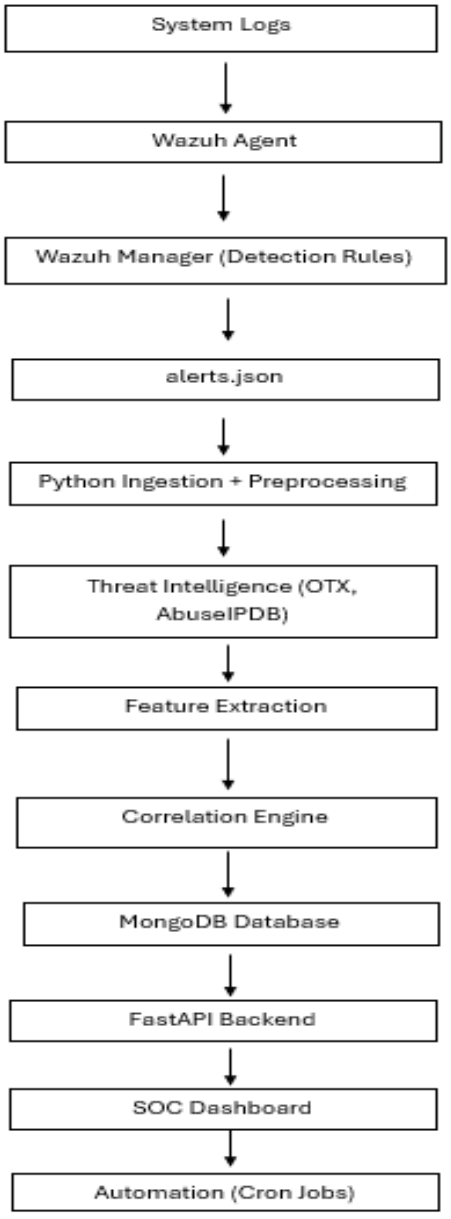
These features form the analytical foundation for the correlation engine and enable accurate classification of security events.

D. Threat Intelligence Feature Enrichment
To further strengthen detection capabilities, extracted features are enriched using external threat intelligence sources. Reputation scores, confidence values, and known malicious indicators are appended to relevant log entries. This enrichment process provides additional context that assists analysts in prioritizing high-risk events and responding more effectively to potential incidents.

E. Importance of Feature Extraction in SOC
Robust feature extraction significantly enhances SOC performance by enabling intelligent security analytics rather than simple log aggregation. Structured datasets improve correlation accuracy, accelerate threat detection, and reduce false positives. Moreover, refined data supports real-time monitoring and enhances dashboard visualization, allowing analysts to interpret attack patterns with greater clarity.
By transforming raw logs into meaningful security attributes, the proposed system establishes a strong analytical pipeline capable of supporting proactive defense strategies and informed incident response.

IV. METHODOLOGY

WORKFLOW



4.1 WORK FLOW EXPLANATION

1. Log Generation

The process begins with real-time log creation from the operating system, including:

- SSH login attempts
- Authentication failures
- System activity logs

These logs act as the primary security data source.

2. Log Collection – Wazuh Agent

The Wazuh Agent continuously monitors system log files and forwards them securely to the Wazuh Manager, enabling centralized monitoring.

3. Log Analysis – Wazuh Manager

The Wazuh Manager applies detection rules and decoders to identify suspicious activities such as brute-force attacks. When a threat is detected, structured alerts are generated and stored in:
/var/ossec/logs/alerts/alerts.json

4. Alert Ingestion and Preprocessing

Python scripts read the alerts file and perform preprocessing:

- Parse JSON data
- Remove duplicates
- Normalize fields
- Standardize timestamps

This converts raw alerts into structured data.

5. Threat Intelligence Integration

External feeds such as AlienVault OTX and AbuseIPDB provide known malicious IP addresses and indicators of compromise (IOCs). These indicators are stored in the database for comparison.

6. Feature Extraction

Important attributes are extracted from logs, including:

- Source IP
- Event type
- Timestamp
- Severity
- Rule description

These features enable efficient threat detection.

