EncryptEdge Labs

Cybersecurity Analyst Internship

Task Report

atalmamun@gmail.com

Task No: 17

Credit: Offensive Security

Table of Contents

1.0 EncryptEdge Labs Internship Task Report

1.1 Introduction

In today's rapidly evolving cybersecurity landscape, traditional security measures such as antivirus and firewalls are no longer sufficient to protect endpoints from sophisticated and persistent threats. Endpoint Detection and Response (EDR) technologies have emerged as a vital component in modern security strategies, providing advanced capabilities for detecting, investigating, and responding to potential security incidents on endpoint devices.

This report focuses on exploring EDR technologies with a practical emphasis on **Wazuh**, an open-source security platform that integrates security information and event management (SIEM) with EDR features. Through this task, I gained both theoretical understanding and hands-on experience in deploying and configuring Wazuh to enhance endpoint security through real-time monitoring, threat detection, and incident response.

1.2 Objective

The main objectives of this task are:

- To understand the significance of Endpoint Detection and Response (EDR) in modern cybersecurity frameworks.
- To explore and evaluate the core functionalities of EDR solutions, particularly focusing on Wazuh.
- To set up a virtualized lab environment and configure Wazuh for basic threat detection, log analysis, and endpoint monitoring.
- To analyze endpoint activity and generate security reports using Wazuh.
- To complete hands-on labs that reinforce EDR concepts and provide practical skills in securing endpoints and responding to threats.

1.3 Requirements

To successfully complete this task, the following tools and resources were used:

- **EDR Solution**: Wazuh (Server and Agent components)
- **Virtualization Software**: VirtualBox (or VMware) for simulating endpoint environments
- **Operating Systems**: Linux-based systems (e.g., Ubuntu or Kali Linux) for Wazuh server and agents
- **Lab Resources**:
    - TryHackMe Room: *Intro to Endpoint Security*
    - TryHackMe Room: *Wazuh Lab*
- **Network Connection**: For downloading software packages, updates, and accessing lab environments
- **System Resources**: Sufficient CPU, RAM, and storage to run virtual machines and Wazuh components efficiently

2.0 Introduction to EDR Technologies

Endpoint Detection and Response (EDR) technologies play a crucial role in modern cybersecurity frameworks by providing continuous monitoring and analysis of endpoint activities. These technologies are specifically designed to detect, investigate, and respond to security threats in real time, offering greater visibility into endpoint behavior and potential compromise.

2.1 Importance of EDR in Cybersecurity

As cyber threats become more sophisticated and targeted, traditional security tools often fall short in identifying advanced attacks. EDR fills this gap by delivering a more proactive and intelligent approach to securing endpoints. Key benefits of EDR include:

- **Real-Time Monitoring:** EDR solutions continuously monitor endpoint activities to detect suspicious behavior, providing visibility that traditional antivirus solutions lack.

- **Threat Detection and Alerting:** By analyzing behavioral patterns and system anomalies, EDR tools can identify potential threats, such as ransomware, fileless malware, or unauthorized access attempts.

- **Incident Response:** EDR platforms enable rapid investigation and remediation of incidents, reducing response times and minimizing the impact of breaches.

- **Forensic Capabilities:** EDR logs and records endpoint activity, allowing security analysts to perform root-cause analysis and understand the scope of an attack.

- **Integration with SIEM and SOAR:** EDR tools often integrate with broader security solutions to automate detection and response workflows, contributing to a more holistic security architecture.

2.2 Core Features of EDR Solutions

Modern EDR technologies share a set of common features that make them indispensable in threat defense. These include:

- **Behavioral Monitoring:** Tracks how processes behave over time to detect abnormal activity, even in the absence of known malware signatures.

- **Log Collection and Analysis:** Aggregates system and application logs to correlate events and identify indicators of compromise (IOCs).
- **Threat Intelligence Integration:** Enriches detections with threat intelligence feeds to provide context and improve accuracy.
- **Automated Response Actions:** Provides the ability to isolate infected systems, terminate malicious processes, or remove unauthorized users automatically.
- **Dashboards and Reporting:** Offers visualization tools and dashboards to help analysts understand endpoint activities and generate reports for compliance or investigation.

2.3 Role of EDR Within a Broader Security Architecture

EDR technologies function as a vital layer in a **defense-in-depth** strategy. While firewalls and network-based security solutions provide perimeter defense, EDR ensures that internal endpoints are continuously protected, especially in decentralized environments with remote users and bring-your-own-device (BYOD) policies.

When combined with **Security Information and Event Management (SIEM)**, **Intrusion Detection Systems (IDS)**, and **Security Orchestration, Automation, and Response (SOAR)** tools, EDR enhances the overall threat detection and incident response capabilities of an organization.

3.0 Wazuh as the Chosen EDR Solution

Wazuh is an open-source security platform that combines the capabilities of SIEM and EDR to provide real-time visibility, threat detection, and response across endpoints and networks. Its flexibility, community support, and robust features make it a strong candidate for securing enterprise environments.

3.1 Overview of Wazuh

Wazuh is built on top of OSSEC (an open-source Host-based Intrusion Detection System) and extends its functionality with advanced features such as log analysis, vulnerability detection, file integrity monitoring, and cloud security. It offers a centralized management interface and supports integration with various platforms and tools like Elastic Stack, AWS, Docker, and more.

3.2 Key Features of Wazuh as an EDR Tool

- **Threat Detection:**
  Wazuh continuously monitors endpoints for suspicious behavior using a rule-based detection engine. It can detect brute-force attacks, malware activity, privilege escalation attempts, and more.

- **Log Collection and Analysis:**
  It collects system, application, and security logs from endpoints and servers. These logs are analyzed for anomalies and correlated to identify potential threats.

- **Real-Time Monitoring:**
  Wazuh agents installed on endpoints send data to a central server where it is analyzed in real-time, enabling quick detection of threats and automated responses.

- **File Integrity Monitoring (FIM):**
  Wazuh monitors critical system files and directories for unauthorized changes, which is essential for identifying tampering or unauthorized access.

- **Rootkit Detection:**
  It can detect rootkits and other forms of stealth malware that aim to hide their presence from traditional monitoring tools.

- **Vulnerability Detection:**
  Wazuh scans installed applications and operating systems to identify known vulnerabilities using vulnerability databases such as NVD (National Vulnerability Database).

- **Compliance Auditing:**
  Built-in compliance modules help assess and enforce regulatory compliance (e.g., GDPR, PCI DSS, HIPAA) across endpoints.

3.3 Wazuh vs Other EDR Tools

Feature	Wazuh	Commercial EDRs (e.g., CrowdStrike, SentinelOne)
Cost	Free and open-source	Commercial, requires subscription
Customization	Highly customizable	Limited customization in managed solutions
Integration	Elastic Stack, AWS, etc.	Proprietary dashboards and integrations
Community Support	Active open-source community	Vendor-based support
Ease of Use	Moderate (requires setup)	User-friendly with professional onboarding
Advanced Response	Basic automation	Advanced AI/ML-based response systems

While Wazuh may lack some of the advanced AI/ML-powered features found in premium EDR platforms, it stands out as a highly capable and cost-effective solution, especially for small to mid-sized organizations or training environments.

