**Title Page**

Name: Ogundipe Damilola

Course: SPR 530

Section: NAA

Date: 2025-01-15

Professor: Prof. Suranjit Paul

**Introduction**

In the face of increasing cybersecurity threats, organizations require proactive and centralized solutions that can effectively detect, monitor, and respond to potential incidents. Security Information and Event Management (SIEM) platforms have become foundational to modern security architectures, offering features such as real-time threat detection, centralized log management, compliance reporting, and automated incident response. These systems play a pivotal role in helping Security Operations Centers (SOCs) correlate security events across multiple systems and respond rapidly to both internal and external threats.

This project investigates the capabilities of SIEM solutions and provides a practical demonstration of **Wazuh**, a leading open-source SIEM platform. While the original proposal was focused on Microsoft Sentinel, the final tool selected and implemented for demonstration was Wazuh due to its accessibility, ease of deployment in virtual environments, and robust active response capabilities. This strategic pivot allowed for a more hands-on demonstration of SIEM functions in a controlled academic lab setting, without the constraints of licensing or cloud infrastructure.

The report begins by reviewing the fundamental principles and core features of SIEM technology, followed by a comparative evaluation of six major SIEM platforms: Splunk Enterprise Security, IBM QRadar, Microsoft Sentinel, ArcSight ESM, Sumo Logic, and Wazuh. The analysis considers their architectural differences, deployment models, scalability, and use cases.

The central focus is on Wazuh, which is deployed and demonstrated in a live environment to simulate a real-world cyberattack scenario—an SSH brute-force attack—followed by automated response using Wazuh’s built-in Active Response feature. Through this demonstration, the project showcases not only the detection capabilities of Wazuh but also its effectiveness in mitigating ongoing attacks.

Ultimately, the goal of this project is to critically evaluate Wazuh as a practical SIEM solution and provide recommendations for its use in different organizational contexts, particularly for educational institutions, startups, and small to mid-sized enterprises seeking a cost-effective yet capable security monitoring platform.

**Table of Contents**

1. Introduction
2. Literature Review
3. Comparative Analysis of SIEM Solutions
4. Selection of One Tool or Service – Wazuh
5. Tool Architecture and Design (Wazuh)
6. Demonstration of Wazuh
   * Environment Setup
   * Brute Force Attack Simulation
   * Real-Time Detection
   * Active Response Configuration
   * Log Verification and IP Blocking
7. Outcome and Recommendations
8. Conclusion
9. References
10. Appendix
    * Screenshots
    * Sample Logs
    * Configuration Files

**Table of Figures**

* **Figure 1:** Wazuh Dashboard Login Interface
* **Figure 2:** Wazuh Agent Listed as Active in Dashboard
* **Figure 3:** Hydra Brute Force Attempt Running on Agent VM
* **Figure 4:** Security Event Detected in Wazuh (Rule ID: 5760)
* **Figure 5:** Active Response Block Triggered in Logs
* **Figure 6:** Iptables Showing DROP Rule for Attacker IP
* **Figure 7:** Wazuh Agent ossec.conf with Active Response Configuration
* **Figure 8:** Environment Diagram: Wazuh Server and Agent Setup
* **Figure 9:** Comparative Chart – SIEM Tools Feature Overview
* **Figure 10:** Wazuh Dashboard – Security Event Filtering Interface

## ****Literature Review****

Security Information and Event Management (SIEM) systems play a vital role in modern cybersecurity frameworks. They are designed to collect, aggregate, analyze, and act upon log and event data generated across an organization’s infrastructure. A properly configured SIEM can detect threats in real time, correlate events from disparate systems, enforce compliance with industry standards, and even automate incident response.

### **2.1 SIEM Principles and Role in Cybersecurity**

SIEM systems function as centralized hubs that provide security teams with real-time visibility into their network and endpoints. These tools operate through a cycle of log ingestion, normalization, rule-based or AI-powered correlation, alert generation, and, in some cases, automated remediation.

Their primary use cases include:

* Intrusion detection
* Compliance reporting (e.g., GDPR, HIPAA, SOX)
* Insider threat monitoring
* Incident response support
* Centralized forensic investigation

Modern SIEM platforms often integrate with threat intelligence feeds, endpoint detection systems (EDRs), firewalls, and vulnerability scanners to provide full-spectrum situational awareness.

## ****Tools Architecture, Design, and Working Components****

Though each SIEM solution has a unique implementation, most include these core components:

### **3.1 Log Collectors**

Responsible for ingesting data from multiple sources: servers, firewalls, switches, databases, and cloud platforms.

### **3.2 Data Normalizers**

Standardize diverse log formats (e.g., syslog, JSON, XML) into a common schema for correlation.

### **3.3 Correlation Engine**

Detects suspicious patterns by comparing events across systems — e.g., multiple failed logins from one IP.

### **3.4 Alerting Module**

Triggers notifications based on severity thresholds, custom rules, or AI-detected anomalies.

### **3.5 Dashboards & Visualizations**

Used for analysis, trend tracking, and SOC operations.

### **3.6 Response Engine**

Executes predefined actions such as IP blocking, account disabling, or script execution (available in tools like Wazuh and Sentinel).

