Network Security
Project Proposal

End-to-End SOC Automation using Open-Source SOAR Tools

Authors: Abdul Rafay & Kamil Saeed
Date: October 2025

Table of Contents

End-to-End SOC Automation using Open-Source SOAR Tools

1. Background and Motivation

In modern organizations, security teams are constantly flooded with thousands of alerts each day from firewalls, intrusion detection systems, and endpoints. Manually investigating and responding to these alerts is time-consuming, repetitive, and prone to human error. As a result, threats may remain undetected or unresolved for an extended period, thereby increasing the risk of data breaches.

Security Operations Centers (SOCs) rely heavily on automation to overcome this challenge. By integrating open-source tools, security teams can automate the collection, correlation, and response to alerts, resulting in faster incident handling and a reduced workload for analysts.

Our motivation for this project comes from the growing need to **modernize and automate SOC operations**. Instead of depending on expensive commercial solutions, we aim to create a **fully open-source, automated SOC** using tools such as **Wazuh, TheHive, Cortex, and Kibana**. This setup will enable automatic detection, case creation, analysis, and response, demonstrating the real-world benefits of automation in network defense.

2. Problem Definition and Complexity Justification

2.1 Problem Definition

Traditional SOC environments rely on manual workflows for incident response. When a security event occurs, analysts must manually correlate logs, check threat intelligence databases, create cases, and decide what action to take (such as blocking an IP). This process is slow and inefficient.

We aim to solve this problem by **automating the entire SOC workflow**, from alert generation to incident analysis and response, using open-source tools.

2.2 Why It is a Complex Computing Problem:

This project qualifies as a **complex computing problem** because it involves:

- **Multiple interacting systems:** SIEM (Wazuh), case management (TheHive), and automation engine (Cortex) integrated via APIs.
- **Real-time event handling:** Automatic forwarding of alerts and responses.
- **Automation and decision-making:** The system must intelligently trigger actions based on alert severity and type.
- **Scalability and reliability:** Ensuring components communicate securely and efficiently.
- **Balance between performance and accuracy:** Avoiding false positives while maintaining quick responses.

The project demonstrates integration, automation, and data-driven decision-making, key characteristics of complex security systems.

3. Objectives and Intended Outcomes

3.1 Objectives:

1. Design and implement a **fully automated SOC workflow** using open-source tools.
2. Automate incident response to reduce average response time.
3. Show how detection (Wazuh), management (TheHive), analysis (Cortex), and visualization (Kibana) connect.
4. Assess the system's performance based on detection accuracy, automation speed, and overall effectiveness.

3.2 Intended Outcomes:

- A functional SOC environment that automatically detects, analyzes, and responds to security incidents.
- Dashboards in Kibana showing real-time alerts, response actions, and performance metrics.
- Quantitative results demonstrating improvement in detection and response efficiency.

4. Methodology / Approach

We will implement the project in **four key phases**:

Phase 1: Environment Setup

- Deploy virtual machines or Docker containers for:
  - **Wazuh** (for detection and log management)
  - **TheHive** (for case management)
  - **Cortex** (for automated enrichment and response)
  - **Kibana** (for visualization and monitoring)
- Configure Wazuh agents on simulated endpoints to generate logs.

Phase 2: System Integration

- Configure **Wazuh → TheHive** integration using webhooks or APIs so that new alerts automatically create cases in TheHive.
- Integrate **TheHive → Cortex**, allowing Cortex to enrich cases using APIs (VirusTotal, AbuseIPDB).
- Ensure secure API communication among all components.

Phase 3: Automation and Response

- Configure **Cortex Analyzers and Responders** to automatically:
    - Enrich IPs, URLs, and file hashes.
    - Trigger response actions (e.g., blocking malicious IPs using a Python script or sending admin alerts).
- Create **playbooks** in TheHive to define automatic workflows for specific alert types.

Phase 4: Testing and Evaluation

- Simulate attacks using tools like **Kali Linux (Nmap, Hydra, Metasploit)** to generate real alerts.
- Observe the end-to-end workflow:
    1. Wazuh detects the attack.
    2. TheHive creates a case.
    3. Cortex enriches and responds automatically.
    4. Kibana visualizes the event timeline.
- Measure key metrics: response time, detection rate, and false positives.

5. Tools, Frameworks, and Technologies

Component	Tool / Technology	Purpose
SIEM	Wazuh	Intrusion detection and alert generation
Case Management	TheHive	Incident organization and workflow automation
SOAR/Automation	Cortex	Automated enrichment and response (via APIs)
Visualization	Kibana	Real-time dashboards and analytics
Log Storage	Elasticsearch	Data storage and indexing for Wazuh & Kibana
Scripting	Python / Bash	Custom responders (e.g., block IPs, send alerts)
Testing Environment	VirtualBox / Docker / Kali Linux	Virtual network simulation for attacks

6. Expected Evaluation Criteria

Our project will be evaluated on measurable and realistic criteria, including:

Metric	Description
Incident Detection Rate	Percentage of simulated attacks detected by Wazuh
Average Response Time	Time between detection and automated response
False Positive Rate	Percentage of incorrect detections or actions
Automation Accuracy	Correct execution of automated enrichment and responses
System Reliability	Stability and communication between integrated components
Visualization Quality	Effectiveness of Kibana dashboards for monitoring and analysis

7. Expected Deliverables

1. **Working SOC System:** Integrated environment demonstrating end-to-end automation (detection → analysis → response).
2. **Project Report (10–15 pages):** Detailed explanation of methodology, architecture, results, and evaluation.
3. **Demonstration:** Live demo showing a simulated attack detected by Wazuh and automatically handled by TheHive and Cortex.
4. **Screenshots and Dashboard Visuals:** Evidence of working setup and performance graphs from Kibana.
5. **Presentation Slides:** For final defense and explanation.

8. Timeline

Week	Task	Expected Outcome
Week 1	Environment setup (Wazuh, TheHive, Cortex, Kibana)	All components installed and connected
Week 2	Integration testing between Wazuh → TheHive → Cortex	Alerts automatically create cases
Week 3	Configure automation playbooks and responders	System performs enrichment and automated responses

Week	Task	Expected Outcome
Week 4	Testing, data collection, and dashboard visualization	Collect performance metrics and screenshots
Week 5	Final documentation and presentation preparation	Complete report and demo ready

9. Conclusion

This project aims to demonstrate how **automation can transform traditional SOC operations** by reducing human effort, improving response times, and providing real-time visibility through open-source tools. The integration of Wazuh, TheHive, Cortex, and Kibana showcases a practical, scalable, and cost-effective solution to a complex, real-world cybersecurity challenge.