4.0 Set Up a Virtual Lab Environment

In order to gain hands-on experience with Endpoint Detection and Response (EDR) technologies, a virtual lab environment was set up to simulate a realistic deployment of Wazuh. This environment enabled testing of various Wazuh functionalities, including log collection, threat detection, and endpoint monitoring. The virtual lab was built using VMware Fusion on a MacBook Pro, and included two Ubuntu-based virtual machines representing both the server (manager) and client (agent) components of a Wazuh deployment.

4.1 Virtual Environment Setup

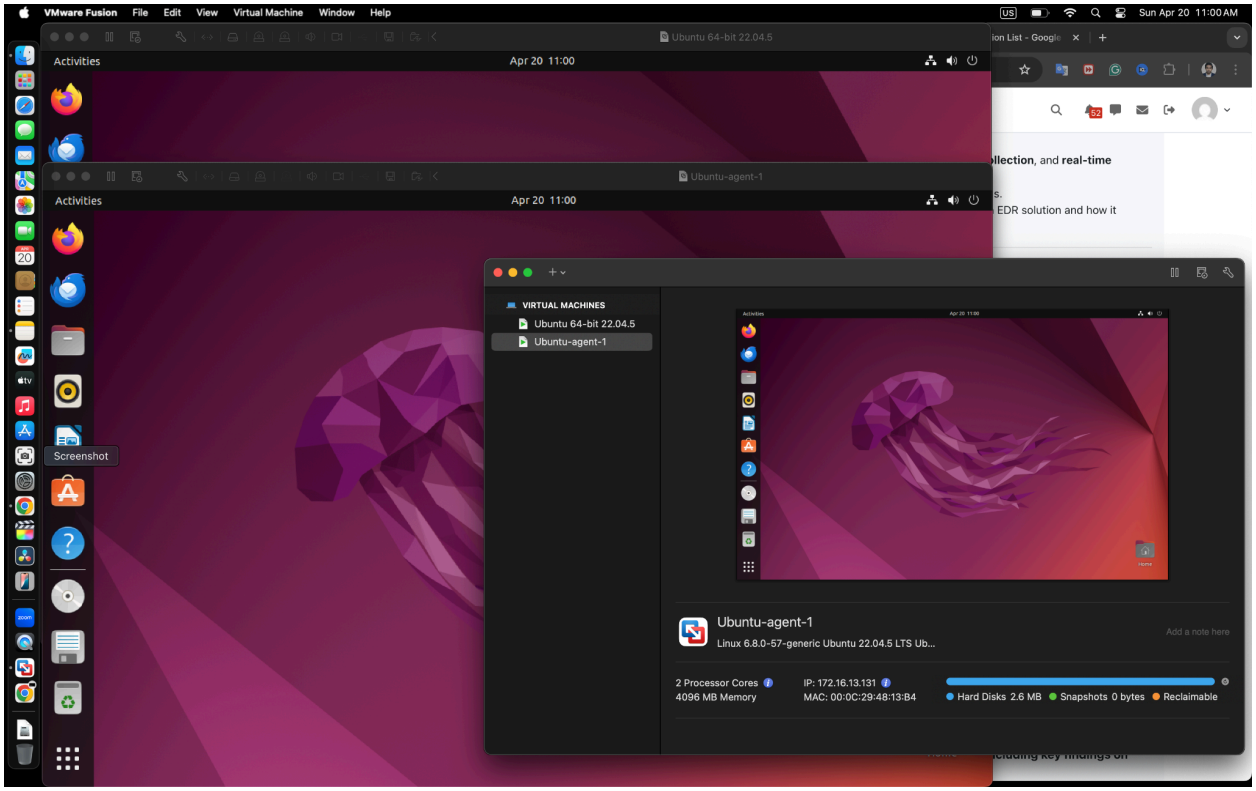
The first step in the process involved creating the virtual machines necessary to build the lab infrastructure. VMware Fusion was used as the virtualization platform due to its performance and compatibility with macOS.

Two Ubuntu 22.04 LTS virtual machines were created with the following roles:

- **VM1 – Wazuh Manager**: This machine acts as the Wazuh server, responsible for managing agents, collecting and analyzing logs, and presenting data through the Wazuh dashboard.

- **VM2 – Wazuh Agent**: This machine serves as a monitored endpoint. It runs the Wazuh agent, which collects system data and sends it to the manager for analysis.

Each virtual machine was allocated sufficient resources (2 CPU cores, 4 GB RAM, and 20 GB disk space) and configured to operate on the same internal network to ensure proper communication. Static IP addresses or consistent internal addressing via NAT or host-only networking were used to facilitate stable communication between the machines.



4.2 Wazuh Manager Installation and Configuration (VM1)

The Wazuh Manager was installed on VM1 using the official Wazuh APT repository. The following steps were performed:

System Update:
All packages were updated to ensure compatibility and stability:

```
sudo apt update && sudo apt upgrade -y
```

1. **Import Wazuh GPG Key**:
   ```
   curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo gpg --dearmor -o /usr/share/keyrings/wazuh.gpg
   ```
2. **Add Wazuh Repository**:
   ```
   echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
   ```

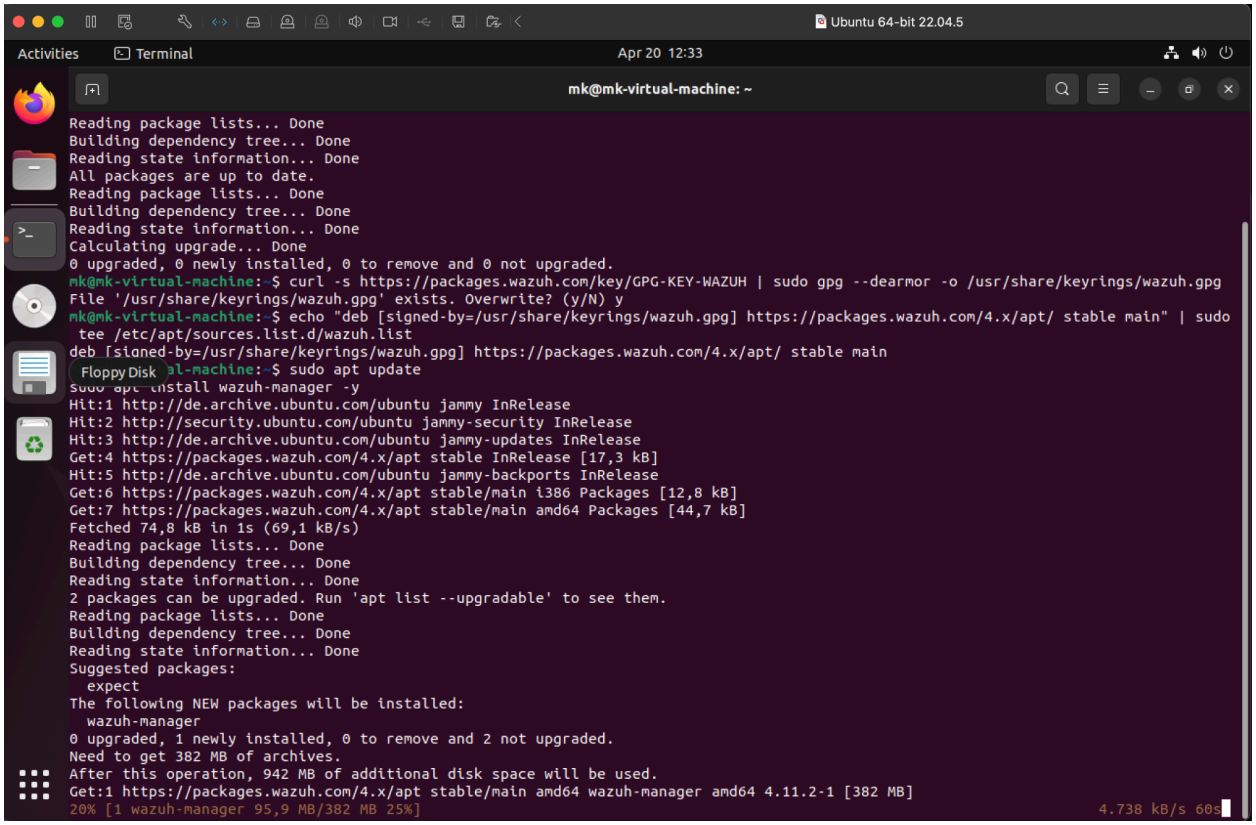
```
https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee
/etc/apt/sources.list.d/wazuh.list
```

