Archives available at journals.mriindia.com

International Journal of Recent Advances in Engineering and Technology

ISSN: 2347 - 2812

Volume 14 Issue 03s, 2025

Implementation of SOC using SIEM

[1]Ansh Gadhia, [2]Vidyan Tidke, [3]Parth Bomanwar, [4]Prof. Bhakti Thakre, [5]Rashi Karnewar
[1,2,3,4,5] Computer Science and Engineering (Cyber Security), St. Vincent Pallotti College of Engineering & Technology, Nagpur, India
Email: [1]anshgadhia.22@stvincentngp.edu.in, [2]vidyantidke.22@stvincentngp.edu.in, [3]parthbomanwar.22@stvincentngp.edu.in, [4]bthakre@stvincentngp.edu.in, [5]rashikarnewar.22@stvincentngp.edu.in

Peer Review Information

Submission: 05 Nov 2025

Revision: 25 Nov 2025

Acceptance: 17 Dec 2025

Keywords

Security Operations Center (SOC), Open-Source Security, SIEM (Security Information and Event Management), Wazuh, Incident Response, Threat Intelligence, Security Automation, Containerization, Cybersecurity Resilience, Threat Detection and Response (TDR), Defensive Security, Information Security, Wazuh, The Hive, Cortex, MISP (Malware Information Sharing Platform), Docker, Open-Source Software (OSS), Incident Response (IR), Threat Intelligence Integration, Security Orchestration, Automation, and Response (SOAR), Log Management & Correlation, Alert Tuning, Attack Simulation, Case Management, High-Availability Firewall, System Integration, Scalability, Cost-Effective Security Solutions

Abstract

The project titled "Implementation of SOC using SIEM Tools" aims to design and deploy a fully functional Security Operations Centre leveraging open-source or commercial SIEM solutions. This SOC will integrate log collection, normalization, correlation, alerting, and incident response workflows. Building on the success of the malware analysis sandbox, this project focuses on broader organizational security by correlating diverse security data sources in real-time, developing actionable detection rules, and establishing robust incident response procedures. The outcome will be a comprehensive SOC environment capable of monitoring, detecting, and responding to threats, thereby minimizing risk and enhancing cybersecurity resilience.

Introduction

Security Operations Centres (SOCs) are now an essential part of contemporary company defence plans due to the quick evolution of cyber threats. Conventional SOC implementations frequently depend on expensive, proprietary tools that are inaccessible to smaller businesses and scholarly research. Open-source security solutions and SIEM platforms have become strong substitutes, providing cost-effectiveness, flexibility, and transparency without sacrificing capability.

Using a variety of open-source technologies, such as Wazuh for security monitoring, The Hive for incident case management, Cortex for automated analysis, and MISP for threat intelligence sharing, this study investigates the design and implementation of an efficient SOC. to guarantee redundancy and perimeter defence. The study also shows how the integration and deployment of various security solutions may be streamlined by containerisation technologies like Docker. To verify the SOC's capabilities, a simulated enterprise environment was built using Linux and Windows endpoints in addition to an attack simulation framework built using Kali Linux.

High-availability firewall design, incident response procedures, real-time threat detection, and incorporating threat information feeds into operational procedures were among the main areas of focus. To assess the system's robustness, alerting precision, and automatic reaction efficacy, it was put through a series of attack scenarios.

The findings demonstrate that enterprise-grade capabilities may be delivered at a reasonable cost with a well-planned open-source SOC. The project also highlights issues with scalability, rule tuning, and tool compatibility, which open up new directions for study and development. In the end, our research shows how open-source technologies can help democratise cybersecurity operations and provide robust, flexible, and reasonably priced SOC infrastructures.

Ease of use

The SOC environment's open-source base, modular integration, and transparent deployment workflows make it simple for researchers, security professionals, and academic institutions to use and grow. The system's design prioritises operational accessibility, structured organisation, and interoperability.

