ADROID Conference Series: Engineering and Technology
Volume 2, Issue 3, pp.272-281, 2026

Honeypot-Based Intrusion Detection with SIEM and Automated Response

Harudhan Kapoor[1], Aditya Turankar[2], Mukesh Kawatghare[3]

[1,2]G H Raisoni College of Engineering, Nagpur, India

[3]Independent Researcher, Nagpur, India

harudhan.kapoor.cse@ghrce.raisoni.net, aditya.turankar@raisoni.net, parthkawatghare@gmail.com

Abstract. Cybercrime is on the rise due to an increase in cyberattacks such as brute-force attacks, network scanning, and malware deployment, leading to a greater need for network security monitoring. Traditional systems rely heavily on signatures and alerts and therefore often do not provide valuable insight into the attacker's behavior. Implementing deception-based security with honeypots enables security professionals to collect data on attacker activity. This paper presents an architecture for deception-based intrusion detection using a honeypot, a Network Intrusion Detection System (NIDS), centralized logging, Security Information and Event Management (SIEM), threat intelligence enrichment, and automated response mechanisms. Using the Cowrie honeypot, Suricata NIDS, syslog, and Wazuh SIEM, the architecture communicates attacker information to one centralized logging repository. A distinct feature is an attacker profiling module that classifies attacker behavior by executed commands and automatically blocks malicious IP addresses. The system uses multiple open-source tools to simulate a small-scale Security Operations Centre (SOC).

Keywords: Honeypot, IDS, SIEM, Wazuh, Suricata, Cowrie, Deception Security, Threat Intelligence, SOC, Log Correlation.

1. INTRODUCTION

The emergence of cyber attacks has significantly altered the landscape of information security, particularly automated network scanning, brute force attacks, and malware distribution. Organizations use intrusion detection systems (IDS), firewalls, and monitoring applications to detect attacks, but these tools produce alerts without insight into attacker behavior [16].

Deception-based security via honeypots creates decoy systems to attract attackers [17]. Integrating honeypots with IDS and SIEM platforms enables correlation of attacker activity across multiple monitoring layers [8]. This paper presents an architecture integrating Cowrie, Suricata, Wazuh SIEM, threat intelligence, and automated response to detect, analyze, and respond to attacks automatically.

2. RELATED WORK

Prior work in honeypot-based intrusion detection addresses individual components but rarely combines all layers into a unified pipeline.

A. Honeypot-Based Deception

Morić et al. [8] compared seven honeypot solutions but implemented no SIEM integration, NIDS, attacker profiling, or automated response. Sikdar et al. [10] proposed a web-only honeypot hierarchy with no SIEM or network IDS. Javadpour et al. [11] surveyed deception techniques theoretically with no concrete implementation [13].

B. Honeypot + SIEM Integration

Yusof et al. [14] deployed Cowrie with a SIEM for educational purposes but lacked Suricata, threat intelligence, and automated response. Alrashdan et al. [12] used Artillery honeypot with Wazuh for anomaly detection but included no attacker profiling or IP blocking.

C. SIEM + NIDS Integration

Pratama et al. [7] combined Wazuh and Suricata with Telegram notifications but deployed no honeypot and no threat intelligence lookup. Hindy et al. [15] reviewed NIDS trends, noting most solutions lack deception mechanisms or behavioral profiling.

D. Automated Response

Harani et al. [9] achieved sub-second blocking using Cowrie with Azure Sentinel, but the architecture is cloud-specific and proprietary, lacking Suricata or AbuseIPDB. Kapatel et al. [6] applied ML classification with Cowrie but required large training datasets and included no SIEM or real-time IP blocking.

E. Research Gap

Table I compares related works. No existing work unifies all seven components—Cowrie, Suricata, Wazuh, attacker profiling, AbuseIPDB, automated IP blocking, and open-source on-premise deployment—into a single pipeline. The proposed system is the first to do so.

F. Gaps and Opportunities

- Semantic and contextual comprehensiveness are lacking in keyword-based search systems.
- Ontology-based retrieval is less flexible even though it improves accuracy.
- RAG in biomedicine serves as inspiration, despite the lack of Ayurvedic-specific systems.
- Domain-specific LLM adaptation has potential, but requires ontology alignment and carefully selected multilingual corpora.