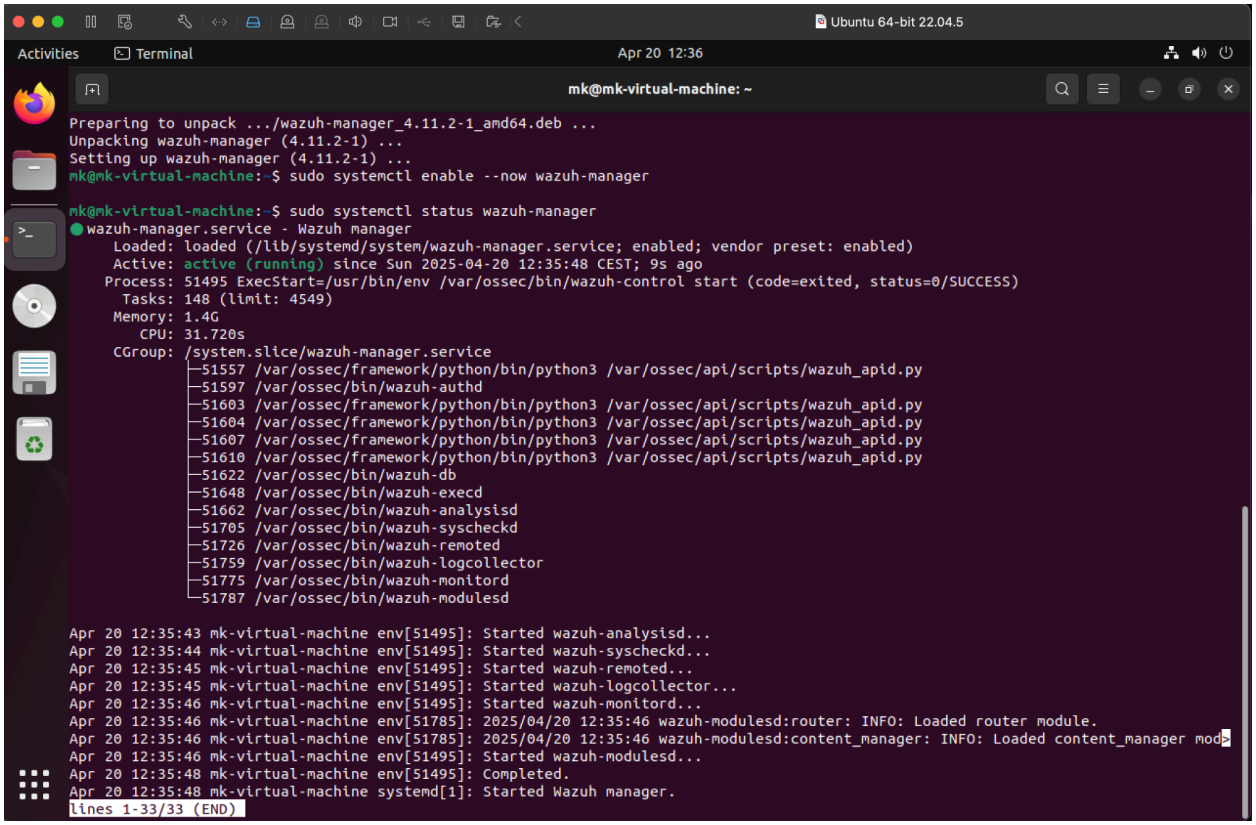
3. **Install Wazuh Manager**:

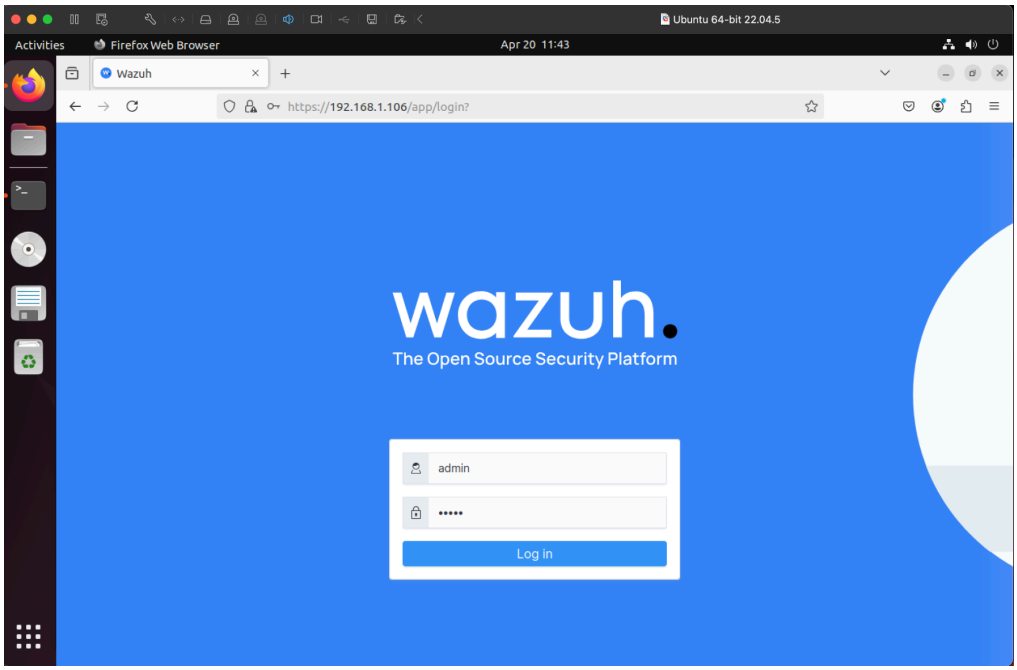
```
sudo apt update

sudo apt install wazuh-manager -y
```

4. **Start and Enable the Service**:

```
sudo systemctl enable --now wazuh-manager

sudo systemctl status wazuh-manager
```