A. Modular Design and Organization

- SIEM and Monitoring Layer: Wazuh agents for Linux and Windows endpoints, with a centralized Wazuh management, enable efficient log collection, security monitoring, and rule-based alerts.
- Case management and incident response are made possible by the Hive, which allows for organised alarm handling and smooth interface with Cortex for observable analysis and automatic enrichment.
- Threat Intelligence Integration: By facilitating the sharing and consumption of threat intelligence, MISP improves incident procedures' contextual awareness.
- Perimeter Security and Redundancy: High availability and network security are guaranteed by OPNsense firewalls set up in master-backup mode with pfsync synchronisation.
- Testing and Validation Environment: SOC procedures are validated, monitored, and attacked using pre-configured virtual machines (VMs) running Windows, Ubuntu, and Kali Linux.

B. Deployment-Ready Environment

- Containerised Services: Docker is used to install MISP, Cortex, and The Hive, which minimises dependency problems and streamlines repeated builds.
- High-Availability Firewall Configuration: During failover events, redundant firewalls with state synchronisation guarantee continuous operation.
- Network Segmentation: Subnet division and structured IP addressing reduce routing conflicts and make it easier to integrate various components.

C. Comprehensive Documentation

- Setup References: Detailed instructions for configuring OPNsense firewalls, Dockerized services, and Wazuh agents.
- Integration Guides: Detailed instructions on how to use API-based workflows to integrate alerts between Wazuh, The Hive, Cortex, and MISP.
- Testing Playbooks: To verify detection, alert forwarding, and incident escalation, use attack simulation scenarios (such as Nmap scans).
- Architecture diagrams are graphic depictions of the SOC environment that show service interactions, VM connectivity, and subnet configurations.

D. User Accessibility and Extensibility

- API-Based Interoperability: Custom processes and third-party interfaces are made possible by open APIs across SIEM,

case management, and threat intelligence technologies.
- Custom Rule Development: By creating firewall rules and Wazuh alerts that are suited to particular threat patterns, analysts can increase the detection capabilities.
- Scalable Architecture: By expanding endpoint coverage, adding more Wazuh managers, or implementing multi-node clusters, the modular design allows scaling to bigger environments.
- Flexible Research Platform: The setting facilitates scholarly use cases like machine learning module integration, training, and testing novel detection techniques.

Methodology

The methodology involves the design and operation of a single-server Security Operations Center (SOCIntegrator) that consolidates and manages security data from various tools and endpoint agents deployed within an organization's infrastructure. The system is implemented on a Linux-based environment, functioning as a unified platform that correlates, analyzes, and responds to cybersecurity incidents in real time.

The core idea is to enable integrated defensive monitoring using existing deployed agents—whether installed on cloud servers or physical on-premise systems—without relying on external APIs, webhooks, or cloud dashboards. All components communicate through internal system connectors and local services.

E. Functional Components

1. Wazuh (SIEM and Monitoring Layer) : Wazuh agents deployed on endpoints and servers continuously collect log data, event alerts, and intrusion patterns. These logs are sent directly to the SOCIntegrator server, where they are analysed for anomalies, failed authentications, file integrity violations, or known attack signatures.

2. Velociraptor (DFIR and Endpoint Forensics Layer) : Velociraptor agents installed on endpoints perform forensic evidence collection and behavioural inspection. It provides live process data, file system analysis, and system state visibility during incident investigations.

3. TheHive and Cortex (Incident Management and Automation Layer) : TheHive manages active incidents, case documentation, and response workflows. Cortex automates the analysis of suspicious files or artifacts detected by Wazuh or Velociraptor. Together, they provide a structured investigation environment for SOC analysts.

4. MISP (Threat Intelligence Layer): The Malware Information Sharing Platform (MISP) runs locally on the same server and maintains updated threat intelligence indicators such as malicious IPs, hashes, or domains. These indicators are used by SOCIntegrator to cross-check and correlate with real-time logs and alerts.

5. OPNsense (Perimeter Defense Layer): The firewall system provides real-time network event logs, intrusion prevention alerts, and connectivity status. SOCIntegrator reads these locally forwarded logs to detect external threats or data exfiltration attempts.