These limitations support the methodology of AyuRAG, a lightweight, RAG-based system designed for accurate, intelligible, and contextually sensitive access to Ayurvedic knowledge.

3. PROPOSED SYSTEM ARCHITECTURE

The proposed system (Fig. 1) integrates a deception layer, network detection layer, log collection layer, SIEM correlation layer, attacker profiling layer, threat intelligence layer, and automated response layer. The **Deception Layer** uses Cowrie [3] for SSH/Telnet session capture. The **Network Detection Layer** uses Suricata [2] for traffic monitoring. Logs are forwarded via **syslog** to Wazuh [1] for correlation and alerting. AbuseIPDB [4] verifies IP reputation; malicious IPs are automatically blocked via firewall rules.

Fig. 2 illustrates the log correlation flow. The Wazuh SIEM dashboard (Fig. 3) provides a centralized view of all security events, combining evidence from Cowrie and Suricata with 28,893 low-severity, 12 medium-severity, and 1 high-severity alerts recorded during the experiment. MITRE ATT&CK mapping and Vulnerability Detection modules are also active.

TABLE I
COMPARATIVE ANALYSIS OF RELATED WORKS VS. PROPOSED ARCHITECTURE.

Reference	Year	Cowrie Honeypot	Suricata NIDS	Wazuh SIEM	Attacker Profiling	Threat Intel.	IP Block	OSS On-Prem
Morić et al. [8]	2025	✓	×	×	×	×	×	×
Pratama et al. [7]	2025	×	✓	✓	×	×	×	Partial
Kapatel et al. [6]	2025	✓	Partial	×	Partial	×	×	Partial
Azure+Sentinel [9]	2025	✓	×	×	Partial	×	✓	×
Javadpour et al. [11]	2024	×	×	×	×	×	×	×
Sikdar et al. [10]	2024	×	×	×	Partial	×	×	×
Artillery+Wazuh [12]	2024	×	×	✓	×	×	×	Partial
Balogh et al. [13]	2024	✓	×	×	×	×	×	×
UiTM Study [14]	2022	✓	×	×	×	×	×	Partial
Hindy et al. [15]	2021	×	✓	×	×	×	×	×
Proposed	**2026**	✓	✓	✓	✓	✓	✓	✓

Legend

- ✓ = Fully Implemented
- **Partial** = Partially Implemented
- × = Not Implemented
- **OSS** = Open-Source
- **Threat Intel.** = AbuseIPDB Reputation Lookup
- **IP Block** = Automated Firewall Blocking

4. IMPLEMENTATION

The system runs on two virtual machines. Cowrie is deployed on Ubuntu Server capturing failed login logs and executed commands. Suricata monitors traffic for port scans and brute force attempts. Syslog forwards logs to Wazuh Manager where detection rules generate dashboard alerts. A Python profiling script (Fig. 4) extracts attacker IPs, classifies commands, queries AbuseIPDB [4], forwards profiles to Wazuh, and blocks malicious IPs via firewall rules.

5. ATTACK WORKFLOW

Nmap scans from the attacker machine trigger Suricata alerts. SSH brute force follows, captured by Cowrie (Fig. 5). The profiling module classifies attack types; AbuseIPDB verifies the IP; Wazuh blocks it and generates correlated alerts.

6. RESULTS AND ANALYSIS

A. Cowrie Log Capture

Fig. 6 shows 30 high-severity Cowrie hits (rule level 14) over 8 days. Attacker commands included **ls**, **sudo etc/passwd**, **git**, and **curl** from IP **192.168.1.16**, indicating reconnaissance and privilege escalation.

TABLE II
SUMMARY OF EXPERIMENTAL DETECTION RESULTS

Metric	Value
Total Suricata alerts (24 h)	27,567
Cowrie high-severity alerts	30
Wazuh low-severity alerts	28,893
Wazuh medium-severity alerts	12
Wazuh high-severity alerts	1
AbuseIPDB score (attacker IP)	90/100
IPs automatically blocked	2
Attack types detected	10+

B. Suricata Network Detection

Figs. 7–8 show 27,567 Suricata alerts in 24 hours: SYN floods (2,138 hits), ICMP sweeps, SSH brute force, FTP/Telnet probes. Primary attacker IP: 192.168.1.16.