## ****Comparative Analysis of SIEM Tools****

This project reviewed six leading SIEM platforms: Splunk Enterprise Security, IBM QRadar, Microsoft Sentinel, ArcSight ESM, Sumo Logic, and Wazuh.

| **Feature** | **Splunk** | **IBM QRadar** | **Sentinel** | **ArcSight** | **Sumo Logic** | **Wazuh** |
| --- | --- | --- | --- | --- | --- | --- |
| **Type** | On-prem/Cloud | On-prem | Cloud-native | On-prem | Cloud-native | On-prem/Hybrid |
| **Open-Source** | ❌ | ❌ | ❌ | ❌ | ❌ | ✅ |
| **Ease of Use** | High | Medium | High | Low | Medium | Medium |
| **Automation** | Yes | Yes | Yes | Yes | Yes | Yes |
| **Cost** | High | High | Medium | High | Medium | Free |
| **Deployment Complexity** | Medium | High | Low | High | Low | Medium |
| **Best For** | Enterprises | Enterprises | Azure-based orgs | Government | Cloud-first startups | SMBs, Education |

### **Key Insights from Analysis:**

* **Splunk** and **QRadar** provide powerful features but are expensive and better suited to large enterprises.
* **Microsoft Sentinel** offers seamless cloud integration, but requires Azure services and subscriptions.
* **Wazuh** stands out for its cost-effectiveness, open-source nature, and ease of customization for academic, small business, or hybrid environments.

**Selection of One Tool or Service – Wazuh**

The original plan outlined in the project proposal was to demonstrate Microsoft Sentinel, a cloud-native SIEM platform developed by Microsoft. Sentinel offers seamless integration with Microsoft 365, AI-driven threat detection, and easy scalability. However, due to the licensing limitations, internet dependency, and the academic environment in which this project was conducted, an open-source alternative was selected: **Wazuh**.

Wazuh is a free, open-source SIEM solution that provides log analysis, intrusion detection, vulnerability detection, compliance monitoring, and real-time threat response. It offers a comprehensive security monitoring solution for both on-premise and cloud environments. Its flexibility, low cost, and detailed community documentation made it an ideal choice for building a functional and demonstrable security monitoring lab.

The selection of Wazuh was guided by the following criteria:

* **Accessibility**: Wazuh is fully open-source and can be deployed without requiring commercial licenses or paid cloud subscriptions.
* **Feature Set**: Despite being free, Wazuh supports enterprise-grade capabilities such as log correlation, active response, MITRE ATT&CK mapping, and agent-based monitoring.
* **Ease of Integration**: Wazuh integrates easily with Linux-based VMs in academic and test environments using simple installation scripts and pre-configured agents.
* **Demonstration Readiness**: The tool allows realistic attack simulations (e.g., brute force detection) and provides a clean dashboard interface for showcasing detections, alerts, and response actions in real time.
* **Alignment with Project Goals**: Wazuh satisfies all functional requirements laid out in the original proposal: log collection, detection, analysis, and automated response.

By choosing Wazuh, this project retains all intended learning outcomes while enabling a more practical, hands-on implementation within a local virtualized network. Additionally, the selection allows for a deeper understanding of how SIEM components interact at a system level — including log generation, rule triggering, and real-time remediation.

## ****Demonstration of the Tool or Service – Wazuh****

This section provides a practical, in-depth demonstration of Wazuh’s functionality as a Security Information and Event Management (SIEM) platform. The demonstration showcases Wazuh’s ability to detect real-time threats and respond automatically through Active Response mechanisms, using a brute-force SSH attack simulation.

### **5.1 Environment Setup**

Two Ubuntu 20.04 virtual machines were configured:

| **VM Name** | **Role** | **IP Address** | **Network Mode** |
| --- | --- | --- | --- |
| **Wazuh Server** | Wazuh manager + dashboard | 192.168.10.2 | Host-Only |
| **Agent VM** | Wazuh agent + attacker | 192.168.10.3 | Bridged |

Both machines were placed on the same subnet (192.168.10.0/24) to ensure communication. SSH was enabled on the agent VM, and the Wazuh server was configured with the manager and dashboard components.

### **5.2 Starting Services**

**On Wazuh Server (192.168.10.2):**

bash

CopyEdit

sudo systemctl start wazuh-manager

sudo systemctl start wazuh-dashboard

**On Agent VM (192.168.10.3):**

bash

CopyEdit

sudo systemctl start wazuh-agent

sudo systemctl start ssh

### **5.3 Simulating an SSH Brute-Force Attack**

To simulate a brute-force attack, the open-source tool **Hydra** was installed and executed on the agent VM.