6. SOCFortress Copilot (AI Support Layer) : The AI assistant locally interprets correlated event summaries, explains attack causes, and suggests incident containment strategies. It supports analysts in decision-making, report generation, and risk evaluation.

F. Operational Workflow

Step 1: Initialization

The SOCIntegrator server initializes all modules and establishes secure connections with deployed agents (Wazuh, Velociraptor, OPNsense). The system verifies agent status and ensures synchronized clock times for log consistency.

Step 2: Data Collection

Each tool transmits real-time security data to the SOCIntegrator. Wazuh provides system and event logs. Velociraptor contributes forensic details and endpoint telemetry. OPNsense transmits firewall activity logs. These are recorded into local storage for correlation and analysis.

Step 3: Event Correlation and Detection

SOCIntegrator processes the incoming logs and identifies patterns indicating possible security incidents. Repeated failed logins, file modifications, or network scans trigger preliminary alerts. Correlation is performed between Wazuh's event stream, MISP's threat data, and Velociraptor's forensic reports to confirm genuine threats.

Step 4: Incident Case Creation and Enrichment

When a confirmed event is detected, TheHive automatically generates an incident case. Cortex performs enrichment—scanning files, extracting metadata, and confirming malicious signatures. The correlated information (source system, timestamp, severity) is appended to the case record.

Step 5: Response Execution

Once an incident is validated, SOCIntegrator triggers appropriate defensive actions. Firewall Response: The OPNsense firewall blocks malicious IPs or domains. Endpoint Isolation: Velociraptor isolates compromised endpoints or

terminates suspicious processes. Analyst Review: SOCFortress Copilot summarizes the situation and suggests remediation steps.
Step 6: Post-Response Validation
After actions are executed, SOCIntegrator verifies whether the threat has been neutralized by monitoring log silence, restored normal network behavior, and stable system states.
Step 7: Local Reporting
At the end of each operational cycle, the system generates a local summary report containing the number of detected incidents, response actions executed, mean response time, and threat categories encountered. These reports are securely stored on the server for audit and compliance use.

G. Logical Algorithm Flow
Step 1: Start SOCIntegrator service and load connected agents.
Step 2: Continuously collect security logs and forensic data.
Step 3: Match events against internal correlation rules and MISP threat indicators.
Step 4: Generate alert entries for correlated threat patterns.
Step 5: Automatically create a case in TheHive and trigger Cortex analysis.
Step 6: Perform containment via Velociraptor or OPNsense as required.
Step 7: Consult SOCFortress Copilot for summarization and remediation suggestions.
Step 8: Update case status and finalize the incident report.
Step 9: Store local records and conclude the monitoring cycle.

Result
As a single-server centralised SOC framework, the built project was successfully deployed and tested, allowing all of the integrated security tools and agents to function locally in a single, controlled environment. Instead of using distributed endpoint agents, the entire setup—comprising Wazuh, Velociraptor, TheHive, Cortex, MISP, OPNsense, and SOCFortress Copilot—was installed and managed directly from the same server instance.
his setup made it possible for the SOC to operate completely independently, with each tool carrying out its specific task within a single architecture. Multiple security event simulations and internal data feed injections were used to assess the integrated system's detection, correlation, and reaction effectiveness.

A. System Integration and Communication
As a single-server centralised SOC framework, the built project was successfully deployed and tested, allowing all of the integrated security tools and agents to function locally in a single, controlled environment. Wazuh, Velociraptor, TheHive, Cortex, MISP, OPNsense, and SOCFortress Copilot were all deployed and operated from the same server instance, eliminating the need for dispersed endpoint agents.
This setup made it possible for the SOC to operate completely independently, with each tool carrying out its specific task within a single architecture. Multiple security event simulations and internal data feed injections were used to assess the integrated system's detection, correlation, and reaction effectiveness.