C. Attacker Profiling and Threat Intelligence

Fig. 9 shows both 10.121.51.181 and 10.121.51.192 receiving AbuseIPDB score 90, severity HIGH, and flagged for blocking.

D. Automated IP Blocking

Fig. 10 confirms iptables DROP rules applied to both IPs, completing the full detection-to-response cycle without human intervention.

E. Summary

Table II and Fig. 11 summarize all quantitative results. The full pipeline from attacker activity to automated blocking was validated successfully.

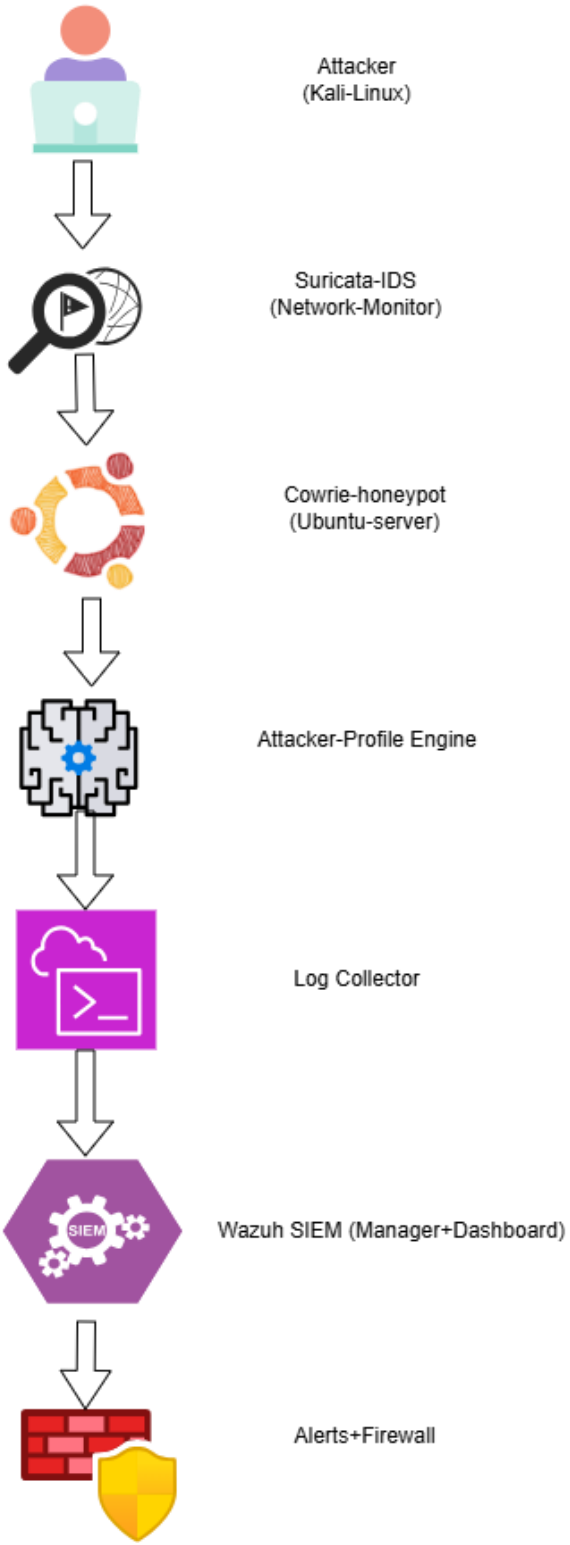

Fig. 1. The proposed system architecture illustrates the end-to-end intrusion detection and response pipeline. An attacker interacts with the Cowrie honeypot while Suricata simultaneously monitors network traffic. Security events are processed by the attacker profiling engine and centralized log collector, then forwarded to Wazuh SIEM for correlation. Automated firewall rules are subsequently applied to block identified malicious IP addresses.

