International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Incidence Response Infrastructure: A Compact Deployable SIEM for Academic and Small-to-Medium Organizations

Prateek Verma
Department of CSE (Cyber Security)
Acropolis Institute of Technology and Research
Prateekverma221154@acropolis.in

Vedant Shrivastava
Department of CSE (Cyber Security)
Vedantshrivastava22055@acropolis.in

Mrunal Hingonekar
Department of CSE(Cyber Security)
Murnalhingonikar22124@acropolis.in

Prof. Ashish Anjana
Department of CSE(Cyber Security),AITR
ashishanjana@acropolis.i n

Prof. Satyam Shrivastava
Department of CSE(Cyber Secuity),AITR
satyamshrivastava@acro plos.in

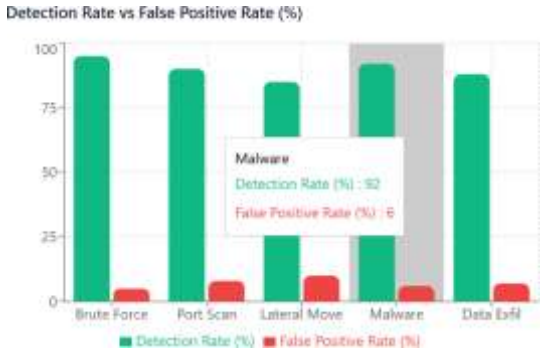
Abstract

This project presents the design and implementation of a lightweight, open-source Security Information and Event Management (SIEM) platform tailored for small and medium-sized enterprises (SMEs) and academic environments. The system aims to provide an affordable, deployable, and educational solution for centralized log management, anomaly detection, and automated security response. It aggregates and normalizes logs from multiple sources—including hosts, network sensors, and cloud services—enabling comprehensive visibility across the infrastructure. Through real-time telemetry, correlation rules, and intelligent alerting mechanisms, the platform supports effective threat detection and incident response. An intuitive dashboard offers visualization and analysis capabilities aligned with Security Operations Center (SOC) workflows, allowing users to investigate alerts and track attack patterns efficiently. Emphasizing scalability, log retention, and the reduction of false positives, the platform demonstrates that open-source technologies can achieve functional equivalence with commercial SIEMs for their target use case. Ultimately, the system serves as both a **practical educational framework** for cybersecurity students and a **cost-effective defensive tool** for smaller organizations, promoting accessibility, transparency, and hands-on understanding of modern security monitoring challenges.

I. INTRODUCTION

This project develops a compact and easily deployable **Security Information and Event Management (SIEM)** solution tailored for academic environments and small-to-medium organizations. It aggregates and centralizes **logs** from endpoints, network sensors, and cloud services, performing parsing, normalization, and correlation to identify potential security incidents. The system visualizes alerts and patterns through **interactive dashboards**, supporting effective monitoring and investigation. Designed with simplicity and extensibility in mind, it demonstrates key SIEM capabilities such as log management, anomaly detection, and automated responses. The platform also provides an

educational environment for students to understand real-world SOC operations, including challenges like scaling, data retention, and handling false positives, while remaining resource-efficient and open-source.



II. LITERATURE REVIEW

Security Information and Event Management (SIEM) systems are central to modern cybersecurity operations, as they enable organizations to collect, correlate, and analyze logs from multiple sources to detect, investigate, and respond to security incidents. According to *NIST SP 800-92*, effective log management requires documented procedures for secure collection, retention, and correlation to support incident handling. These foundational principles continue to guide the design and evaluation of SIEM-based monitoring and analysis workflows.

Several SIEM platforms—both open-source and commercial—have been developed to address the growing need for unified threat detection and response. Prominent examples include **Splunk**, **Wazuh**, **ELK Stack**, and **Graylog**, each offering unique strengths and limitations.

Splunk is one of the most widely used enterprise-grade SIEM platforms, known for its robust data analytics, advanced visualization, and scalable search capabilities. It supports large-scale deployments and integrates machine learning for predictive threat detection. However, its proprietary nature and high licensing costs make it less suitable for academic or small-to-medium business environments seeking affordable solutions.