Ubuntu 64-bit 22.04.5
Activities
Terminal
Apr 20 12:36
mk@mk-virtual-machine: ~
Preparing to unpack .../wazuh-manager_4.11.2-1_amd64.deb ...
Unpacking wazuh-manager (4.11.2-1) ...
Setting up wazuh-manager (4.11.2-1) ...
mk@mk-virtual-machine:~$ sudo systemctl enable --now wazuh-manager
mk@mk-virtual-machine:~$ sudo systemctl status wazuh-manager
wazuh-manager.service - Wazuh manager
Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
Active: active (running) since Sun 2025-04-20 12:35:48 CEST; 9s ago
Process: 51495 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
Tasks: 148 (limit: 4549)
Memory: 1.4G
CPU: 31.720s
CGroup: /system.slice/wazuh-manager.service
├─51557 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
├─51597 /var/ossec/bin/wazuh-authd
├─51603 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
├─51604 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
├─51607 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
├─51610 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
├─51622 /var/ossec/bin/wazuh-db
├─51648 /var/ossec/bin/wazuh-execd
├─51662 /var/ossec/bin/wazuh-analysisd
├─51705 /var/ossec/bin/wazuh-syscheckd
├─51726 /var/ossec/bin/wazuh-remoted
├─51759 /var/ossec/bin/wazuh-logcollector
├─51775 /var/ossec/bin/wazuh-monitord
└─51787 /var/ossec/bin/wazuh-modulesd
Apr 20 12:35:43 mk-virtual-machine env[51495]: Started wazuh-analysisd...
Apr 20 12:35:44 mk-virtual-machine env[51495]: Started wazuh-syscheckd...
Apr 20 12:35:45 mk-virtual-machine env[51495]: Started wazuh-remoted...
Apr 20 12:35:45 mk-virtual-machine env[51495]: Started wazuh-logcollector...
Apr 20 12:35:46 mk-virtual-machine env[51495]: Started wazuh-monitord...
Apr 20 12:35:46 mk-virtual-machine env[51785]: 2025/04/20 12:35:46 wazuh-modulesd:router: INFO: Loaded router module.
Apr 20 12:35:46 mk-virtual-machine env[51785]: 2025/04/20 12:35:46 wazuh-modulesd:content_manager: INFO: Loaded content_manager mod
Apr 20 12:35:46 mk-virtual-machine env[51495]: Started wazuh-modulesd...
Apr 20 12:35:48 mk-virtual-machine env[51495]: Completed.
Apr 20 12:35:48 mk-virtual-machine systemd[1]: Started Wazuh manager.
lines 1-33/33 (END)


Ubuntu 64-bit 22.04.5
Activities
Firefox Web Browser
Apr 20 11:43
Wazuh
https://192.168.1.106/app/login?
wazuh.
The Open Source Security Platform
admin
Log in

4.3 Wazuh Agent Installation and Configuration (VM2)

The Wazuh agent was installed on the second virtual machine to simulate an endpoint that would be monitored by the manager.

System Update:

```
sudo apt update && sudo apt upgrade -y
```

1. **Import GPG Key and Add Repository**:
   (Same commands as used for the manager above.)

Install the Wazuh Agent:

```
sudo apt install wazuh-agent -y
```

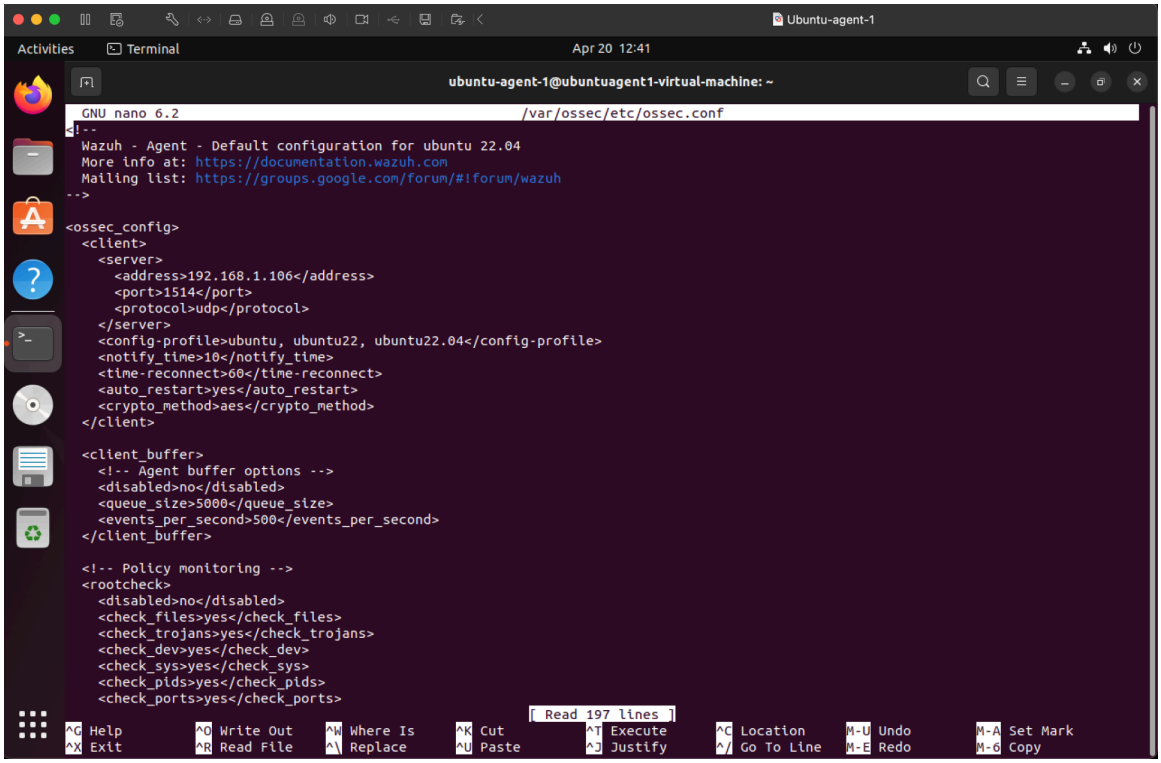
2. **Configure the Agent**: The agent was configured to communicate with the manager. The configuration file was edited:
   `sudo nano /var/ossec/etc/ossec.conf`

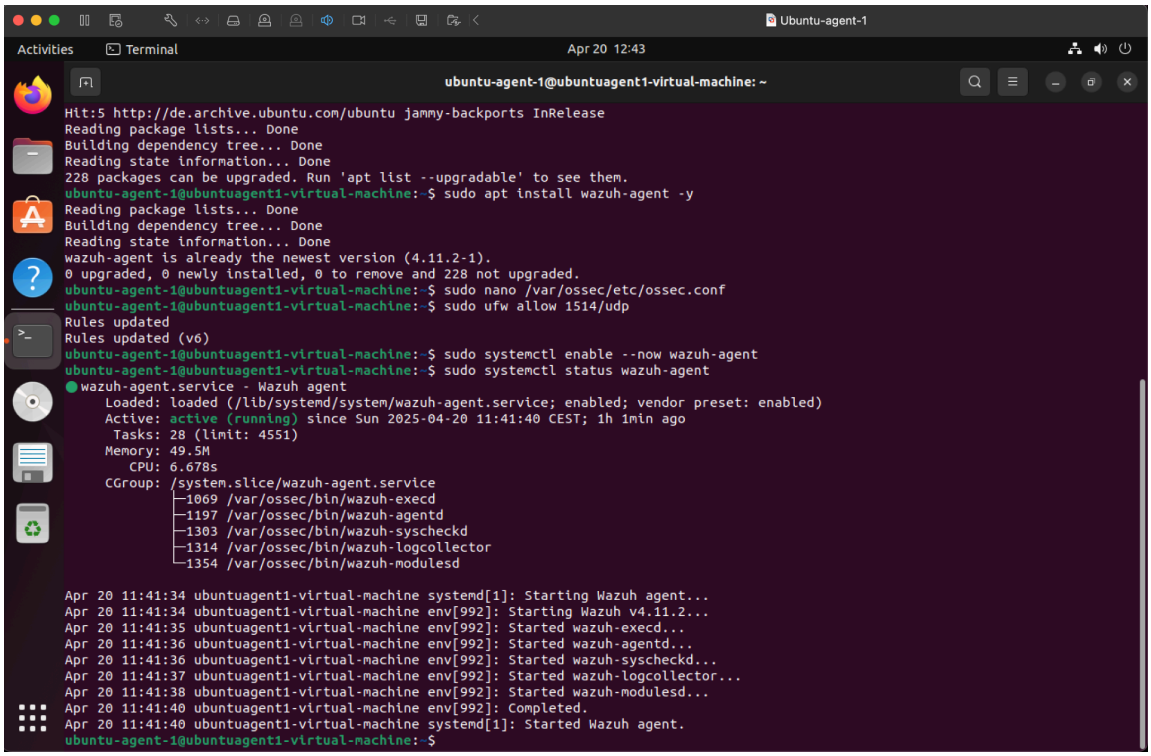
The following block was added/modified within the `<client>` section:

```
<client>
  <server>
    <address>192.168.1.106</address>
    <port>1514</port>
    <protocol>udp</protocol>
  </server>
</client>
```