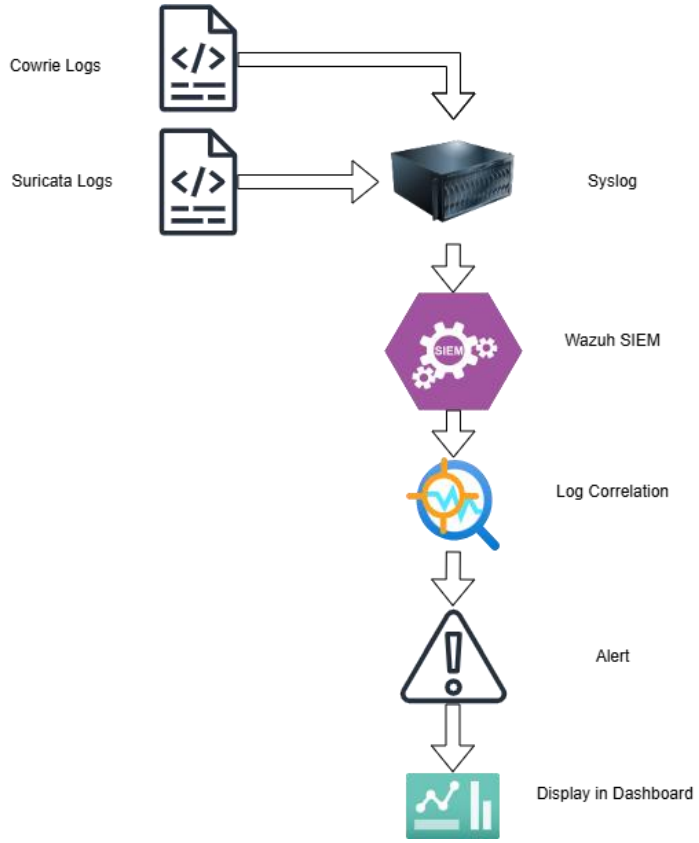

Fig.2. Log correlation flow illustrating how Cowrie and Suricata events are forwarded via Syslog to Wazuh SIEM, where rule-based correlation generates actionable security alerts displayed on the monitoring dashboard.