7. Correlation Engine (Core Component)

The correlation engine compares log features with threat intelligence data.

If a match occurs:

- The event is marked malicious
- Severity is assigned
- A security alert is generated

This step transforms raw logs into actionable intelligence.

8. Data Storage – MongoDB

The system maintains three main collections:

- logs → Parsed events
- iocs → Threat indicators
- alerts → Correlated threats

MongoDB ensures scalable and flexible storage.

9. Backend API Layer

FastAPI exposes endpoints such as:

/Logs

/iocs

/alerts

These APIs allow real-time retrieval of security data.

10. SOC Dashboard Visualization

The dashboard displays:

- Total alerts
- Malicious IP addresses
- Threat activity
- Event timelines

This helps analysts quickly understand the system's security posture.

11. Automation

Bash scripts and cron jobs automate:

- IOC ingestion
- Alert ingestion
- Correlation

This enables continuous SOC monitoring with minimal manual effort.

V. SOC DASHBOARD

The diagram illustrates the operational workflow of the proposed Security Operations Center system. Security logs generated from operating systems are collected by the Wazuh SIEM, where they are analyzed and filtered for suspicious activity. These events are then enriched using external threat intelligence feeds such as AlienVault OTX and AbuseIPDB to provide contextual information about potentially malicious sources.

A correlation engine processes the enriched data by matching system logs with known Indicators of Compromise (IOCs). When a match is identified, alerts are generated and stored in the MongoDB database. Finally, the SOC dashboard visualizes these alerts through graphs and charts, enabling faster threat analysis and supporting efficient incident response.

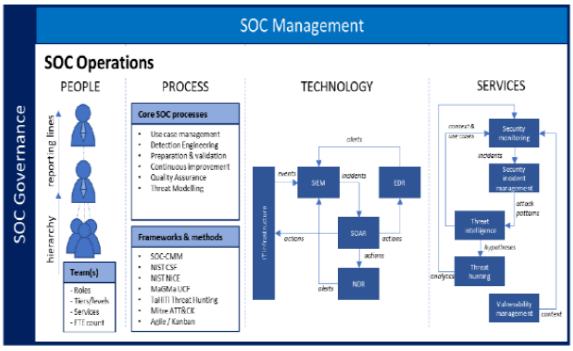


Fig 1. SOC DASHBOARD

5.1 HOW ITS WORKING

The proposed system demonstrates threat detection by simulating malicious behavior and observing how the Security Operations Center (SOC) identifies and processes such activities.

1.Simulating Malicious Activity

The system mimics suspicious behavior such as repeated SSH login attempts, unauthorized access trials, or connections from flagged IP addresses. These simulated activities act as controlled security events to validate the detection pipeline.

2.Triggering Detection Logic

Once the activity is generated, multiple detection mechanisms are activated:

• Signature-Based Detection:

Wazuh applies predefined rules to identify known attack patterns such as brute-force login attempts.

- Heuristic Analysis:

The system evaluates abnormal behavior including excessive login failures, unusual access times, or irregular system activity.

- Threat Intelligence Correlation:

IP addresses are compared against external threat intelligence feeds (e.g., AlienVault OTX, AbuseIPDB). If a match is found, the event is marked as high risk.

Thus, the system does not rely on artificial alerts; instead, it simulates realistic attack scenarios to demonstrate how modern SOC detection mechanisms respond.

Fig. 2: Threat Detection and SOC Alert Pipeline.

VI. THREAT ACTIVITY AND FORENSIC LOGS

Threat activity monitoring involves capturing, storing, and analyzing system events to identify potential security breaches. The SOC platform collects logs from multiple sources and transforms them into actionable intelligence.

Security monitoring tools record important attributes such as IP addresses, timestamps, event descriptions, and rule identifiers. These logs support forensic investigations by providing a traceable record of system behavior.

While monitoring tools can be used for legitimate administrative purposes such as auditing and troubleshooting, they are essential for detecting malicious activities including unauthorized access, credential misuse, and system compromise.