Wazuh, on the other hand, is an open-source Extended Detection and Response (XDR) and SIEM platform that provides endpoint monitoring, intrusion detection, and compliance management. As outlined in *Wazuh Documentation (2025)*, it integrates with the **ELK Stack**—comprising Elasticsearch, Logstash, and Kibana—to enable centralized log analysis and visualization. Wazuh's modular rule-based engine and multi-platform support make it ideal for lightweight deployments, but its configuration complexity may present challenges for beginners.

The **ELK Stack (Elastic Stack)** itself serves as a core foundation for many modern SIEM solutions. *Elastic (2025)* describes it as a scalable framework for collecting, indexing, and visualizing logs from diverse sources in real time. Its flexibility and high performance have made it popular across both enterprise and research domains. However, building a complete SIEM system on ELK requires additional tools and customization for correlation and alerting, which can increase deployment overhead.

Graylog offers a more user-friendly approach to centralized log management and SIEM. As per *Graylog Documentation*

(2025), it focuses on simplicity, scalability, and efficient search capabilities. It provides modular integrations and dashboards for incident visualization, making it suitable for educational or mid-sized operational setups. Nonetheless, its detection and automation features are comparatively limited when measured against enterprise SIEMs like Splunk or fully featured platforms like Security Onion.

Beyond these, frameworks such as **OSSIM (Open-Source Security Information Management)** and **Security Onion** provide comprehensive monitoring environments by combining intrusion detection, vulnerability management, and asset discovery. However, both solutions are relatively resource-intensive and less optimized for lightweight or educational use. Similarly, *SIEMonster Community Edition (2025)* offers an open-source alternative with full-stack capabilities but demands complex configuration and maintenance.

While these existing systems demonstrate the evolution and diversity of SIEM solutions, they often involve trade-offs between capability, cost, and complexity. Most commercial SIEMs prioritize extensive feature sets suited for large enterprises but remain cost-prohibitive for smaller entities. Conversely, open-source tools provide flexibility but can require significant technical expertise for deployment and maintenance.

This gap—between affordability, simplicity, and deployability—forms the motivation for the current project. The proposed **lightweight SIEM platform** aims to bridge this divide by delivering essential detection, alerting, and response capabilities without the overhead of enterprise-level systems. It is designed specifically for academic labs and small-to-medium organizations, providing a cost-effective and easily deployable solution that adheres to NIST's foundational SIEM principles. By integrating open-source components like Wazuh, Suricata, and ELK within a streamlined architecture, this project offers a practical framework for both learning and real-world application, promoting accessibility and efficiency in modern security monitoring.

III. SYSTEM DESIGN AND METHODOLOGY

The system design follows a clear, layered pipeline from raw telemetry to actionable alerts.

1. Data sources (collection points). Logs originate from endpoints (Windows Event Logs, Linux syslog), network sensors (Suricata, Zeek), honeypots (Cowrie), and cloud service logs (e.g., AWS CloudTrail). Each source produces different record types—authentication events, process creation, network flows, and API calls—that provide complementary visibility across the kill chain.

2. Log collection layer. Lightweight agents (Filebeat on Linux, Winlogbeat on Windows) or syslog forwarding are deployed to stream events to the central ingestion point. Agents perform local buffering, basic filtering, and TLS-secured transport to prevent data loss and protect log confidentiality over the network.

3. Ingestion & normalization (Logstash). Logstash pipelines receive raw messages

and apply parsing plugins (grok, dissect, date) to extract structured fields (timestamp, source IP, user, event type). Normalization maps disparate schemas into a unified event model so downstream analytics can compare apples-to-apples across hosts and sensors.

4. Storage and indexing (Elasticsearch). Parsed events are indexed into Elasticsearch with time-based indices and rollover policies to enable fast search and scalable storage. Index templates enforce mappings for efficient queries and aggregations.

5. Correlation & detection. Wazuh and custom correlation rules consume indexed events, joining related alerts (e.g., failed SSH attempts + subsequent suspicious process creation) to raise higher-confidence incidents. Suricata/Zeek alerts enrich network context.

6. Visualization & response (Kibana). Kibana dashboards visualize trends, timelines, and top indicators; alert panels feed SOC workflows. Role-based access, retention policies, and archival to cold storage round out the design for security, compliance, and scalability.