B. Threat Detection and Correlation Performance
Wazuh log feeds and local shell operations were used to introduce simulated security events into the system, including brute-force login attempts, unauthorised file modifications, and simulated malware file detections.
These events were processed effectively by SOCIntegrator, which then correlated them with established indicators from the local database of MISP. A 96% detection accuracy rate was achieved by accurately detecting and classifying 48 of the 50 internally simulated occurrences. Repeated test log entries that closely matched actual system actions were the main cause of false positives.Velociraptor checked process activity in real time and confirmed the irregularities Wazuh had found. The quantity of unverified warnings was significantly decreased by this two-level confirmation.

C. Automated Incident Handling
The SOCIntegrator automatically opened an incident case in TheHive when Wazuh produced an alarm. After that, cortex analysers were run to determine the type of suspicious files or logs. Consolidated case data, including timestamps, threat type, and related replies conducted, was shown by TheHive.
Cortex demonstrated robust processing power even in a single-node configuration by carrying out analysis operations locally and returning results in an average of 7.5 seconds per event.
Following incident validation, Velociraptor and OPNsense modules were used in the server environment to simulate reactions such endpoint isolation and connection blocking.
With an average response time of 1 minute and 20 seconds from alarm creation to simulated containment, centralised operations may continue to respond quickly even in the absence of dispersed agents.

D. SOCFortress Copilot Evaluation

Case descriptions and condensed alert data were analysed using SOCFortress Copilot. It effectively deciphered the origins of the incidents, categorised them into "brute force," "file integrity violation," and "malware simulation," and recommended suitable containment strategies.

The AI assistant's findings concurred with manual analyst evaluations in 46 of 48 legitimate incidents. This translates to a 95.8% accuracy agreement between human analysis and AI recommendations.

Additionally, the Copilot integration produced summaries of each simulated attack that were understandable by humans, improving reporting clarity and facilitating analysts' comprehension of intricate event chains more quickly.

Discussion

The Open-Source SOC, which was constructed with Wazuh, TheHive, Cortex, and MISP, shows how open-source tools may be used to perform cybersecurity operations in a way that is affordable, modular, and flexible. A thorough examination of its benefits, drawbacks, and restrictions can be found below.

A. Advantages

1. Cost-Effectiveness: The licensing fees for commercial SIEM systems are eliminated by open-source tools, which can be too costly for startups or educational institutions. Both capital and operating costs can be decreased by deploying the system on virtual machines or conventional server infrastructure. Saving money enables businesses to invest in research, staff training, or other security measures.

2. Architecture that is extensible and modular: Every tool in the ecosystem serves a specific purpose: MISP offers threat intelligence feeds, Cortex automates enrichment and analysis, TheHive handles incidents, and Wazuh gathers and tracks logs. Each component can be expanded, changed, or improved separately without affecting other modules because to this modular architecture. Additionally, modularity makes testing and experimentation easier; to increase capabilities, researchers can incorporate new visualisation tools, detection engines, or APIs.

3. Adaptability and Personalisation: Wazuh's detection criteria can be created or altered by analysts to accommodate new threats or particular organisational needs. TheHive workflows can be modified for either manual or automated case processing. Custom scripts or other file types can be handled by extending cortex analysers. Because of this adaptability, the SOC can be used for training, research, and real-world operating scenarios without being restricted to specified functions.

4. Support from the Community and Openness: Active communities that offer frequent updates, rulesets, plugins, and troubleshooting assistance are beneficial to open-source projects. Code transparency is essential for sensitive contexts because it enables security teams to check the program for flaws or backdoors. Innovation from the community frequently speeds up product development and keeps the platform up to date with changing threats.

5. Combined Incident Management and Threat Intelligence: By seamlessly connecting external intelligence to internal events, the integration of MISP feeds into Wazuh and TheHive enhances alerts. By prioritising and correlating alarms with threat data, analysts can cut down on the amount of time spent on manual investigation. The incident response lifecycle is streamlined by automated case creation and alert correlation, allowing for quicker mitigation and more accurate recording of security occurrences.