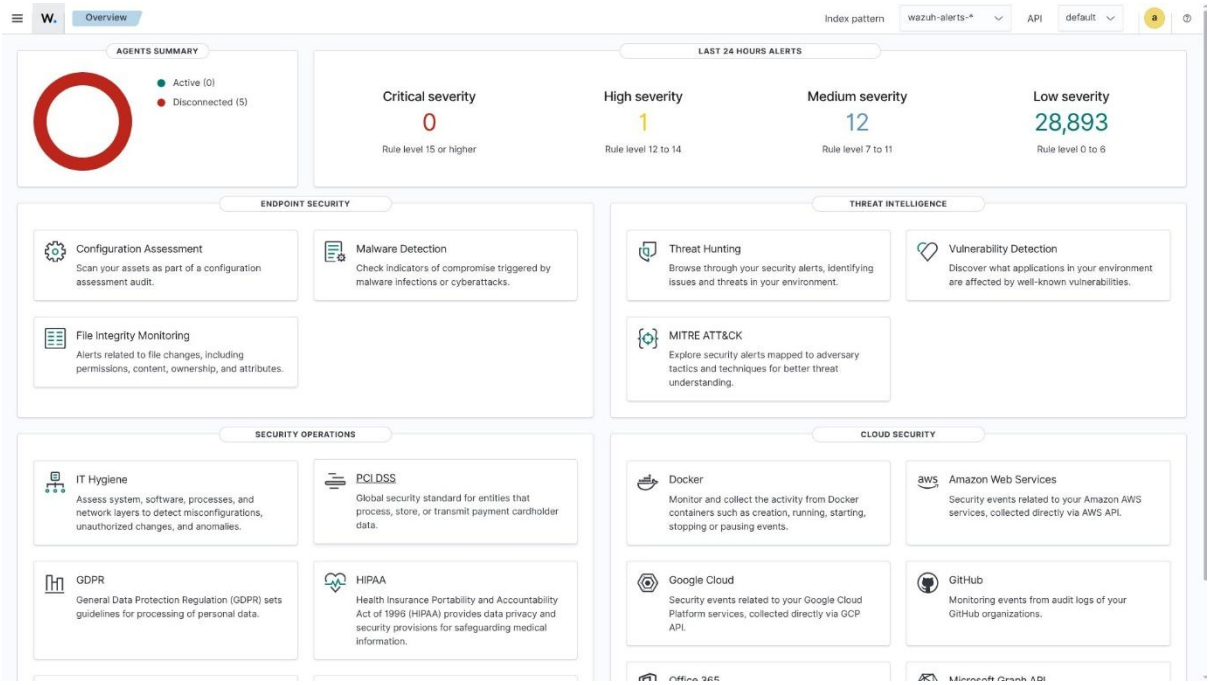

Fig. 3. Wazuh SIEM centralized dashboard aggregating alerts from multiple sources. During the experiment, **28,893 low-severity, 12 medium-severity, and 1 high-severity** alert were recorded. MITRE ATT&CK mapping and Vulnerability Detection modules are active.

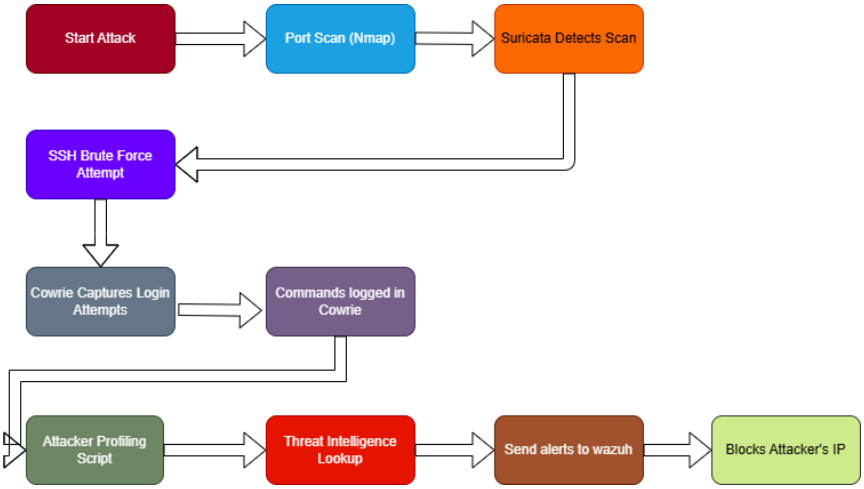

Fig. 4. Attack lifecycle and defensive workflow depicting the sequence from initial network scanning and brute-force attempts through honeypot interaction, command logging, AbuseIPDB threat intelligence verification, and automated firewall-based IP blocking

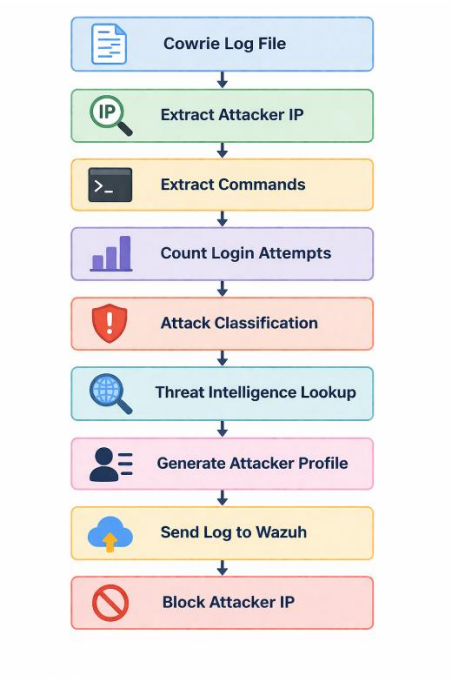

Fig. 5. Detailed operational sequence of the attacker profiling module: extraction of attacker IPs and executed commands from Cowrie logs, login attempt counting, attack behavior classification, threat intelligence lookup, attacker profile generation, Wazuh alert forwarding, and automated IP blocking.