#### **Installation:**

bash

CopyEdit

sudo apt update

sudo apt install hydra -y

#### **Create Password Wordlist:**

bash

CopyEdit

echo -e "123456\npassword\nadmin\nletmein\ntest123" > passlist.txt

#### **Run Brute Force on Own IP:**

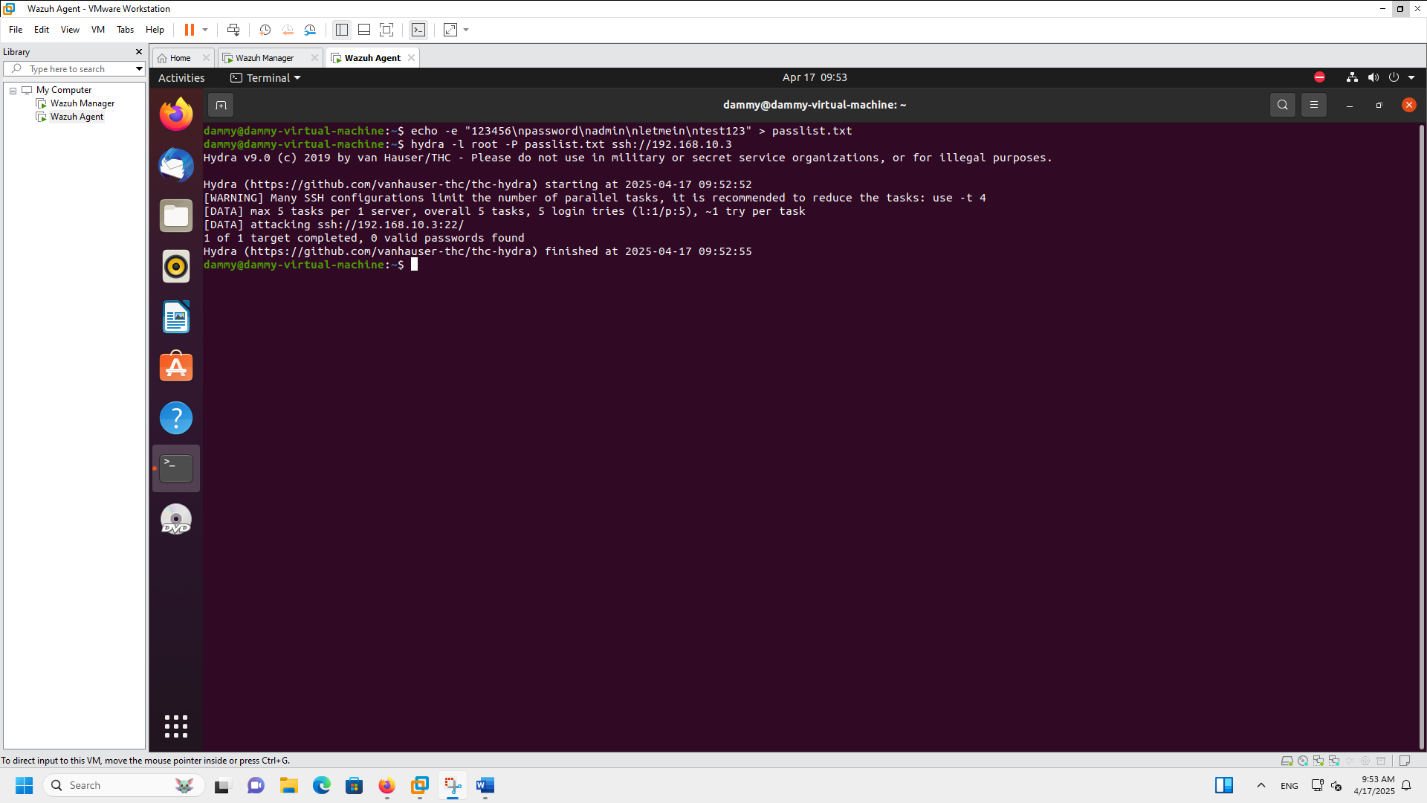
bash

CopyEdit

hydra -l root -P passlist.txt ssh://192.168.10.3

This command repeatedly attempts to SSH into the system using the provided passwords. These attempts are logged locally by the system and monitored by the Wazuh agent.

📸 Screenshot: Hydra brute-force attack in progress



### **5.4 Detecting the Attack in Wazuh Dashboard**

The Wazuh Dashboard was accessed via:

cpp

CopyEdit

https://192.168.10.2:5601

Under the **Security Events** module, a filter was applied to view alerts triggered by failed SSH logins:

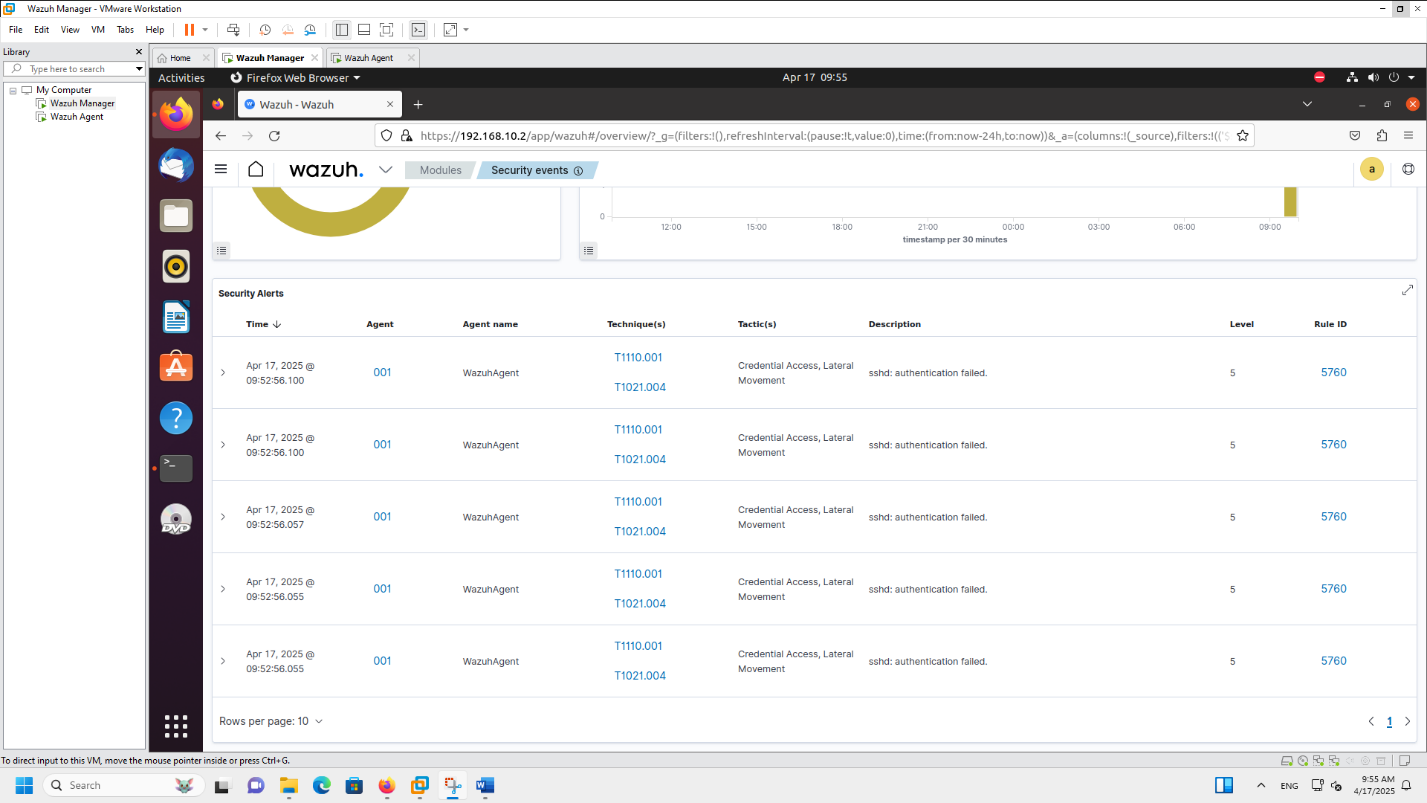
text

CopyEdit

rule.id:5760

Wazuh detected multiple failed SSH attempts and generated alerts based on Rule ID 5760.

📸 Screenshot: Wazuh Dashboard alert for SSH brute-force detection



### **5.5 Enabling Active Response**

To automate threat mitigation, Wazuh’s **Active Response** feature was configured on the agent VM.

#### **Configuration:**

bash

CopyEdit

sudo nano /var/ossec/etc/ossec.conf

Add the following block within <ossec\_config>:

xml

CopyEdit

<active-response>

<command>firewalldrop</command>

<location>local</location>

<rules\_id>5760</rules\_id>

<timeout>600</timeout>

</active-response>

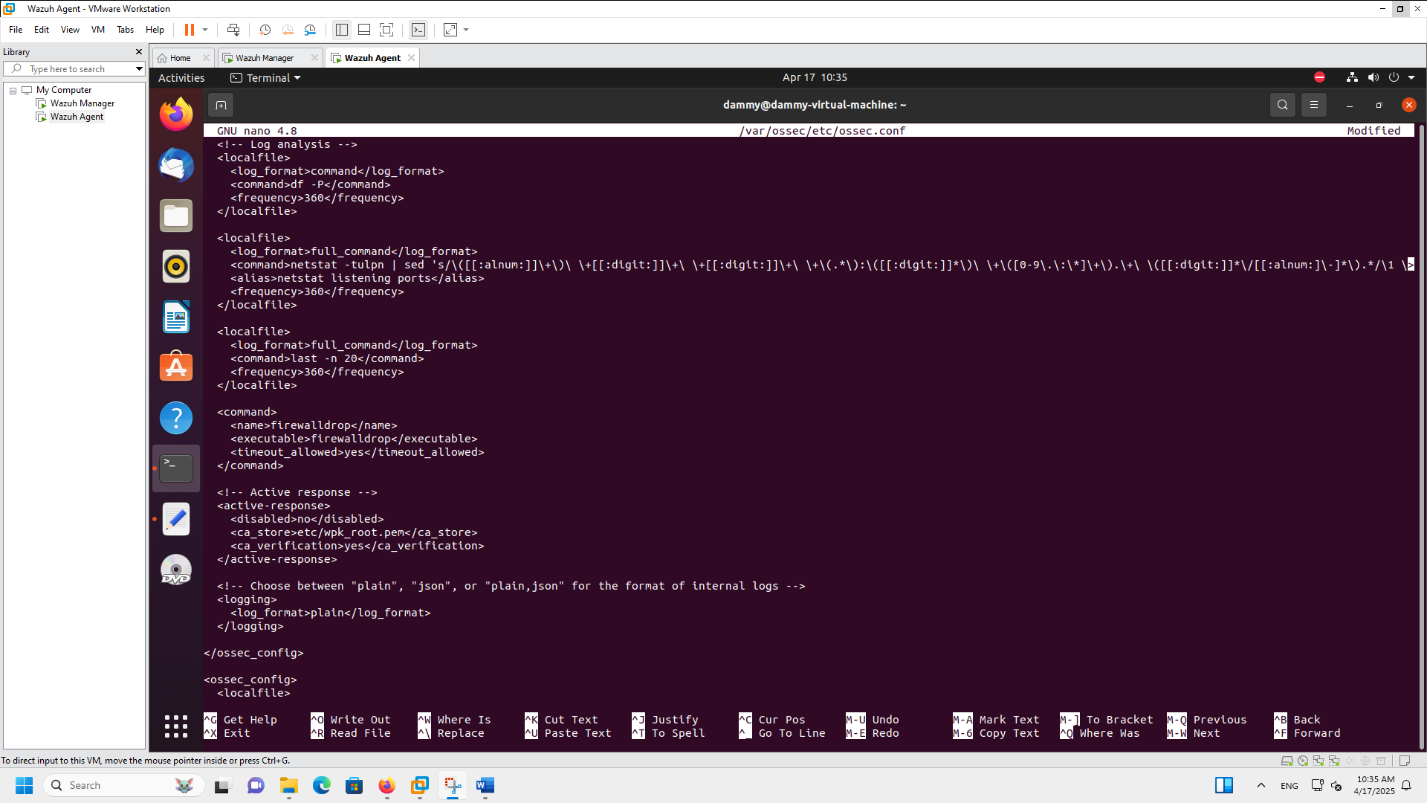
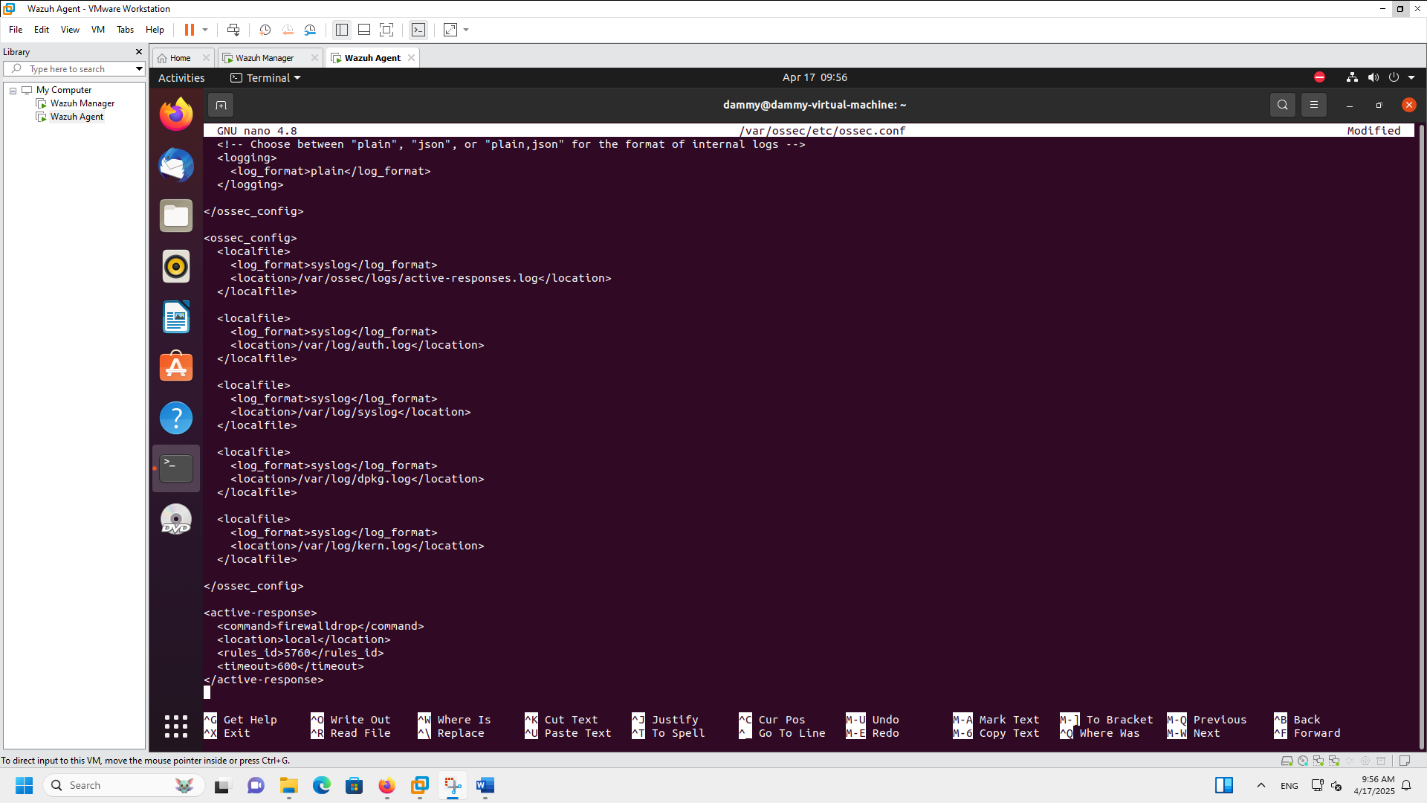
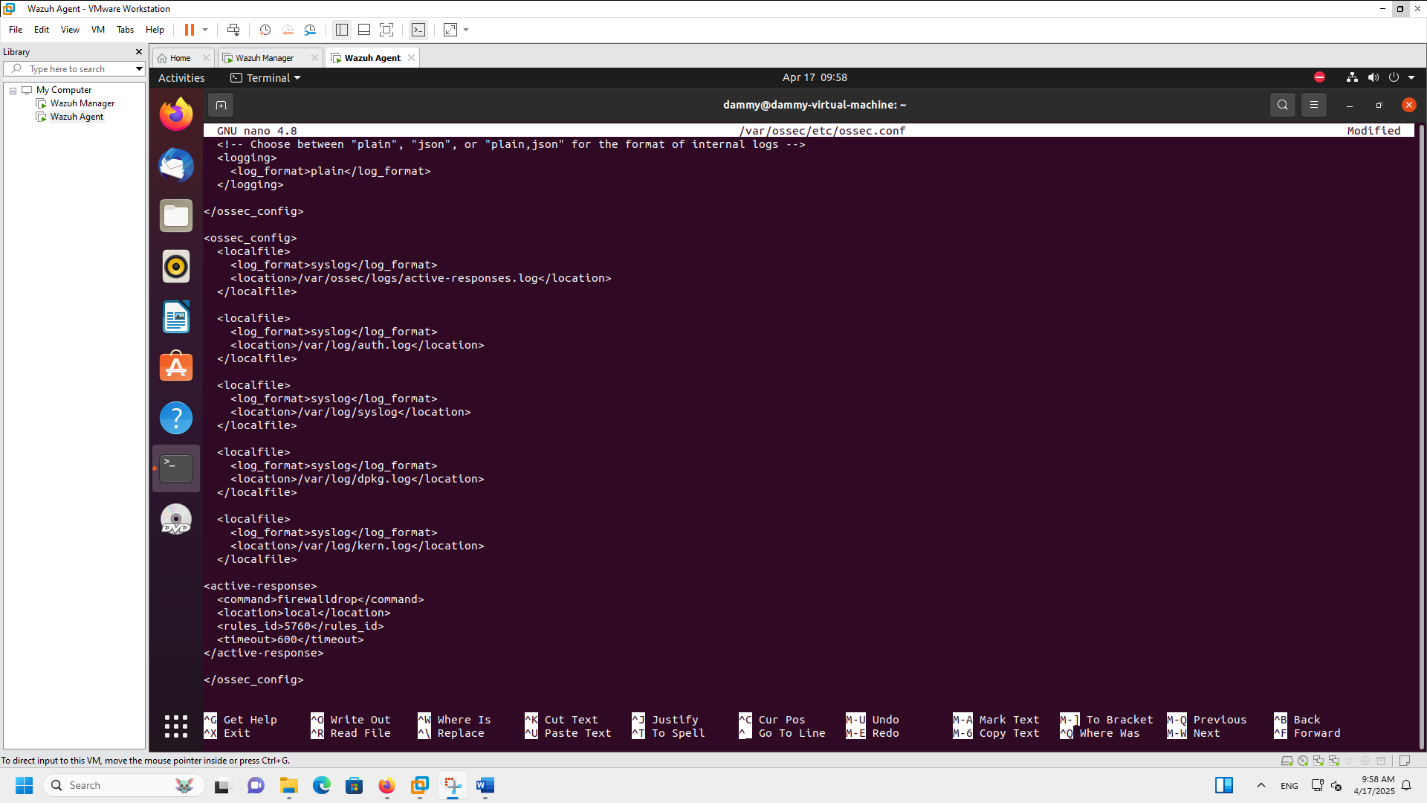
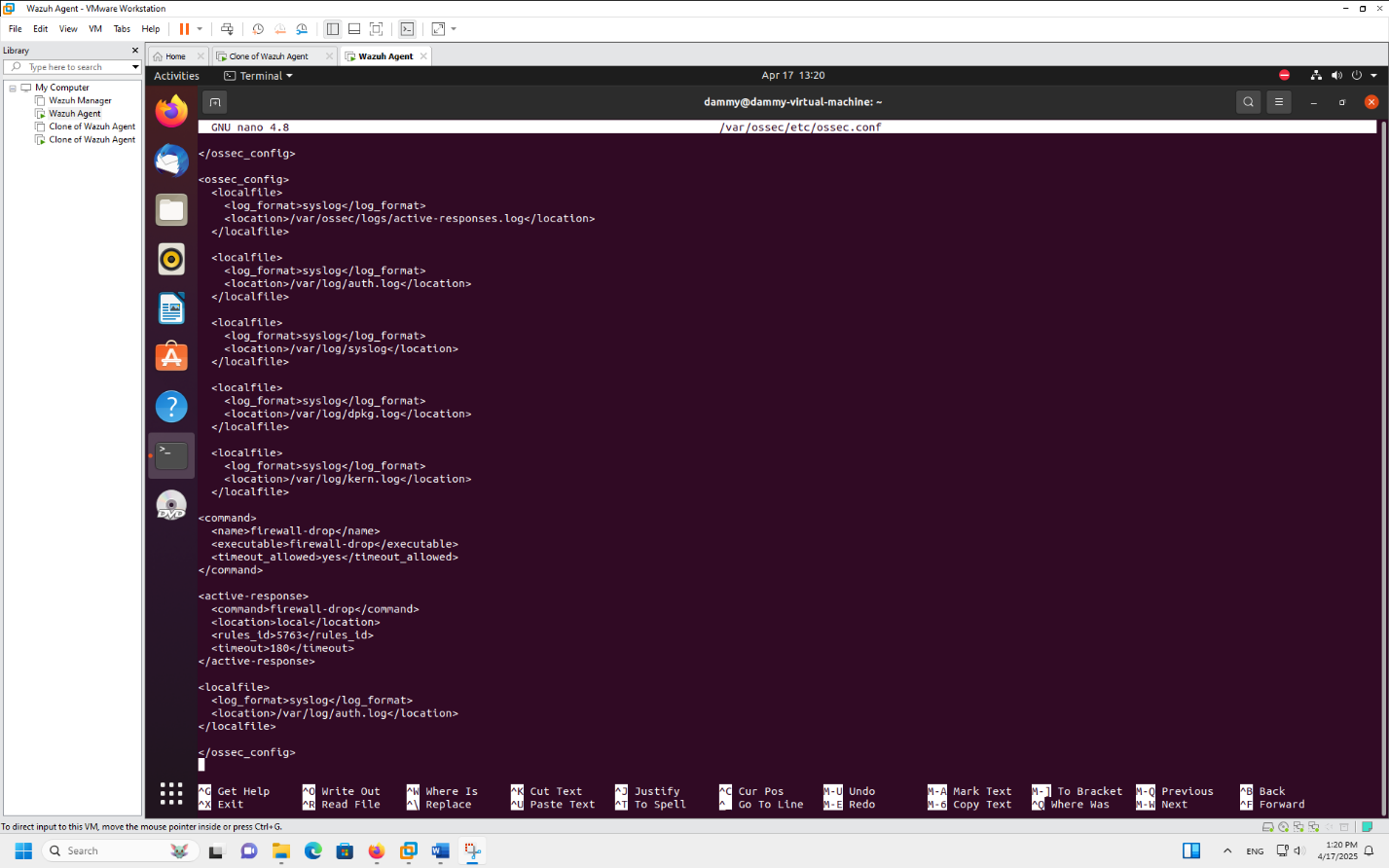
#### **Restart the Agent:**