3. **Enable and Start the Agent**:

```
sudo systemctl enable --now wazuh-agent

sudo systemctl status wazuh-agent
```


Ubuntu-agent-1
Activities
Terminal
Apr 20 12:41
ubuntu-agent-1@ubuntuagent1-virtual-machine: ~
GNU nano 6.2 /var/ossec/etc/ossec.conf
<!--
Wazuh - Agent - Default configuration for ubuntu 22.04
More info at: https://documentation.wazuh.com
Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->
<ossec_config>
<client>
<server>
<address>192.168.1.106</address>
<port>1514</port>
<protocol>udp</protocol>
</server>
<config-profile>ubuntu, ubuntu22, ubuntu22.04</config-profile>
<notify_time>10</notify_time>
<time-reconnect>60</time-reconnect>
<auto_restart>yes</auto_restart>
<crypto_method>aes</crypto_method>
</client>
<client_buffer>
<!-- Agent buffer options -->
<disabled>no</disabled>
<queue_size>5000</queue_size>
<events_per_second>500</events_per_second>
</client_buffer>
<!-- Policy monitoring -->
<rootcheck>
<disabled>no</disabled>
<check_files>yes</check_files>
<check_trojans>yes</check_trojans>
<check_dev>yes</check_dev>
<check_sys>yes</check_sys>
<check_pids>yes</check_pids>
<check_ports>yes</check_ports>
[Read 197 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy


Ubuntu-agent-1
Activities
Terminal
Apr 20 12:43
ubuntu-agent-1@ubuntuagent1-virtual-machine: ~
Hit:5 http://de.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
228 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu-agent-1@ubuntuagent1-virtual-machine:~$ sudo apt install wazuh-agent -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wazuh-agent is already the newest version (4.11.2-1).
0 upgraded, 0 newly installed, 0 to remove and 228 not upgraded.
ubuntu-agent-1@ubuntuagent1-virtual-machine:~$ sudo nano /var/ossec/etc/ossec.conf
ubuntu-agent-1@ubuntuagent1-virtual-machine:~$ sudo ufw allow 1514/udp
Rules updated
Rules updated (v6)
ubuntu-agent-1@ubuntuagent1-virtual-machine:~$ sudo systemctl enable --now wazuh-agent
ubuntu-agent-1@ubuntuagent1-virtual-machine:~$ sudo systemctl status wazuh-agent
wazuh-agent.service - Wazuh agent
Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; vendor preset: enabled)
Active: active (running) since Sun 2025-04-20 11:41:40 CEST; 1h 1min ago
Tasks: 28 (limit: 4551)
Memory: 49.5M
CPU: 6.678s
CGroup: /system.slice/wazuh-agent.service
├─1069 /var/ossec/bin/wazuh-execd
├─1197 /var/ossec/bin/wazuh-agentd
├─1303 /var/ossec/bin/wazuh-syscheckd
├─1314 /var/ossec/bin/wazuh-logcollector
└─1354 /var/ossec/bin/wazuh-modulesd
Apr 20 11:41:34 ubuntuagent1-virtual-machine systemd[1]: Starting Wazuh agent...
Apr 20 11:41:34 ubuntuagent1-virtual-machine env[992]: Starting Wazuh v4.11.2...
Apr 20 11:41:35 ubuntuagent1-virtual-machine env[992]: Started wazuh-execd...
Apr 20 11:41:36 ubuntuagent1-virtual-machine env[992]: Started wazuh-agentd...
Apr 20 11:41:36 ubuntuagent1-virtual-machine env[992]: Started wazuh-syscheckd...
Apr 20 11:41:37 ubuntuagent1-virtual-machine env[992]: Started wazuh-logcollector...
Apr 20 11:41:38 ubuntuagent1-virtual-machine env[992]: Started wazuh-modulesd...
Apr 20 11:41:40 ubuntuagent1-virtual-machine env[992]: Completed.
Apr 20 11:41:40 ubuntuagent1-virtual-machine systemd[1]: Started Wazuh agent.
ubuntu-agent-1@ubuntuagent1-virtual-machine:~$

This virtual environment is now ready to be used in the next stages of the project for performing simulated attacks, monitoring endpoint behavior, and generating reports through the Wazuh dashboard.

5.0 Basic Configuration of EDR

This section focuses on configuring the Wazuh platform to enable core EDR functionalities such as threat detection, log collection, and behavioral monitoring. Additionally, it includes simulating basic security incidents to observe Wazuh's response in real-time.

5.1 Environment and Platform Note

While the original plan was to configure Wazuh directly within a local virtual environment on a MacStudio system, compatibility issues were encountered due to **Wazuh**'s current limitations with **Apple Silicon architecture**. As a result, a practical alternative was chosen using the **TryHackMe platform**, specifically within the **Wazuh Lab** room. This online virtual environment provides a pre-configured instance of Wazuh and supports hands-on exploration of its security features.

