VM-Based SOC Lab: Integrated Network and Host Monitoring System

Report of the Project

Abstract

This project demonstrates the design and implementation of an enterprise-grade Security Operations Center (SOC) lab within a virtualized environment. The SOC setup was built using multiple virtual machines (VMs) configured in bridge mode to enable full network visibility. At the core, an Ubuntu-based central SIEM system was deployed, integrating **Snort** for network intrusion detection and **Wazuh Manager** for log collection and analysis. Various systems, including **Windows Server 2019**, **Kali Linux**, and application servers, were configured with Wazuh agents, enabling comprehensive monitoring of system and application logs. Custom rules were created on the Wazuh Manager to monitor critical events across all agents, with alerts above level 3 configured to be sent directly to **Slack**. Attack simulations were performed to validate the effectiveness of the SOC, ensuring that both network and host-based monitoring were functioning correctly. Additionally, a web application was hosted on an **AWS EC2 instance** and connected to the Wazuh Manager via **Tailscale VPN**, allowing continuous remote monitoring. This project demonstrates the practical deployment of an integrated SIEM solution, combining network and host monitoring, alerting, and secure remote log collection for enterprise-grade cybersecurity management.

Objective & Problem Statement

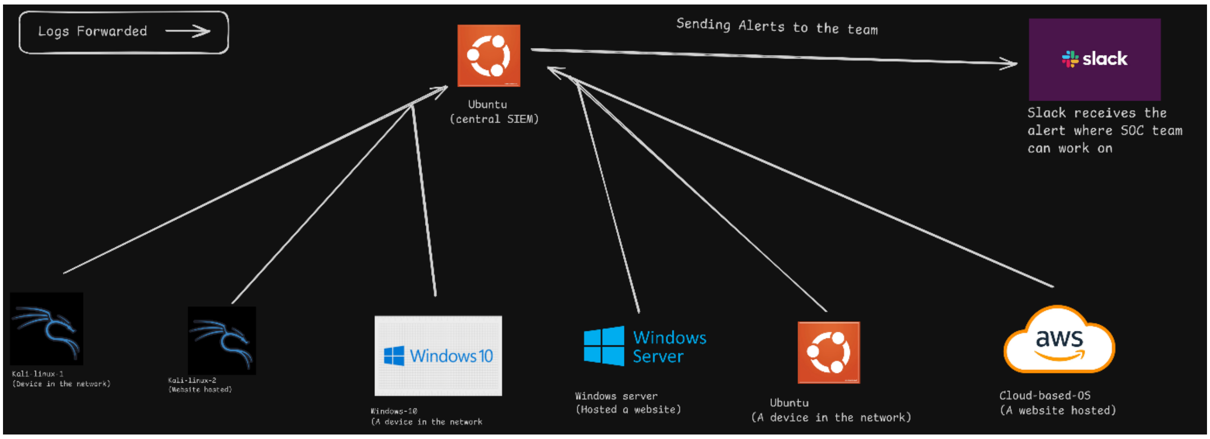
Modern enterprise networks face significant challenges in maintaining comprehensive security visibility across diverse systems and applications. Security events are often dispersed across multiple devices and platforms, making it difficult to detect and respond to threats in a timely manner. This project addresses the challenge of centralized security monitoring and alert automation by implementing a fully functional Security Operations Center (SOC) lab in a virtual environment.

The primary objectives of this project are:

1. To establish a centralized SIEM system capable of monitoring network traffic and host activity across multiple virtual machines.
2. To integrate network intrusion detection (via Snort) and host-based monitoring (via Wazuh) for unified visibility into security events.
3. To implement custom alerting rules that automatically notify administrators of critical incidents, including integration with communication tools such as Slack.
4. To simulate attacks in a controlled environment to validate the effectiveness of detection and alerting mechanisms.
5. To demonstrate secure remote monitoring by connecting a hosted web application on AWS EC2 to the central SOC using a VPN.

This project not only provides hands-on experience in SOC operations but also demonstrates a scalable approach to centralized monitoring, alert automation, and proactive threat detection, which are critical requirements for enterprise cybersecurity management.

System Architecture / Lab Setup



The SOC lab is built entirely within a virtualized environment, using multiple virtual machines (VMs) interconnected in bridge mode to allow full network visibility. The architecture integrates network and host monitoring tools with alerting and cloud connectivity for a realistic enterprise setup.