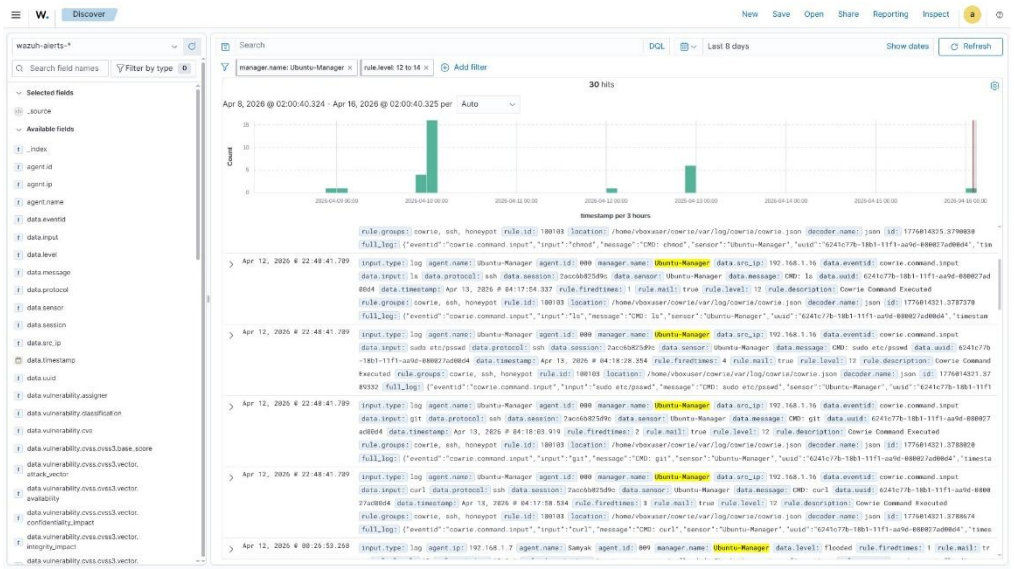

Fig. 6. Cowrie honeypot logs visualized in Wazuh Discover showing 30 high-severity SSH events over 8 days. Attacker commands including ls, sudo, git, and curl from IP 192.168.1.16 indicate active reconnaissance and privilege escalation attempts.

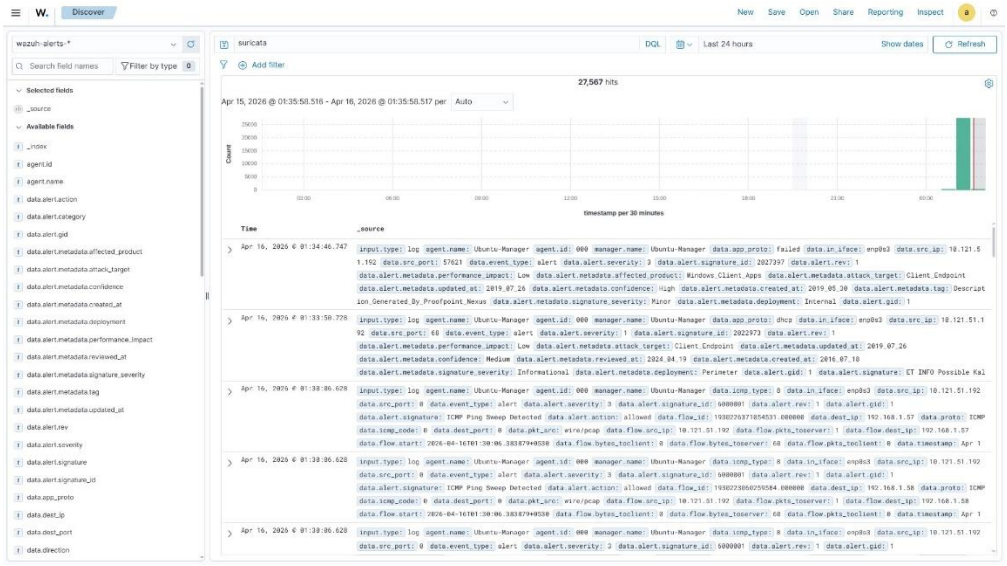

Fig. 7. Suricata network alert log showing 27,567 detections within a 24-hour monitoring window, including ICMP sweep activity, SSH brute-force attempts, and other anomalous network behavior indicative of active intrusion attempts.