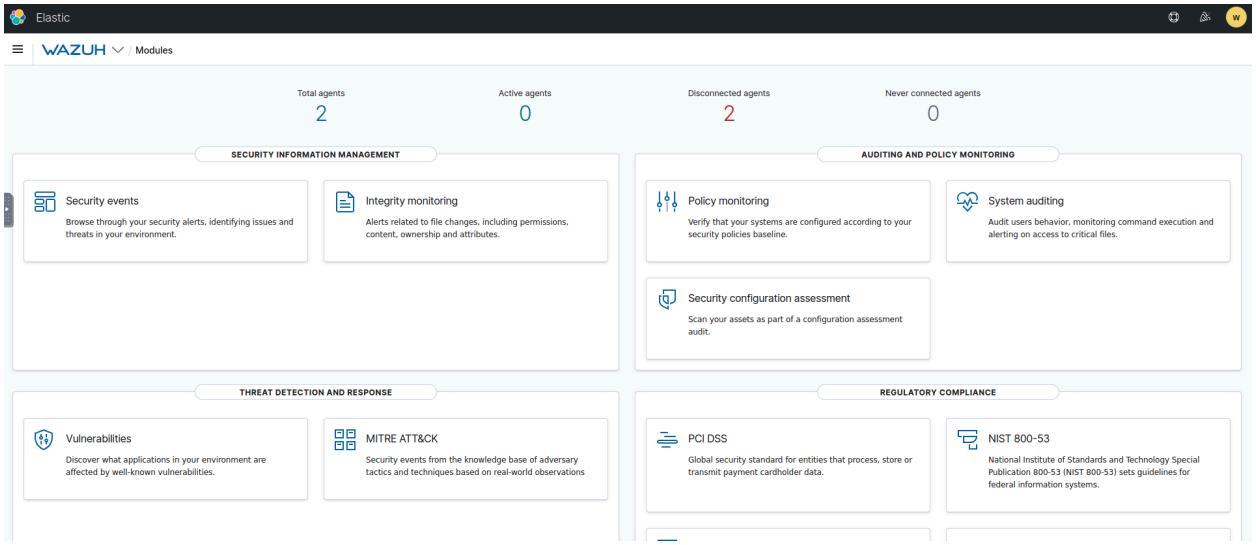
It is important to note that the TryHackMe environment uses pre-generated logs and activities for demonstration purposes. The data and detection events are from the year **2022**, as provided within the lab instance.

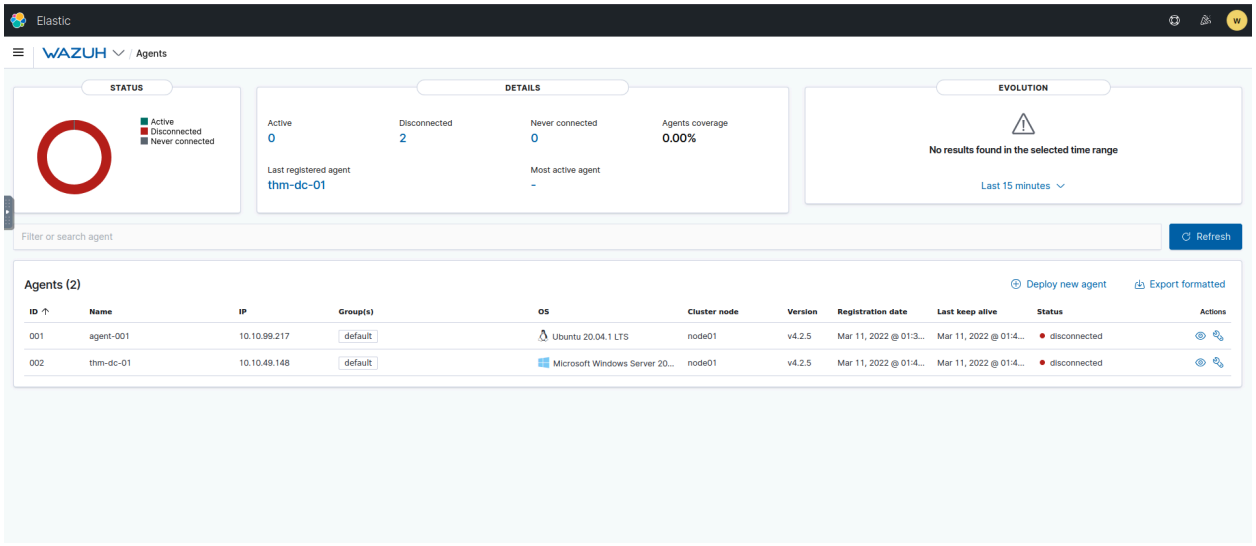
Despite this limitation, the lab effectively demonstrates the key features and capabilities of Wazuh in identifying and responding to endpoint threats, which aligns with the learning objectives of this internship task.

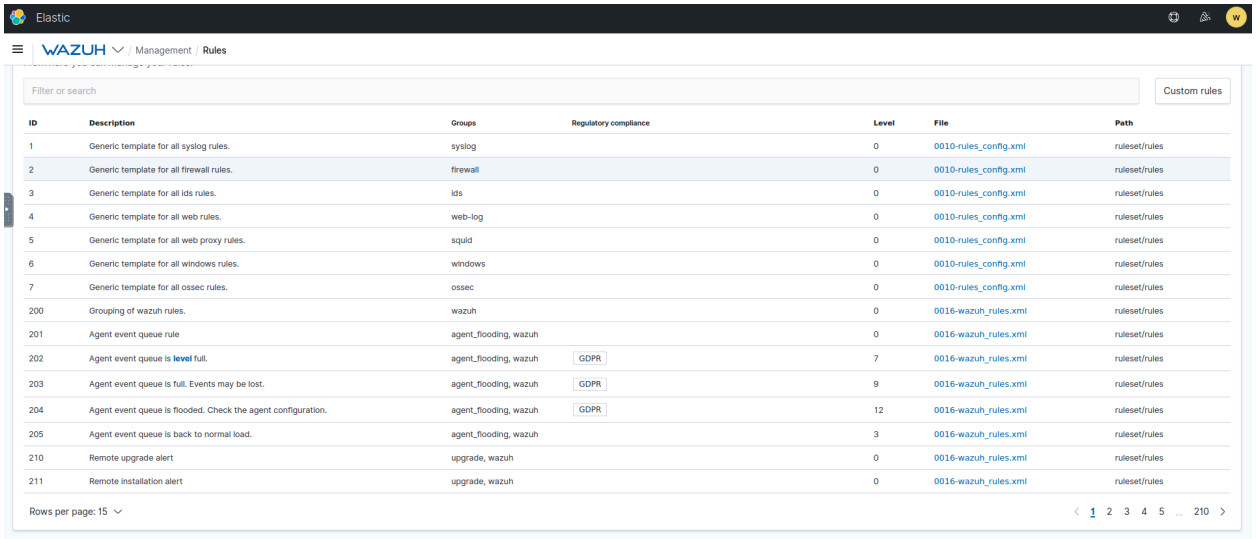
5.2 Configuring Wazuh for Threat Detection and Log Collection

In the TryHackMe Wazuh Lab, Wazuh was pre-configured to collect logs from monitored endpoints and detect threats based on predefined rules and decoders. The following configurations were verified and explored:

- **File Integrity Monitoring (FIM):** Automatically detects unauthorized changes to critical system files.
- **Authentication Log Monitoring:** Captures failed login attempts and potential brute-force attacks from `/var/log/auth.log`.
- **Rootcheck and System Auditing:** Identifies hidden processes, malware presence, and insecure configurations.