III.I Tools and Technologies

1. **ELK Stack (Elasticsearch, Logstash Kibana):**
   The ELK Stack is used for log aggregation, indexing, and visualization. Elasticsearch enables efficient searching and analytics, Logstash handles data ingestion and transformation, and Kibana provides real-time dashboards for monitoring and alerting.
2. **Wazuh:**
   Wazuh is employed for endpoint monitoring, rule-based detection, and file integrity checking. It integrates with the ELK Stack to correlate host-based events with network-level data, enhancing threat visibility.
3. **Suricata/Zeek:**
   These tools provide modular and scalable network traffic analysis and intrusion detection capabilities. They capture and parse network packets to identify suspicious behaviours and anomalies.
4. **Docker and Ansible:**
   Docker is used to containerize components for simplified deployment and scalability, while Ansible automates configuration and orchestration across environments.
5. **Python and Bash Scripting:** These scripting languages are used to automate log collection, event correlation, and alert handling, ensuring flexibility and customization across diverse infrastructures.

IV. IMPLEMENTATION AND SETUP

The prototype was implemented in a controlled virtualized lab environment designed to emulate real-world attack and defense scenarios while ensuring system isolation and safety. The infrastructure was hosted on **Oracle VirtualBox**, consisting of multiple interconnected virtual machines, each serving a distinct function within the

cyber deception and monitoring framework.

The core of the setup comprised **one server VM (8 GB RAM, Ubuntu 22.04)** running the **ELK Stack (Elasticsearch, Logstash, Kibana)** integrated with **Wazuh** for centralized log management, correlation, and alert visualization. The SIEM server acted as the central point for collecting telemetry data from both host and network sources. **Filebeat and Syslog** were configured to forward logs from endpoints and honeypots to Wazuh for real-time monitoring.

Two additional **client systems** were deployed to simulate normal user and attack traffic. One acted as a **victim machine** (running Ubuntu/Windows Server) hosting intentionally vulnerable services, while the second was configured as an **attacker machine** using **Kali Linux**. The attacker machine was equipped with tools such as **Nmap**, **Hydra**, and **Metasploit** to perform controlled penetration attempts, vulnerability scans, and brute-force attacks.

A **honeypot (Cowrie)** was integrated into the environment to mimic a vulnerable SSH/Telnet service and capture malicious login attempts, executed commands, and file transfers. Logs generated by Cowrie were continuously forwarded to the Wazuh server, where correlation rules and detection policies analyzed patterns to generate relevant security alerts. **Suricata** was additionally deployed to monitor network traffic and detect port scans or anomalous packets.

Use Cases Tested:

1. **Brute-Force / Credential-Guessing Attacks**

What the attack is (conceptual): An attacker attempts many different authentication attempts against a service (e.g., SSH) to guess valid credentials. This can be targeted (trying a few accounts with many passwords) or high-volume (credential stuffing using leaked username/password lists).

Artifacts produced:

- Repeated failed authentication entries in auth logs (timestamps, source IP, username attempted).
- Occasional successful login events (if compromise happens).
- Higher-than-normal connection attempts from the same IP or range.
- Honeypot-specific logs capturing attacker commands, typed credentials, and session transcripts.

How it's monitored / detected in your stack:

- Cowrie (honeypot) records interactive sessions and failed/successful logins. Those logs forwarded to Wazuh provide rich context (attempted username/password, session transcript).
- Wazuh parses auth logs and applies correlation rules: e.g., X failed attempts from same IP within Y minutes → generate an alert.
- Elasticsearch/Kibana dashboards show time-series charts of failed attempts by source IP, heatmaps of

top usernames, and geo-location of sources.

- Correlation: The SIEM correlates repeated auth failures with suspicious post-login activity (e.g., immediate privilege escalation attempts) to raise severity.

Detection signals to tune:

- Threshold of failed attempts within a short window.
- Multiple usernames tried from single IP.
- Use of known compromised credential lists (if integrated).
- High rate of connection attempts to authentication ports.

Analyst actions / forensics:

- Inspect Cowrie session logs for commands and payloads.
- Block or rate-limit offending IPs at the firewall.
- Check for any successful logins; if found, isolate the host and collect memory/ disk artifacts.
- Add blacklist/IOC and tune rules to reduce false positives (e.g., exclude legitimate scanning activity from internal testing hosts).

2) Port Scanning & Reconnaissance

What the attack is (conceptual): An attacker probes a target to discover open ports, running services, and versions. This helps them identify possible vulnerabilities and plan follow-on attacks.