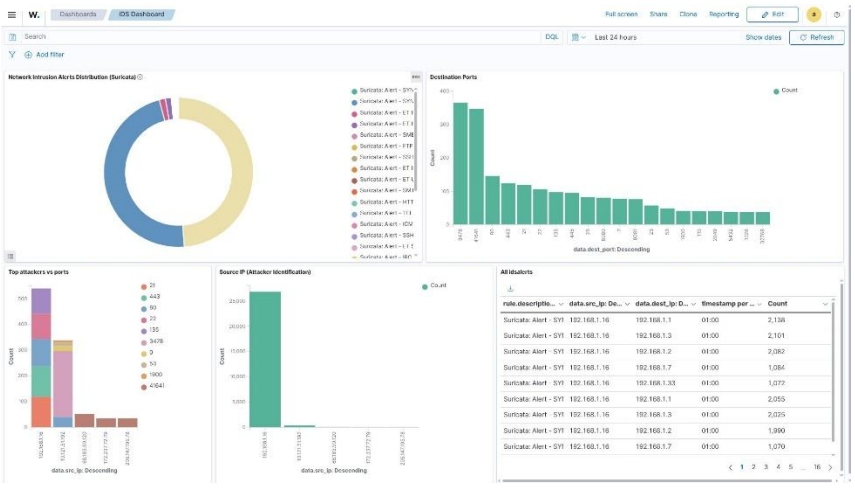

Fig. 8. IDS dashboard summarizing detected attack categories and attacker statistics. SYN flood attacks account for the highest alert volume with 2,138 detections. The most active attacker IP 192.168.1.16 generated over 25,000 security alerts during the observation period.

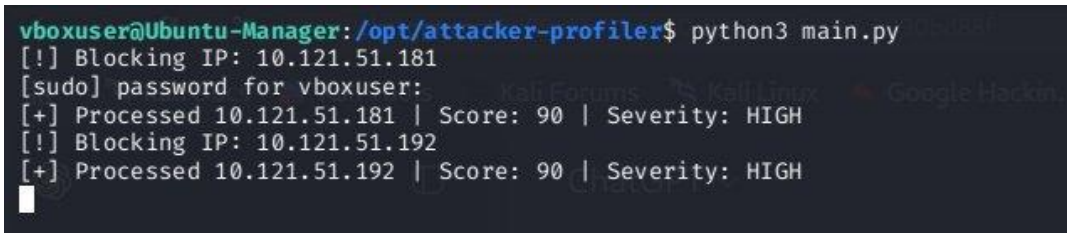

Fig. 9. Output of the attacker profiling script integrated with AbuseIPDB threat intelligence. Attacker IPs 10.121.51.181 and 10.121.51.192 received abuse confidence scores of 90 out of 100 and were classified as high-risk, triggering automated blocking.

Fig. 10. Automated response output confirming iptables DROP rules applied to 10.121.51.181 and 10.121.51.192 following threat intelligence validation, preventing further malicious communication without any manual intervention.

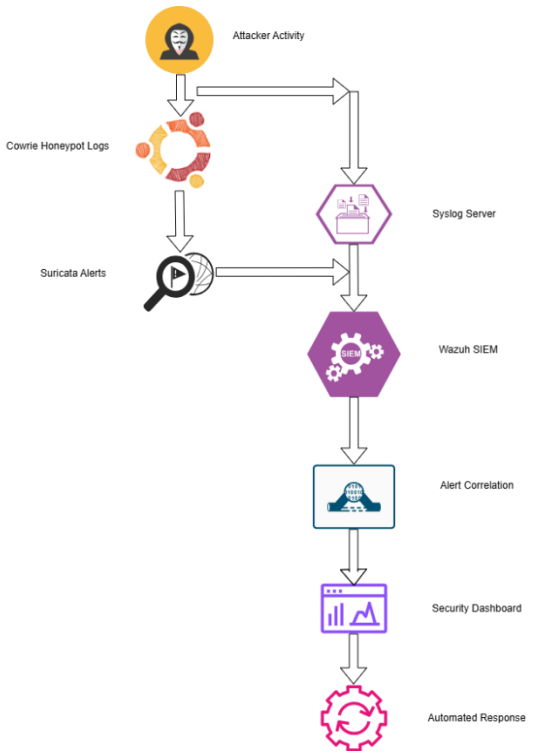

Fig. 11. Consolidated overview of the proposed security architecture depicting the interaction between attacker activity, honeypot logging, network intrusion detection, SIEM correlation, dashboard visualization, and automated response—demonstrating continuous multi-layer threat detection and mitigation.