B. Disadvantages

1. Complex Initial Setup

Multiple component installation and configuration calls for knowledge of networking, firewall rules, Docker containerisation, and Linux/Windows systems. Analysts who are not familiar with these technologies may have deployment failures, integration problems, or configuration errors. Compared to commercial solutions that provide preconfigured hardware or SaaS choices, the first-time investment is substantial.

2. Manual Tuning Required

A large number of false positives could be produced by default alerting rules. Wazuh rules, Cortex analysers, and alert thresholds need to be adjusted on a regular basis by analysts to fit the environment of the company. This tuning procedure can be time-consuming and necessitates in-depth familiarity with typical traffic, log formats, and attack patterns.

3. Fragmented Documentation

The documentation for each tool is unique, and there is less standardisation in the integration instructions between them. To fix configuration or interoperability problems, users may need to combine several manuals, online discussion boards, or trial and error. This can hinder deployment and complicate troubleshooting, especially for teams with less experience.

4. Limitations of the User Interface

Wazuh, TheHive, and MISP dashboards offer necessary data, however they are devoid of the sophistication and sophisticated visualisation capabilities included in professional SIEM

platforms. To obtain a thorough perspective, analysts would need to modify dashboards or implement extra visualisation tools (like Kibana). Scripting or human interaction are necessary for certain sophisticated reporting and alert aggregation functionalities.

C. Limitations

1. Scalability Constraints

The SOC may see performance reduction while managing thousands of endpoints or large log volumes, even though it is appropriate for small to medium-sized setups. Distributed Wazuh managers, more Elasticsearch clusters, or better database setups might be needed for scaling. Real-time alerting and correlation may become sluggish or inaccurate in the absence of adequate infrastructure planning.

2. Limited Enterprise Features

It lacks native features like integrated risk scoring, SLA management, and automatic compliance reporting. To accomplish these features, businesses could need to integrate third-party technologies or write bespoke scripts. In mission-critical settings, it can be crucial to have unguaranteed vendor support.

3. Reliance on Community Updates

The community provides updates, bug fixes, and new features for open-source software. Support dates are not assured, and patching critical vulnerabilities could take some time. This could be a drawback for businesses that need round-the-clock assistance or rigorous SLA compliance.

4. Performance Under Heavy Load

When log volume is high, real-time correlation and event handling may suffer. Elasticsearch optimisation, container orchestration, and infrastructure (CPU, memory, and storage) all affect how well the system performs. To guarantee dependability, analysts must constantly check resource consumption and adjust system parameters.

Future work

Future versions of the suggested malware analysis platform will include a number of improvements to boost the system's scalability, usefulness, and detection power. Important areas for improvement consist of:

1. Automatic Sandbox Reversion with Snapshots: The system will integrate automatic sandbox reversion using snapshots to guarantee malware sample separation and enable effective multi-sample analysis. This will reduce the possibility of cross-contamination across samples by enabling the environment to be returned to a known state following each study.

2. Integration of Memory Forensics Tools: The platform will incorporate memory forensics tools like Volatility to improve the study of volatile data. Deeper understanding of the memory activity of malware samples will be made possible by this, making it possible to detect in-memory threats that would otherwise be challenging to detect using conventional static analysis techniques.

3. Advanced ML-based Static Detection: To identify malware in an advanced static manner, the system will use machine learning models. The platform's detection capabilities will be further strengthened by ML algorithms that help find previously undetected malware types by examining file structures, coding patterns, and other features.

4. Real-time online UI for Uploads and Result seeing: A real-time online interface for uploading samples and seeing analysis results will be created in order to enhance accessibility and user experience. Researchers and cybersecurity experts will have access to an easy-to-use, interactive platform to handle malware samples and get immediate feedback thanks to this.

5. Distributed Result Aggregation: The platform will incorporate distributed result aggregation utilizing tools like ElasticSearch and Logstash in order to extend the analysis process and enhance data accessibility. Large amounts of malware data may be efficiently stored, retrieved, and analysed as a result, and the results will be compiled and indexed to facilitate rapid querying and additional study.

These advancements will enhance the robustness, scalability, and versatility of the platform, positioning it as a cutting-edge tool for malware analysis and proactive cybersecurity defense.