Components and Roles:

1. Ubuntu (Central SIEM Server)
   - Role: Central monitoring system
   - Tools:
     - Wazuh Manager: Aggregates logs from all connected agents, performs analysis, and applies custom rules for alerting.
     - Snort: Network intrusion detection system (NIDS) that monitors all network traffic in promiscuous mode.
   - Connectivity: Receives logs from all VMs and forwards critical alerts to Slack via webhook integration.
2. Windows Server 2019 (Web Server)

- Role: Hosts enterprise web application
- Tools: Wazuh agent installed to send system and application logs to the central manager
- Connectivity: Connected to SIEM server for centralized monitoring

3. Kali Linux (Attack Simulation & Pentesting VM)
    - Role: Simulates attacks on network and hosts to validate SOC detection capabilities
    - Tools: Offensive security tools for vulnerability scanning and exploit testing
    - Connectivity: Sends logs and monitored events to Wazuh Manager via agent
4. Other VMs / Linux Servers
    - Role: Hosts additional services or endpoints for realistic enterprise scenarios
    - Tools: Wazuh agents for log monitoring
    - Connectivity: Bridge mode for network visibility and integration with central SIEM
5. AWS EC2 Instance (Cloud Web Application)
    - Role: Demonstrates secure remote monitoring
    - Tools: Wazuh agent installed on the instance, connected through Tailscale VPN to the SIEM
    - Connectivity: Continuous log collection from cloud-hosted service to central manager
6. Slack Integration
    - Role: Receives alerts for events above severity level 3

- Tools: Slack webhook integrated with Wazuh Manager
- Connectivity: Provides real-time notification of critical security incidents

Implementation Summary

This project demonstrates the step-by-step implementation of a fully functional SOC lab within a virtualized environment. The following summarizes the key phases of implementation:

1. Operating System (OS) Setup
    - Multiple virtual machines were deployed in bridge mode to enable full network visibility.
    - Ubuntu served as the central SIEM server hosting Wazuh Manager and Snort.
    - Windows Server 2019 hosted a web application and acted as a monitored endpoint.
    - Kali Linux was used for attack simulations and penetration testing.
    - Additional Linux servers were configured as endpoints for realistic enterprise scenarios.
    - An AWS EC2 instance was set up to demonstrate secure remote monitoring through Tailscale VPN.
2. Snort + Wazuh Integration
    - Snort was installed on the Ubuntu SIEM VM in promiscuous mode to capture and analyze all network traffic.
    - Wazuh Manager was configured to collect logs from all connected agents, including Windows, Linux, and cloud-hosted systems.
    - Custom Wazuh rules were implemented to monitor critical system events and network activities.

    - Log decoders were fine-tuned to ensure accurate detection and aggregation of security events across all agents.

3. Slack Alert Setup
    - Alerts with a severity level above 3 were configured to trigger notifications.
    - Wazuh Manager was integrated with Slack via a webhook to send real-time alerts for critical security events.
    - The setup ensures immediate notification of potential threats for proactive incident response.
4. Attack Simulation and Validation
    - Simulated attacks were carried out from the Kali Linux VM to test detection capabilities of the SIEM setup.
    - Various scenarios, including web application exploitation, brute-force attempts, and suspicious network activity, were executed.
    - Logs and alerts were successfully collected, analyzed, and displayed on the Wazuh dashboard.
    - Slack notifications were received for all critical alerts, confirming the proper functioning of the alerting mechanism and overall SOC lab setup.

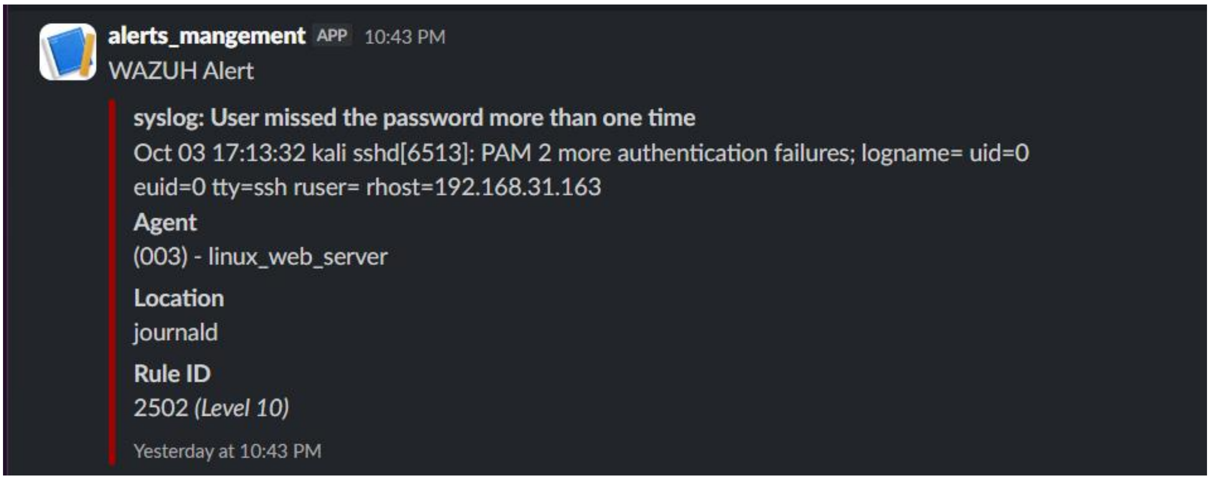
Outcome:
The implementation validates the effectiveness of a centralized SOC environment, demonstrating integrated network and host monitoring, automated alerting, and secure remote log collection for both on-premises and cloud-hosted systems.