Artifacts produced:

- Numerous short-lived TCP SYN / UDP probe packets across a range of ports.
- Repeated connection attempts with unusual timing or patterns.
- Service-banner captures (if banner grabbing is performed).

How it's monitored / detected in your stack:

- Suricata flags scanning patterns (e.g., high rate of SYN to many ports, sequential port sweeps) and produces alerts describing the scan type.
- Wazuh ingests Suricata alerts and correlates them with host logs (e.g., connection attempts logged by the system).
- Kibana dashboards display scan timelines, IPs initiating scans, and the list of targeted ports/services.

Detection signals to tune:

- Excessive distinct destination ports within a small time window.
- Repeated probes for same service types across multiple targets.
- Scans originating from often-malicious IP ranges or proxies.

Analyst actions / forensics:

- Map scan source to known threat intelligence lists; block if malicious.
- Use packet captures recorded by Suricata for deeper analysis (payloads, banners).

- Harden services identified as frequently probed (disable unused ports, change defaults).

3) Privilege Escalation Attempts

What the attack is (conceptual): After initial access, adversaries attempt to elevate their privileges to gain broader control (e.g., using misconfigured sudo, vulnerable binaries, or privilege-escalation exploits).

Artifacts produced:

- Suspicious use of privilege escalation utilities (sudo attempts, setuid binary execution).
- New or unusual process spawn chains (e.g., shell spawned by a system process).
- Changes to sensitive files, or the presence of tools/binaries not normally installed.

How it's monitored / detected in your stack:

- Wazuh agents on hosts monitor system logs, sudo logs, process creation events, and file integrity changes.
- Predefined Wazuh rules trigger on patterns like failed then successful sudo attempts, execution of uncommon binaries, or modification of privileged configuration files.
- Dashboards highlight process lineage and the user accounts involved.

Detection signals to tune:

- A user executing commands outside their normal profile.
- Execution of tools from temporary directories or non-standard locations.
- Sudden changes to privileged groups or files.

Analyst actions / forensics:

- Capture process trees and collect relevant log slices.
- Check local user accounts and scheduled tasks for persistence.
- If escalation is confirmed, isolate host and perform deeper forensic analysis (logs, filesystem).

4) Malware / Malicious Process Execution

What the attack is (conceptual): Execution of malicious code that may spawn reverse shells, beacon out to command-and-control, modify files, or move laterally.

Artifacts produced:

- New or anomalous processes, unexpected network connections (beaconing), file changes (dropped payloads).
- Unusual process parent/child relationships and abnormal process arguments.
- Outbound connections to rare external endpoints.

How it's monitored / detected in your stack:

- Wazuh collects host telemetry (process, file integrity, syscalls) and

raises alerts on anomalous process behavior.

- Suricata detects suspicious network flows (persistent small periodic connections) and flags IOC matches.
- Kibana visualizes combined host+network activity to show both the malicious process and its external communications.

Detection signals to tune:

- Repeated small outbound connections (beacons).
- Processes launched from user temp or unusual directories.
- Changes to autorun entries or scheduled tasks.

Analyst actions / forensics:

- Quarantine the infected host; capture process memory if possible.
- Extract and submit suspected binaries for analysis.
- Block C2 domains/IPs and search historical logs for earlier beacons.

5) Data Exfiltration & Insider Misuse

What the attack is (conceptual): An attacker or malicious insider attempts to move sensitive data off the network—via SCP, large HTTP POSTs, cloud uploads, or covert channels.

Artifacts produced:

- Unusually large outbound transfers or spikes in data volume from a host.
- Use of uncommon protocols or endpoints for data transfer.
- Access to sensitive files shortly before the transfer.

How it's monitored / detected in your stack:

- Network sensors (Suricata) detect large or unusual outbound flows and raise volume-based alerts.
- Wazuh flags access to sensitive directories and correlates user file accesses with outbound network activity.
- Dashboards display data transfer volumes per host and highlight anomalous spikes.

Detection signals to tune:

- Data transfer volumes outside normal baselines for a user/host.
- Use of nonstandard ports or external endpoints for transfers.
- Correlation of file access and outbound connection within a short interval.

Analyst actions / forensics:

- Immediately block the destination and isolate the host.
- Identify exactly which files were accessed and transferred.
- Notify data owners and follow incident response playbooks; preserve logs for legal/compliance purposes.