ID	Description	Groups	Regulatory compliance	Level	File	Path
1	Generic template for all syslog rules.	syslog		0	0010-rules_config.xml	ruleset/rules
2	Generic template for all firewall rules.	firewall		0	0010-rules_config.xml	ruleset/rules
3	Generic template for all ids rules.	ids		0	0010-rules_config.xml	ruleset/rules
4	Generic template for all web rules.	web-log		0	0010-rules_config.xml	ruleset/rules
5	Generic template for all web proxy rules.	squid		0	0010-rules_config.xml	ruleset/rules
6	Generic template for all windows rules.	windows		0	0010-rules_config.xml	ruleset/rules
7	Generic template for all ossec rules.	ossec		0	0010-rules_config.xml	ruleset/rules
200	Grouping of wazuh rules.	wazuh		0	0016-wazuh_rules.xml	ruleset/rules
201	Agent event queue rule	agent_flooding, wazuh		0	0016-wazuh_rules.xml	ruleset/rules
202	Agent event queue is level full.	agent_flooding, wazuh	GDPR	7	0016-wazuh_rules.xml	ruleset/rules
203	Agent event queue is full. Events may be lost.	agent_flooding, wazuh	GDPR	9	0016-wazuh_rules.xml	ruleset/rules
204	Agent event queue is flooded. Check the agent configuration.	agent_flooding, wazuh	GDPR	12	0016-wazuh_rules.xml	ruleset/rules
205	Agent event queue is back to normal load.	agent_flooding, wazuh		3	0016-wazuh_rules.xml	ruleset/rules
210	Remote upgrade alert	upgrade, wazuh		0	0016-wazuh_rules.xml	ruleset/rules
211	Remote installation alert	upgrade, wazuh		0	0016-wazuh_rules.xml	ruleset/rules

Rows per page: 15 1 2 3 4 5 ... 210

5.3 Simulating Security Incidents

Several simulated activities were performed or reviewed in the lab to demonstrate Wazuh's incident detection capabilities:

1. Unauthorized Login Attempts

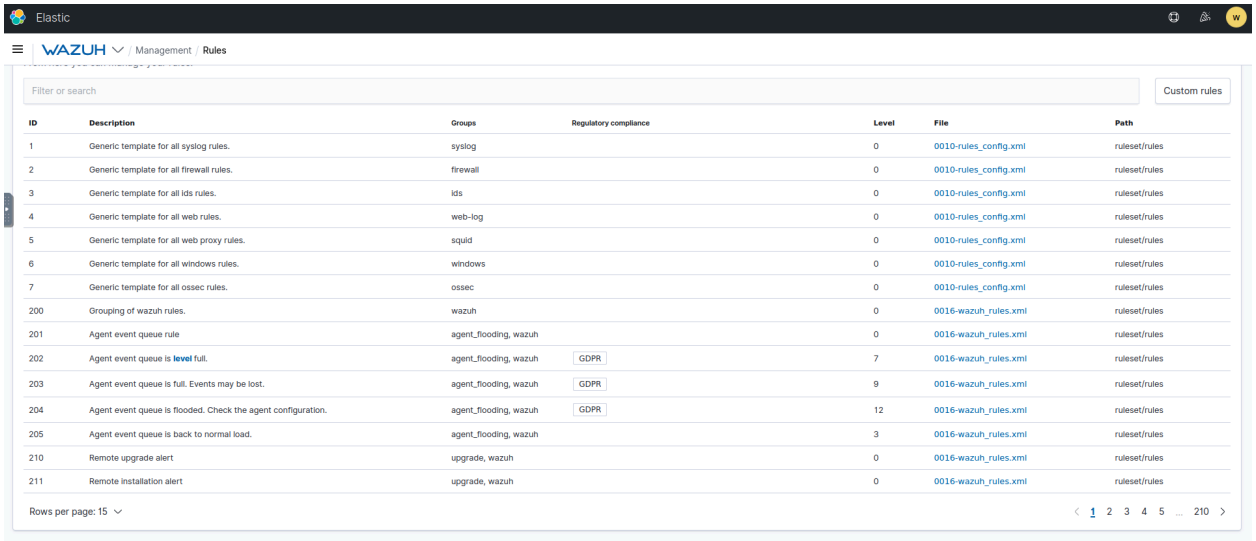
Simulated brute-force SSH login attempts were triggered, which Wazuh detected via authentication log analysis. These were visible as high-severity alerts in the dashboard.

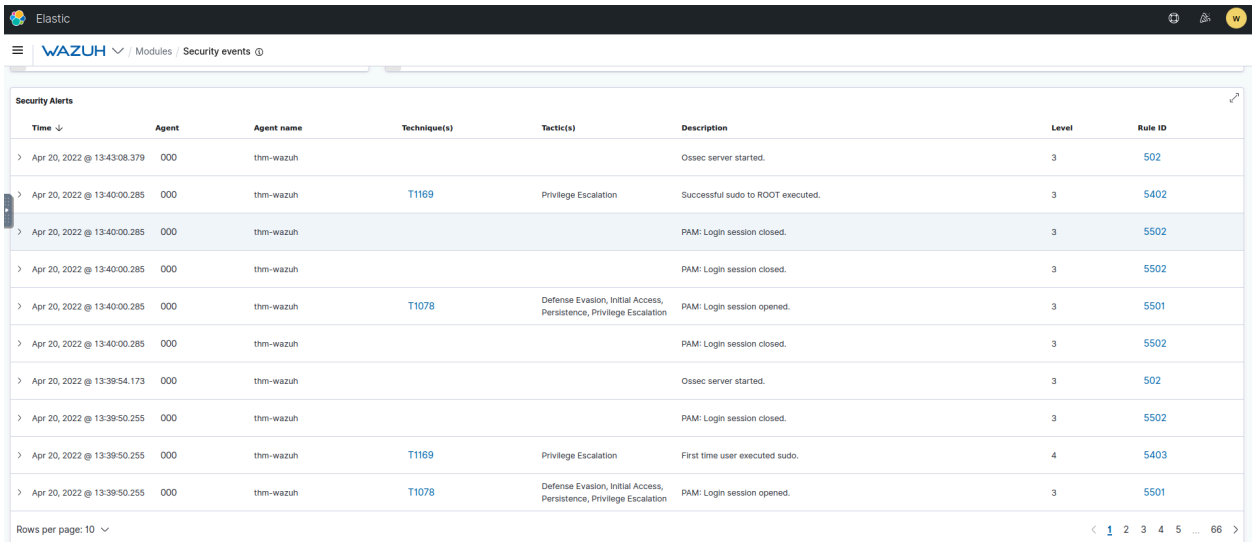
2. File Modification Events

Modifying sensitive files such as `/etc/hostname` triggered file integrity monitoring alerts. These alerts help identify unauthorized tampering with system configurations.

3 Suspicious Script Creation

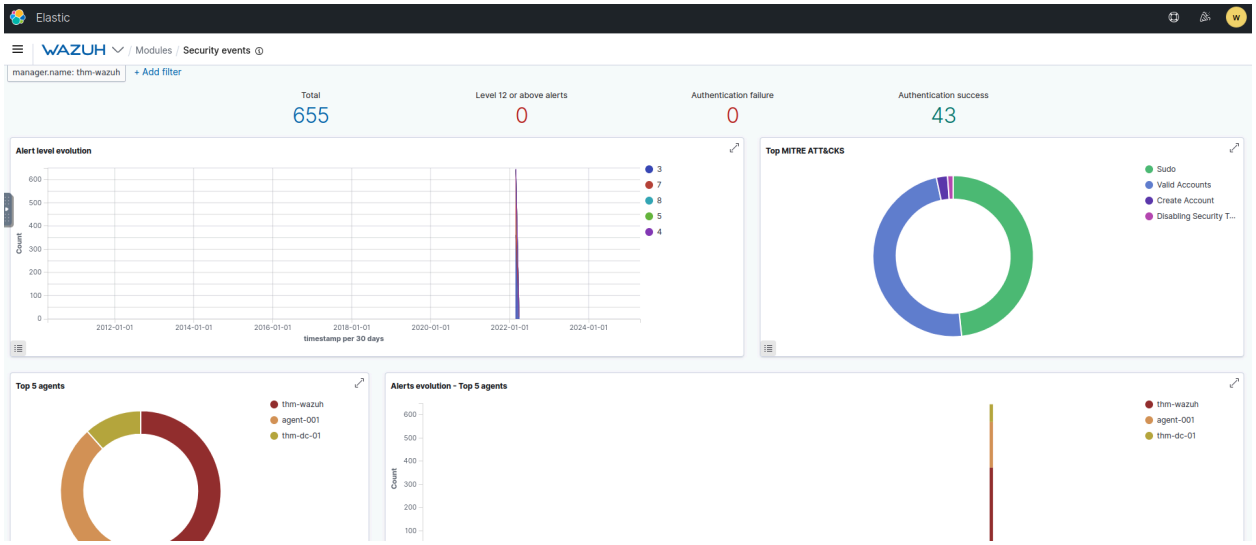
A simulated malicious script was created and executed within the endpoint. Wazuh flagged the activity based on behavioral analysis and command execution monitoring.