Types of Security Monitoring

Security monitoring mechanisms can be broadly classified into two categories:

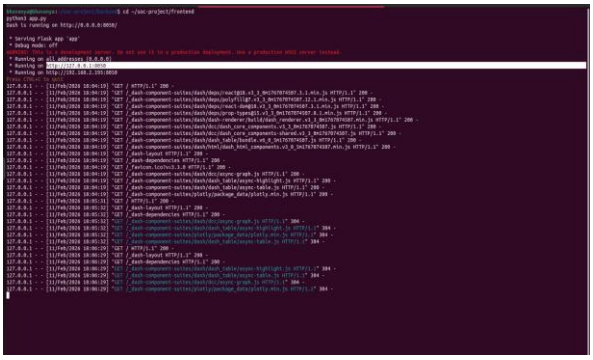
1. Host-Based Monitoring

- Implemented through software agents like Wazuh.
- Continuously observes system logs, authentication records, and file activity.
- Detects insider threats and unauthorized system changes.

2. Network-Based Monitoring

- Focuses on incoming and outgoing traffic.
- Identifies suspicious communication with malicious IP addresses.

Step 1: Starting the Frontend Dashboard

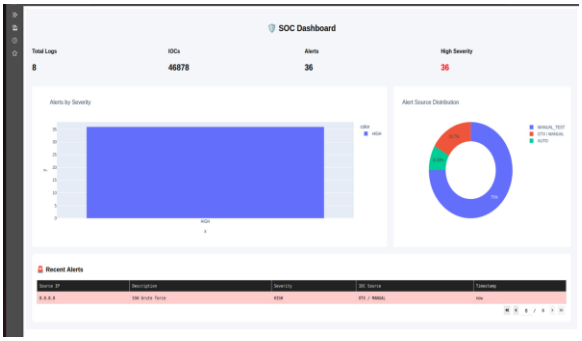


The terminal output shows that the frontend SOC dashboard has been successfully launched using the Dash framework. The application is running on port 8050, allowing users to access the dashboard through a web browser. This confirms that the visualization layer of the SOC system is active and ready to display real-time security data.

Step 2: Backend Server Initialization

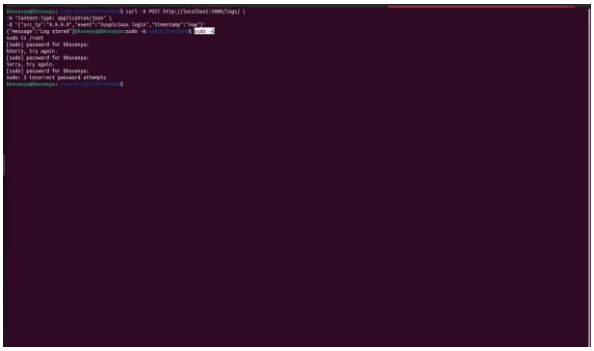
This screen confirms that the FastAPI backend server has started successfully using Uvicorn on port 5000. The server logs display multiple HTTP 200 OK responses, indicating that API endpoints such as /logs, /iocs, and /alerts are functioning correctly. This ensures proper communication between the database, detection engine, and frontend dashboard.

Step 3: SOC Dashboard Visualization



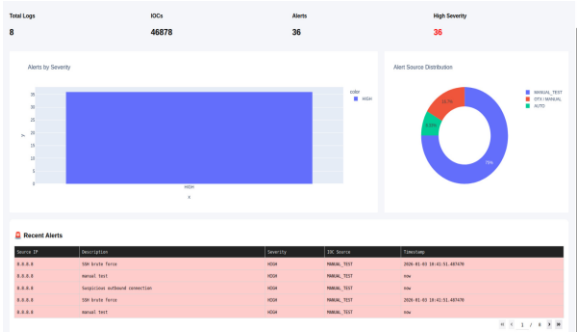
The figure illustrates the live SOC dashboard displaying key security metrics such as total logs, indicators of compromise (IOCs), alerts, and high-severity threats. Graphical representations help analysts quickly understand threat distribution, while the recent alerts table provides detailed information including source IP, severity level, and event description.