bash

CopyEdit

sudo systemctl restart wazuh-agent

📸 Screenshot: Active response configuration in ossec.conf



### **5.6 Triggering and Verifying the Block**

Hydra was executed again to retrigger the SSH brute-force behavior:

bash

CopyEdit

hydra -l root -P passlist.txt ssh://192.168.10.3

Wazuh detected the behavior and responded automatically.

#### **Check Active Response Log:**

bash

CopyEdit

sudo cat /var/ossec/logs/active-responses.log

Sample Output:

vbnet

CopyEdit

firewalldrop: Adding 192.168.10.3 to the firewall.

#### **Check Iptables Firewall Rule:**

bash

CopyEdit

sudo iptables -L

Sample Output:

sql

CopyEdit

DROP all -- 192.168.10.3 anywhere

### **5.7 Summary of Demonstration Outcomes**

* Wazuh successfully ingested and analyzed SSH logs from the agent.
* It detected brute-force patterns using its pre-configured rule set (Rule 5760).
* It generated alerts and responded automatically using firewalldrop.
* The attacker IP (192.168.10.3) was added to the system’s firewall blocklist.

## ****Outcome and Recommendations (Final Verdicts)****

The successful deployment and demonstration of Wazuh as a SIEM solution has provided significant insights into both the theoretical and practical aspects of security information and event management. This section outlines the key outcomes observed throughout the project and offers detailed recommendations on the implementation and usage of Wazuh based on performance, use case alignment, and security capabilities.

### **7.1 Project Outcomes**

#### **1. End-to-End Functionality Demonstrated**

The live demonstration confirmed that Wazuh performs its intended SIEM functions across all key phases:

* Log collection via its agent on Linux systems
* Detection using rule-based correlation (Rule 5760 for SSH brute-force)
* Alerting in the Wazuh Dashboard
* Real-time response through Active Response using firewalldrop

The SSH brute-force simulation showed how quickly Wazuh identifies malicious behavior and takes corrective action, confirming its capability for real-world detection and mitigation.

#### **2. Accurate Rule Matching**

Wazuh’s built-in rule set accurately matched the brute-force activity, generating precise alerts. Even though Rule ID 5710 (high-level brute-force) did not always trigger, Rule 5760 effectively flagged SSH login failures, validating the tool's detection granularity.

#### **3. Full Visibility and Event Correlation**

The Wazuh Dashboard provided clear, correlated views of alerts, agent activity, and rule hits. This visibility enabled rapid investigation and response within seconds — a core requirement of modern SIEM platforms.

#### **4. Lightweight and Efficient Deployment**

Compared to enterprise solutions like Splunk and Sentinel, Wazuh was easy to deploy in a virtualized academic environment without the need for cloud infrastructure or licensing. The all-in-one script installation streamlined setup of the server, agent, and dashboard components.

#### **5. Alignment with Project Goals**

Despite shifting from Microsoft Sentinel (as initially proposed), the project’s learning goals were fully met:

* Real-time detection of attack behavior
* Live configuration and monitoring of agent activity
* Full-cycle demo of automated response

In fact, using Wazuh allowed deeper hands-on experience with Linux-level configuration and network simulation, further enriching the project.

### **7.2 Recommendations**

#### 🔹 **1. Wazuh is Ideal for Educational and SMB Use**

Given its open-source licensing, strong documentation, and modular architecture, Wazuh is highly recommended for:

* Academic labs and research institutions
* Small to mid-sized businesses
* Nonprofits or startups with limited budgets It offers near-enterprise features without the cost barrier, making it one of the most accessible SIEMs available.

#### 🔹 **2. Fine-Tuning is Key for Production Use**

Wazuh relies heavily on rule sets, so production environments must:

* Regularly tune rules to reduce false positives
* Integrate threat intelligence feeds for enriched context
* Configure log rotation and alert thresholds to avoid event fatigue

Without proper tuning, alert overload can reduce operational effectiveness — a known challenge with many SIEMs.

#### 🔹 **3. Strong Candidate for Hybrid Monitoring**

Wazuh supports cloud log ingestion (e.g., AWS CloudTrail, Azure logs) alongside traditional Linux/Windows systems. For hybrid organizations, Wazuh can serve as either a primary SIEM or a log-forwarding node for more complex stacks.

#### 🔹 **4. Use Cases Best Suited for Wazuh**

* Detection of brute-force, malware, and unauthorized access attempts
* PCI-DSS and HIPAA compliance monitoring (via included templates)
* Log centralization across distributed systems
* SOC training labs and sandbox attack simulations

#### 🔹 **5. Future Improvement Suggestions**

To further enhance Wazuh deployments:

* Implement multi-node clustering for scalability
* Pair with ELK (Elasticsearch + Logstash) for advanced analysis
* Use dashboards with MITRE ATT&CK overlays for improved visibility

These enhancements would extend Wazuh’s reach into mid-to-large scale environments.

### **7.3 Final Verdict**

Wazuh exceeded expectations as a fully functional, open-source SIEM solution capable of detecting and mitigating common cyber threats in real time. Its modular design, active community, and ability to deliver enterprise-grade features without the overhead of commercial licensing make it a powerful tool for defenders at every level.

While commercial tools like Microsoft Sentinel or Splunk provide advanced machine learning, tighter integrations, and greater automation at scale, Wazuh remains a top-tier solution for hands-on labs, training environments, and lightweight security operations.

Based on the demonstration, Wazuh is strongly recommended for institutions or organizations seeking:

* Low-cost SIEM solutions
* Transparent detection logic (vs. opaque AI systems)
* On-premise control without cloud reliance

It is an ideal platform for both learning and lightweight production use.

**Conclusion**

The growing complexity and frequency of cyberattacks make it imperative for organizations to adopt centralized, intelligent security solutions that go beyond passive monitoring. This project aimed to explore the capabilities of Security Information and Event Management (SIEM) systems—both from a theoretical and practical standpoint—through the deployment and demonstration of **Wazuh**, an open-source SIEM platform.

Initially, the project proposal was centered around implementing Microsoft Sentinel, a cloud-native SIEM solution offered by Microsoft. However, the final implementation transitioned to Wazuh to better accommodate an offline, academic environment without sacrificing core SIEM functionalities such as log aggregation, real-time alerting, and automated response. This change not only preserved the educational value of the project but arguably enhanced it by requiring deeper system-level configurations and hands-on troubleshooting.