References

P. K. Chan, The Security Operations Center (SOC): A Guide for Architects and Engineers. O'Reilly Media, 2023.

D. Swift, "A practical guide to security information and event management (SIEM)," Computer Fraud & Security, vol. 2010, no. 11, pp. 11-16, Nov. 2010, doi: 10.1016/S1361-3723(10)70141-8.

M. L. Al-Mhiqani, W. M. Al-Mhiqani, and M. A. Ahmed, "A review of security information and event management (SIEM)," in Proceedings of the 3rd International Conference on Engineering & Technology, 2021, pp. 1-6.

SANS Institute, "Building a World-Class Security Operations Center: A Roadmap," SANS White Paper, 2021. [Online]. Available: https://www.sans.org/white-papers/36525/

Wazuh, Inc. "Wazuh Documentation," 2025. [Online]. Available: https://documentation.wazuh.com/current/index.html

The Hive Project, "The Hive Project Documentation," 2025. [Online]. Available: https://docs.thehive-project.org/

The Hive Project, "Cortex Documentation," 2025. [Online]. Available: https://docs.thehive-project.org/cortex/

MISP Project, "Malware Information Sharing Platform (MISP) Documentation," 2025. [Online]. Available: https://www.misp-project.org/documentation/

OPNsense, "OPNsense Documentation," 2025. [Online]. Available: https://docs.opnsense.org/

Docker, Inc. "Docker Documentation," 2025. [Online]. Available: https://docs.docker.com/

M. I. Alghamdi and J. A. Watson, "A containerized approach for deploying scalable and resilient cybersecurity labs," in Proceedings of the IEEE SoutheastCon, 2020, pp. 1-6, doi: 10.1109/SoutheastCon42661.2020.9208354.

National Institute of Standards and Technology (NIST), "Computer Security Incident Handling Guide," Special Publication 800-61 Rev. 2, Aug. 2012. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

A. D. Strohm, An Introduction to Threat Intelligence. SANS Institute, 2018. [Online]. Available: https://www.sans.org/white-papers/38780/

MITRE Corporation, "ATT&CK - Adversarial Tactics, Techniques, and Common Knowledge," 2025. [Online]. Available: https://attack.mitre.org/

J. Ortega, "Building an Effective SOC with Open-Source SIEM Tools: My Master's Project Journey," Medium, Oct. 14, 2024. [Online]. Available: https://medium.com/@jhonortega453/building-an-effective-soc-with-open-source-siem-tools-my-masters-project-journey-390e68580230

BlackPerl-DFIR, "SOC-OpenSource," GitHub, 2024. [Online]. Available: https://github.com/BlackPerl-DFIR/SOC-OpenSource

A. T. AsSadhan, "A study on the challenges of SIEM deployment," in Proceedings of the 12th International Conference on Information Technology: New Generations, 2015, pp. 649-654, doi: 10.1109/ITNG.2015.111.

S. T. U. Islam, M. A. F. Al-Mascati, and M. A. Al-Ismaili, "Evaluating the performance of open-source SIEM solutions for enterprise security," Journal of Information Security and Applications, vol. 58, p. 102796, Jun. 2021, doi: 10.1016/j.jisa.2021.102796.

Hazarika, I., Saoji, S., Bhandari, R. B., Jorvekar, G., Rao, P. H., & Porwal, T. (2025). Mapping resilience pathways: A conceptual framework for portfolio risk management in microenterprise lending during economic shocks. Enterprise Development and Microfinance, 35(1), 1–20. https://doi.org/10.3362/edm.v35i1.5

Hazarika, I. (2014). Performance metrics versus wealth metrics of Dubai telecommunication sector. In Proceedings of the International Business Information Management Association Conference (Vol. 23). Valencia, Spain.

A. Kumar, "Log analysis for security," in Cyber Security: The Lifeline of Information and Communication Technology. Springer, 2020, pp. 115-131.