Results / Observations

The SOC lab was successfully implemented, integrating both network and host-based monitoring across multiple virtual machines and a cloud instance. Key observations include:

- Real-time detection: Snort captured network-level anomalies, including suspicious traffic and port scans, while Wazuh agents collected system and application logs from all endpoints.
- Custom alerting: Alerts above severity level 3 were automatically sent to Slack, providing timely notification of potential threats.
- Attack simulation validation: Simulated attacks from Kali Linux, including brute-force attempts and web exploits, triggered corresponding alerts in the Wazuh dashboard.
- Centralized visibility: All logs from Windows Server 2019, Linux servers, and AWS EC2 instance were successfully aggregated, enabling a comprehensive view of network and host activity.

Sample Alert / Log Entry Examples:



alerts_mangement APP 10:43 PM
WAZUH Alert

syslog: User missed the password more than one time

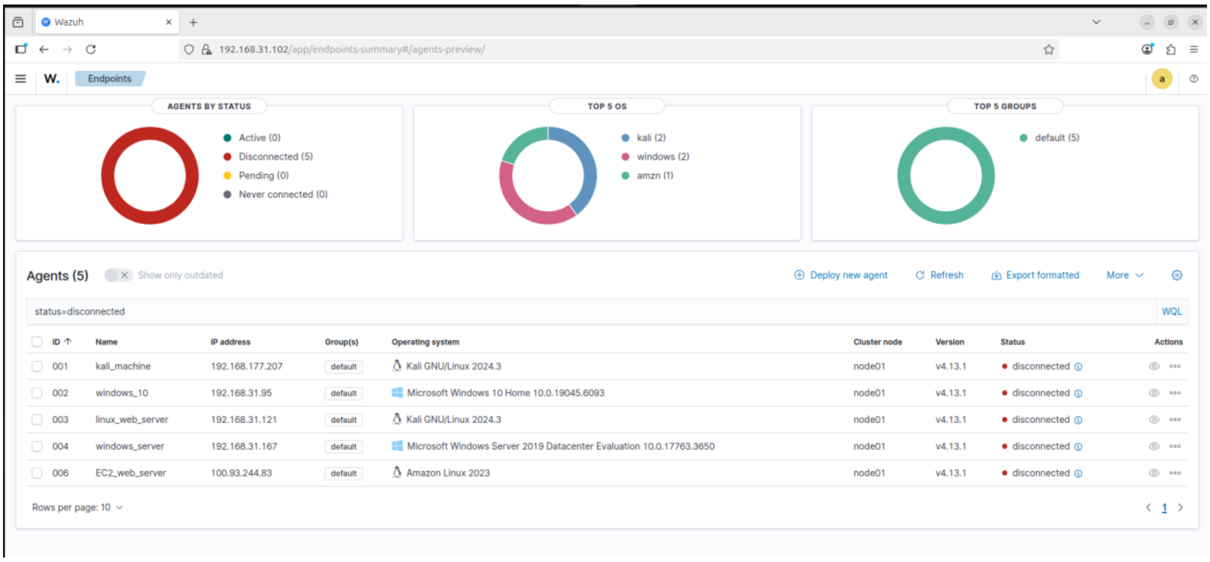
```
Oct 03 17:13:32 kali sshd[6513]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.31.163
```

Agent
(003) - linux_web_server

Location
journald

Rule ID
2502 *(Level 10)*

Yesterday at 10:43 PM



status=disconnected

ID ↑	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	kali_machine	192.168.177.207	default	Kali GNU/Linux 2024.3	node01	v4.13.1	● disconnected	
002	windows_10	192.168.31.95	default	Microsoft Windows 10 Home 10.0.19045.6093	node01	v4.13.1	● disconnected	
003	linux_web_server	192.168.31.121	default	Kali GNU/Linux 2024.3	node01	v4.13.1	● disconnected	
004	windows_server	192.168.31.167	default	Microsoft Windows Server 2019 Datacenter Evaluation 10.0.17763.3650	node01	v4.13.1	● disconnected	
006	EC2_web_server	100.93.244.83	default	Amazon Linux 2023	node01	v4.13.1	● disconnected	

Rows per page: 10

Conclusion & Future Enhancements

Conclusion:

The project demonstrates the practical deployment of an enterprise-grade SOC lab capable of centralized monitoring, real-time alerting, and attack detection. By integrating Snort and Wazuh, configuring agents across multiple systems, and enabling Slack notifications, the setup provides a comprehensive security monitoring framework for both on-premises and cloud-hosted environments. The successful simulation of attacks validates the reliability and effectiveness of the SOC architecture.

Future Enhancements:

1. Advanced Threat Intelligence Integration: Incorporate threat feeds and automated correlation to detect sophisticated attacks.
2. Machine Learning for Anomaly Detection: Implement ML models in Wazuh or SIEM to identify unusual patterns beyond predefined rules.
3. Extended Cloud Monitoring: Expand monitoring to multiple cloud providers and containerized environments (Docker/Kubernetes).

4. Automated Incident Response: Integrate SOAR (Security Orchestration, Automation, and Response) tools to automate response for critical alerts.
5. Web Application Security: Implement detailed monitoring for web application attacks (SQLi, XSS) with automated alerts and reporting.

This SOC lab provides a strong foundation for enterprise-level security monitoring while offering multiple avenues for further enhancement and automation.