Step 4: Log Injection via API



The terminal demonstrates sending a POST request to the backend API to insert a security log into the system. Once the log is stored successfully, it becomes available for correlation and visualization on the dashboard. This step validates that the ingestion pipeline is operating correctly.

Step 5: Real-Time Alert Monitoring



This screen highlights the dashboard's ability to present correlated alerts in real time. Security events such as SSH brute-force attempts and suspicious connections are classified with high severity, enabling faster identification of potential threats and improving incident response.

VII. CONCLUSION

The proposed SOC Dashboard for Threat Intelligence and Log Correlation successfully demonstrates the implementation of a real-time security monitoring system using open-source technologies. By integrating Wazuh SIEM for log collection, external threat intelligence feeds for identifying malicious indicators, and a Python-based correlation engine, the system effectively detects and analyzes potential security threats.

The centralized storage of logs, indicators of compromise, and alerts in MongoDB, combined with FastAPI and a Dash-based visualization dashboard, enables efficient monitoring and faster incident identification. Automation further enhances the system by ensuring continuous data ingestion and analysis without manual intervention.

Overall, this project proves that a scalable and cost-effective Security Operations Center can be developed without relying on commercial tools. The architecture not only supports academic research but also provides practical exposure to real-world SOC operations, preparing users for modern cybersecurity environments.

REFERENCES

[1] AbuseIPDB, "IP Address Abuse Reporting Platform,"
Available: https://www.abuseipdb.com

[2] AlienVault Open Threat Exchange (OTX), "Open Threat Intelligence Community,"
Available: https://otx.alienvault.com

[3] Barnum, S., "Standardizing Cyber Threat Intelligence Information with the Structured Threat Information Expression (STIX)," MITRE Corporation, 2012.

[4] Behl, A., & Behl, K., *Cybersecurity and Cyberwar: What Everyone Needs to Know*, Oxford University Press, 2017.

[5] Dash Documentation, "Dash User Guide,"
Available: https://dash.plotly.com

[6] FastAPI, "FastAPI Framework Documentation,"
Available: https://fastapi.tiangolo.com

[7] Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E., "Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges," *Computers & Security*, vol. 28, no. 1–2, pp. 18–28, 2009.

[8] MongoDB Inc., "MongoDB Documentation,"
Available: https://www.mongodb.com/docs

[9] NIST, “Guide to Computer Security Log Management,” Special Publication 800-92, National Institute of Standards and Technology, 2006.
[10] Offensive Security, “Kali Linux Documentation,” Available: https://www.kali.org/docs
[11] Open Web Application Security Project (OWASP), “Top 10 Web Application Security Risks,” Available: https://owasp.org
[12] Plotly Technologies Inc., “Plotly Python Graphing Library,” Available: https://plotly.com
[13] Sabahi, F., & Movaghar, A., “Intrusion Detection: A Survey,” *International Conference on Systems and Networks Communications*, 2008.
[14] Scarfone, K., Souppaya, M., & Cody, A., “Guide to Intrusion Detection and Prevention Systems (IDPS),” NIST Special Publication 800-94, 2007.
[15] Shackleford, D., “Security Information and Event Management (SIEM) Implementation,” SANS Institute, 2012.
[16] The MITRE Corporation, “MITRE ATT&CK Framework,” Available: https://attack.mitre.org
[17] Uvicorn, “ASGI Server Documentation,” Available: https://www.uvicorn.org
[18] Wazuh Inc., “Wazuh Documentation – Open-Source Security Platform,” Available: https://documentation.wazuh.com
[19] Wagner, C., Dulaunoy, A., Wagener, G., & Iklody, A., “MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform,” *ACM Workshop on Information Sharing and Collaborative Security*, 2016.
[20] Zuech, R., Khoshgoftaar, T., & Wald, R., “Intrusion Detection and Big Data: A Survey,” *Journal of Big Data*, vol. 2, no. 1, 2015.