7. ADVANTAGES OF THE PROPOSED SYSTEM

The proposed system: (1) combines deception and NIDS for layered detection [11]; (2) captures real attacker behavior via Cowrie [3]; (3) provides centralized SIEM correlation [1]; (4) enriches attacker data with AbuseIPDB threat intelligence [4]; (5) automatically blocks malicious IPs without human intervention; (6) uses exclusively open-source tools [15]; (7) simulates a full SOC environment on-premise.

8. LIMITATIONS AND FUTURE WORK

The system is deployed in a virtual lab and may not scale to enterprise networks [9]. AbuseIPDB API dependency may introduce latency. Automated blocking may produce false positives without careful threshold tuning [7]. Future work includes ML-based classification [6], real-time dashboards, cloud deployment [9], and additional honeypot types [5].

9. CONCLUSION

This paper presented a deception-based intrusion detection architecture integrating Cowrie, Suricata, Wazuh SIEM, threat intelligence, and automated response. Cowrie captured 30 high-severity attacker events; Suricata recorded 27,567 network alerts in 24 hours; Wazuh correlated all events centrally. The profiling module assigned AbuseIPDB scores of **90/100** to malicious IPs, which were blocked automatically via iptables. Table I confirms this is the only open-source on-premise architecture to unify all seven detection and response layers, making it a strong foundation for cybersecurity research and SOC simulation.

Conflict of Interest

The authors declare no conflicts of interest regarding the current research.

References

[1] Wazuh, Inc., *"Wazuh Open Source Security Platform Documentation,"* 2026.

[2] Open Information Security Foundation, *"Suricata NIDS Documentation,"* 2026.

[3] M. Oosterhof, *"Cowrie SSH/Telnet Honeypot,"* 2026.

[4] AbuseIPDB, *"AbuseIPDB Threat Intelligence API,"* 2026.

[5] The MITRE Corporation, *"MITRE ATT&CK Framework,"* 2026.

[6] P. Kapatel et al., *"AI-Powered Intrusion Detection System with Honeypot Integration," Int. J. Intelligent Information Systems*, vol. 14, no. 4, Sep. 2025.

[7] R. Pratama, D. Prayama, and F. Nova, *"Integration of Wazuh and Suricata with Telegram for Enhanced Threat Detection," Int. J. Research and Innovation in Social Science*, vol. 9, no. 2, 2025.

[8] Z. Morić, V. Dakić, and D. Regvart, *"Advancing Cybersecurity with Honeypots and Deception Strategies," Informatics*, vol. 12, no. 1, p. 14, Jan. 2025.

[9] L. Harani et al., *"A Practical Honeypot-Based Threat Intelligence Framework for Cyber Defence in the Cloud," arXiv preprint* arXiv:2512.05321, Dec. 2024.

[10] S. Sikdar, P. Basu, and M. Kule, *"Honeypot Deception: A Clever Approach of Web Intrusion Detection and Prevention,"* in *Proc. ICFCS*, Springer, 2024, pp. 141–152.

[11] A. Javadpour et al., *"A Comprehensive Survey on Cyber Deception Techniques to Improve Honeypot Performance," Computers & Security*, vol. 140, p. 103718, Mar. 2024.

[12] B. Alrashdan et al., *"Network Anomaly Detection using Artillery Honeypot and Wazuh SIEM," ResearchGate*, 2024.

[13] I. Balogh et al., *"Honeypots in Cybersecurity: Analysis, Evaluation and Importance," Preprints.org*, Aug. 2024.

[14] M. S. Yusof, H. Haron, and N. Omar, *"Deployment of Honeypot and SIEM Tools for Cyber Security Education in UiTM," ResearchGate*, Oct. 2022.

[15] H. Hindy et al., *"Research Trends in Network-Based Intrusion Detection Systems: A Review," IEEE*

Access, vol. 9, pp. 157439–157479, Nov. 2021.

[16] K. Scarfone and P. Mell, *"Guide to Intrusion Detection and Prevention Systems (IDPS),"* NIST Special Publication 800-94, Feb. 2007.

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