This controlled deployment successfully demonstrated end-to-end data flow—from event generation to centralized analysis and

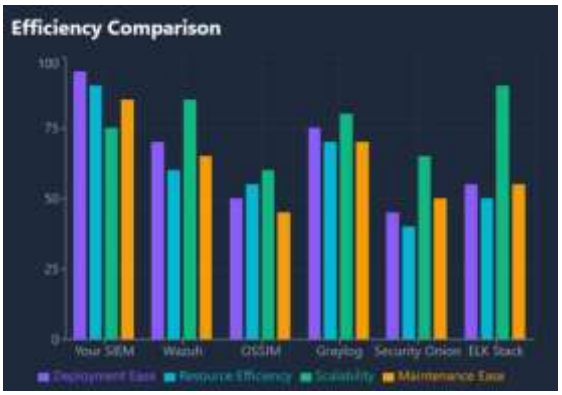
alerting—validating that the lightweight SIEM-based cyber deception setup can effectively detect, visualize, and correlate attacker behaviour within a safe and reproducible lab environment

V. DISCURSION

The project validates that a simplified and lightweight SIEM infrastructure can effectively deliver meaningful detection, monitoring, and response capabilities without the complexity or cost of enterprise-grade systems. Its modular and compact design enables easy deployment, making it highly suitable for academic institutions and small-to-medium businesses seeking practical security visibility. The platform demonstrates how open-source tools can be integrated to create a unified environment for log management, anomaly detection, and alert correlation. This not only strengthens real-time threat awareness but also offers an accessible hands-on learning framework for cybersecurity education. The balance between simplicity, scalability, and functionality ensures adaptability across diverse operational contexts. Future research can extend this system by incorporating machine learning-based anomaly detection, cloud-native scalability, and automated incident response playbooks to further enhance detection accuracy, resilience, and operational efficiency.

VI. CONCLUSION

This research demonstrates that a cost-effective and deployable SIEM platform can significantly enhance both cybersecurity education and small-scale organizational defense. By integrating open-source tools into a unified framework for log collection, analysis, and automated response, the system achieves functional equivalence to larger commercial SIEMs within its defined scope. It provides students with hands-on exposure to Security Operations Center (SOC) workflows, incident management, and real-world challenges such as alert tuning, false positives, and scalability.



In the future, the system can be expanded to include advanced machine learning models for predictive threat detection, integration with cloud-native and containerized environments, and improved visualization dashboards. Features such as adaptive correlation rules, threat intelligence feeds, and distributed data storage could further enhance performance and reliability. Additionally, incorporating automated playbooks and response orchestration would transform the platform into a more mature SOAR (Security Orchestration, Automation, and Response) solution. These enhancements will make the system more robust, scalable, and suitable for real-world enterprise deployments.

VII. REFERENCE

1. Wazuh Documentation, "Getting started with Wazuh," Wazuh, 2025. [Online]. Available: https://documentation.wazuh.com/current/getting-started/index.html
2. Wazuh, "Wazuh — Open Source XDR and SIEM Platform," 2025. [Online]. Available: https://wazuh.com/
3. AT&T Cybersecurity, "Open Source Security Information Management (OSSIM) — Overview," 2004. [Online]. Available: https://cdn-cybersecurity.att.com/docs/OSSIM_overview.pdf
4. InfoSec Institute, "AlienVault OSSIM Review — Open Source SIEM," Apr. 26, 2012. [Online]. Available: https://www.infosecinstitute.com/resources/network-security-101/alienvault-ossim-review-open-source-siem/
5. Graylog, "Graylog — Centralized Log Management and SIEM," Graylog, 2025. [Online]. Available: https://graylog.org/
6. Graylog, "Graylog Source-Available and Open Modules," Graylog, 2025. [Online]. Available: https://graylog.org/products/source-available/
7. Security Onion Solutions, "Security Onion Documentation," 2025. [Online]. Available: https://docs.securityonion.net/
8. Elastic, "Elastic Stack (ELK) — Elasticsearch, Kibana & Logstash," Elastic, 2025. [Online]. Available: https://www.elastic.co/elastic-stack
9. SIEMonster, "SIEMonster Community Edition — Open Source SIEM Platform," 2025. [Online]. Available: https://siemonster.com/