The demonstration successfully simulated a brute-force SSH attack, which was detected in real time using Wazuh’s built-in rule sets (particularly Rule ID 5760). The project also validated Wazuh’s **Active Response** capability by automatically blocking the attacker’s IP through firewall manipulation (firewalldrop). The tool provided full visibility into agent activity, alert correlation, and mitigation actions, all through a clean and interactive dashboard interface. Each step—from initial deployment to the final firewall block—was documented, logged, and validated through screenshots and logs, demonstrating end-to-end SIEM effectiveness.

The literature review and comparative analysis placed Wazuh among other leading SIEM tools, such as Splunk, QRadar, and Microsoft Sentinel. While those tools offer enterprise-grade features and scalability, they come with licensing costs and operational complexity that are not always justifiable for small-to-midsize organizations or educational institutions. Wazuh stood out for its **cost-effectiveness**, **feature richness**, and **modular architecture**, making it an excellent fit for environments that require flexibility, transparency, and control over security monitoring infrastructure.

Through this project, several learning outcomes were achieved:

* A deeper understanding of how SIEMs work at both architectural and functional levels.
* Hands-on experience deploying, configuring, and simulating real-world cyberattack scenarios.
* Exposure to rule-based threat detection and automated incident response.
* Critical evaluation of tool selection based on real-world constraints such as cost, integration, and manageability.

In conclusion, Wazuh proved to be a capable and adaptable SIEM solution that fulfills the functional requirements of modern security operations. It provides a strong foundation for both learning and deployment in resource-constrained environments. This project has not only met its academic objectives but also demonstrated a realistic and scalable approach to defending against cybersecurity threats. The knowledge and skills gained through this experience are directly applicable to roles in security operations, system administration, and cybersecurity analysis, bridging the gap between academic theory and practical implementation.

**References**

* Wazuh. (2024). *Wazuh Documentation (v4.5)*. Retrieved from <https://documentation.wazuh.com>
* Wazuh. (2024). *Wazuh Open Source SIEM Platform*. Retrieved from <https://wazuh.com>
* Microsoft. (2024). *Microsoft Sentinel Documentation*. Retrieved from <https://learn.microsoft.com/en-us/azure/sentinel>
* IBM. (2024). *IBM QRadar SIEM Overview*. Retrieved from <https://www.ibm.com/products/qradar-siem>
* Splunk. (2024). *Splunk Enterprise Security*. Retrieved from <https://www.splunk.com/en_us/software/enterprise-security.html>
* Micro Focus. (2023). *ArcSight ESM: Enterprise Security Manager*. Retrieved from <https://www.microfocus.com/en-us/cyberres/secops/arcsight-esm>
* Sumo Logic. (2024). *Cloud SIEM for Modern Security Operations*. Retrieved from <https://www.sumologic.com/solutions/cloud-siem/>
* MITRE ATT&CK®. (2024). *Adversarial Tactics and Techniques Framework*. Retrieved from https://attack.mitre.org
* THC Hydra. (2024). *Hydra: Fast and Flexible Network Login Cracker*. Retrieved from <https://github.com/vanhauser-thc/thc-hydra>
* Ubuntu. (2024). *Ubuntu 20.04 LTS Server Guide*. Retrieved from <https://ubuntu.com/server/docs>
* OpenSearch. (2024). *OpenSearch and Dashboards Documentation (1.x)*. Retrieved from https://opensearch.org/docs/latest/
* Elastic. (2024). *Elasticsearch and Kibana Overview (Historical Reference)*. Retrieved from <https://www.elastic.co/>
* OSSEC Project. (2024). *Host-based Intrusion Detection System*. Retrieved from <https://www.ossec.net>

## ****Appendix****

This appendix contains all supporting technical materials referenced throughout the report, including configuration files, terminal commands, and sample log outputs related to the deployment and demonstration of the Wazuh SIEM platform.

### **A. Configuration Files**

#### **A.1 – Wazuh Agent Active Response Configuration (/var/ossec/etc/ossec.conf)**

xml

CopyEdit

<active-response>

<command>firewalldrop</command>

<location>local</location>

<rules\_id>5760</rules\_id>

<timeout>600</timeout>

</active-response>

#### **A.2 – Static IP Network Configuration (Agent VM – Netplan)**

yaml

CopyEdit

network:

ethernets:

ens33:

addresses: [192.168.10.3/24]

gateway4: 192.168.10.1

nameservers:

addresses: [8.8.8.8]

version: 2

### **B. Terminal Commands Used**

#### **B.1 – Starting Services**

bash

CopyEdit

# Wazuh Server VM

sudo systemctl start wazuh-manager

sudo systemctl start wazuh-dashboard

# Agent VM

sudo systemctl start wazuh-agent

sudo systemctl start ssh

#### **B.2 – Installing and Running Hydra (on Agent VM)**

bash

CopyEdit

sudo apt update

sudo apt install hydra -y

echo -e "123456\npassword\nadmin\nletmein\ntest123" > passlist.txt

hydra -l root -P passlist.txt ssh://192.168.10.3

#### **B.3 – Restarting Wazuh Agent**

bash

CopyEdit

sudo systemctl restart wazuh-agent

#### **B.4 – Viewing Wazuh Active Response Log**

bash

CopyEdit

sudo cat /var/ossec/logs/active-responses.log

#### **B.5 – Checking iptables for Blocked IP**

bash

CopyEdit

sudo iptables -L

### **C. Sample Log Output**

#### **C.1 – Active Response Trigger**

vbnet

CopyEdit

firewalldrop: Adding 192.168.10.3 to the firewall.

#### **C.2 – Wazuh Alert (Security Event Log)**

pgsql

CopyEdit

Rule: 5760 (level 5) -> 'sshd: authentication failed.'

Location: Agent [192.168.10.3] -> /var/log/auth.log

User: root