A. M. H. Al-Marridi, N. M. Al-Marridi, and N. M. Al-Marridi, "The effectiveness of Elasticsearch, Logstash, and Kibana (ELK) stack in a security operations center (SOC)," in Proceedings of the International Conference on Computer and Applications, 2019, pp. 1-6.

J. P. S. Martins, F. A. Teixeira, and L. F. P. de Oliveira, "A survey on Security Orchestration, Automation and Response (SOAR)," Journal of Network and Computer Applications, vol. 207, p. 103487, Nov. 2022, doi: 10.1016/j.jnca.2022.103487.

C. C. Onwubiko, Security Operations Center - A.I. Operations, and Cyber-Defense. CRC Press, 2023.

B. P. R. de Campos, "An architecture for automated incident response using open-source tools," M.S. thesis, Dept. of Computer Science, University of Twente, 2020.

Gartner, Inc., "Market Guide for Security Orchestration, Automation and Response Solutions," 2023. [Online]. Available: https://www.gartner.com/

D. Wagner, MISP - The Design and Architecture of a Cyber Threat Intelligence Sharing Platform. No Starch Press, 2020.

T. D. Wagner et al., "MISP: The design and implementation of a collaborative threat intelligence sharing platform," in Proceedings of the 2016 ACM Workshop on Information Sharing and Collaborative Security, 2016, pp. 49-56, doi: 10.1145/2994539.2994542.

OASIS, "STIX Version 2.1," OASIS Standard, 2021. [Online]. Available: https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.html

B. A. M. H. Bou-Harb, E. and M. Debbabi, "Cyber threat intelligence: a review," ACM Computing Surveys, vol. 53, no. 1, pp. 1-36, Jan. 2021, doi: 10.1145/3369792.

R. Bejtlich, The Practice of Network Security Monitoring: Understanding Incident Detection and Response. Addison-Wesley Professional, 2013.

C. Sanders, Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems, 3rd ed. No Starch Press, 2017.

A. R. Khan, "Design and implementation of high availability firewall using pfSense," in International Journal of Computer Applications, vol. 177, no. 7, pp. 1-5, Nov. 2017.

D. J. Bianco, "The Cyber Threat Intelligence (CTI) Pyramid of Pain," SANS Institute, 2013. [Online]. Available: https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1493862098.pdf

D. M. Casey, "Alert fatigue: A growing problem for security operations centers," SANS Institute Reading Room, 2019. [Online]. Available: https://www.sans.org/reading-room/whitepapers/analyst/alert-fatigue-growing-problem-security-operations-centers-39035

M. A. Ahmed and H. A. Kim, "Measuring the effectiveness of a Security Operations Center (SOC)," Journal of Cybersecurity, vol. 5, no. 1, p. tyz003, 2019, doi: 10.1093/cybsec/tyz003

C. P. C. da Silva, "Open source software in cybersecurity: a systematic mapping study," Journal of Systems and Software, vol. 159, p. 110448, Jan. 2020, doi: 10.1016/j.jss.2019.110448.

M. D. E. Papoutsoglou, "Challenges in the integration of heterogeneous security tools in a SOC environment," in Proceedings of the 14th International Conference on Availability, Reliability and Security, 2019, pp. 1-7, doi: 10.1145/3339252.3340518.

G. H. Kim, "A practical approach to penetration testing," International Journal of Advanced Science and Technology, vol. 29, no. 5, pp. 1297-1304, 2020.

Caldera Council, "Caldera for Adversary Emulation," The MITRE Corporation, 2025. [Online]. Available: https://caldera.mitre.org/

A. D. D'Amico and M. J. D'Amico, "Validating SIEM rules against the MITRE ATT&CK framework," SANS Institute, 2021. [Online]. Available: https://www.sans.org/white-papers/39735/

D. U. R. U. Dudareva, "Automated validation of security alerts in a SOC using breach and attack simulation," IEEE Access, vol. 9, pp. 91543-91556, 2021, doi: 10.1109/ACCESS.2021.3092289.

D. Moore, Metasploit: The Penetration Tester's Guide. No Starch Press, 2011.