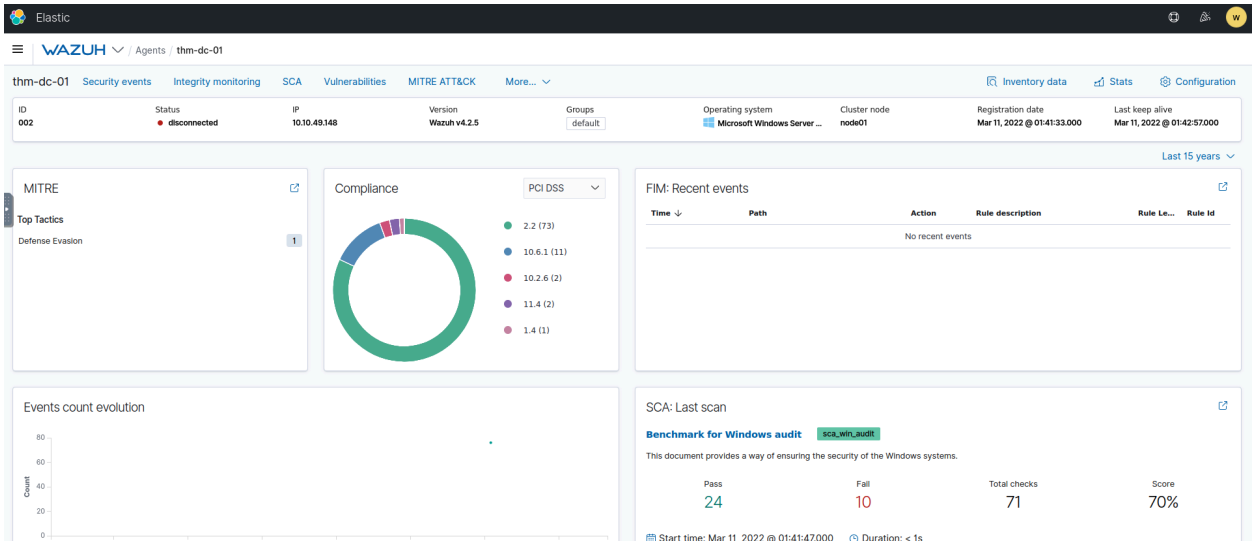




5.4 Observing Wazuh's Response

All incidents were logged and categorized by severity in the **Security Events** and **Rules** section of the Wazuh dashboard. The alerts included relevant metadata such as timestamp, source agent, rule ID, and a brief description of the threat.





Although local testing was not fully possible due to hardware limitations, the TryHackMe Wazuh Lab successfully demonstrated the fundamental configuration and response capabilities of Wazuh as an EDR solution. Through simulated incidents, Wazuh proved effective in identifying unauthorized activities, alerting the administrator, and providing meaningful insights for incident response.

6.0 Endpoint Visibility and Reporting

This section focuses on leveraging Wazuh's powerful monitoring and reporting capabilities to analyze endpoint activities, such as file integrity changes and process executions. The goal is to understand how Wazuh provides real-time visibility into endpoint behavior and supports informed decision-making through its built-in reporting features.

6.1 Environment Note: Apple Silicon Compatibility Limitation

Due to the **Apple Silicon (M1/M2)** architecture of the local MacStudio system, there were compatibility issues running the full Wazuh stack within a virtualized environment (e.g., VirtualBox or VMware Fusion or UTM). At the time of this report, Wazuh's official support and Docker images are optimized for x86_64 systems and may not function correctly on ARM-based Macs without advanced configuration or performance trade-offs.

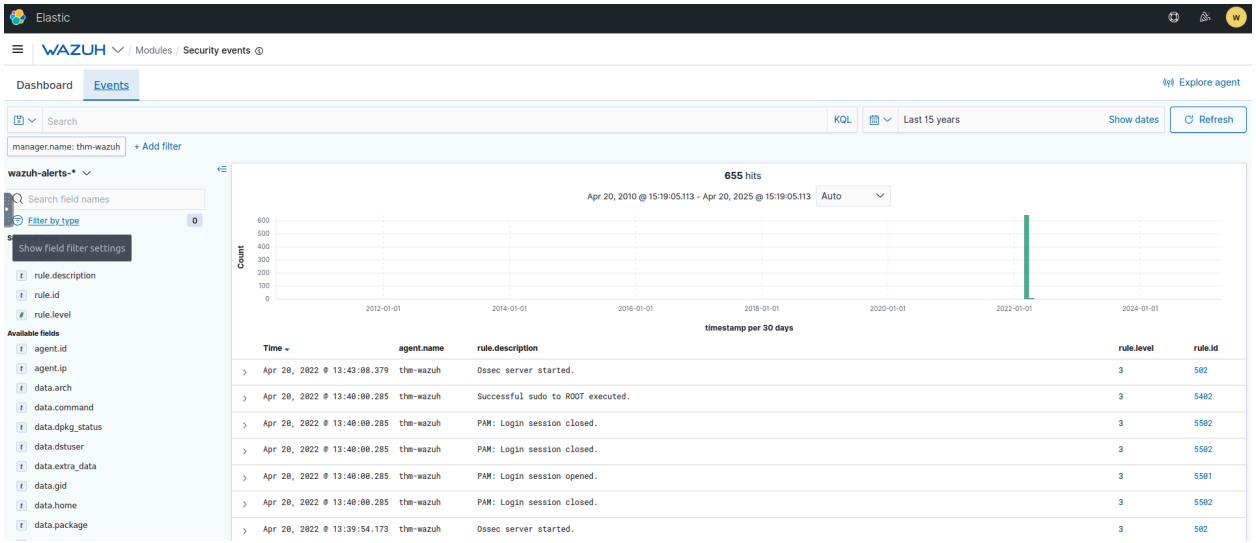
To work around this issue and complete the practical requirements of this task, the **TryHackMe Wazuh Lab** environment was utilized. This online platform provides a fully functional and pre-configured Wazuh setup, allowing for secure testing and demonstration of endpoint monitoring features.

6.2 File Integrity Monitoring (FIM)

Wazuh provides robust **File Integrity Monitoring (FIM)** capabilities that detect changes to critical files and directories on monitored endpoints. Within the TryHackMe lab, this functionality was explored by observing FIM-generated alerts related to:

- Creation of suspicious scripts
- Modifications to system configuration files
- Deletion or tampering with log files

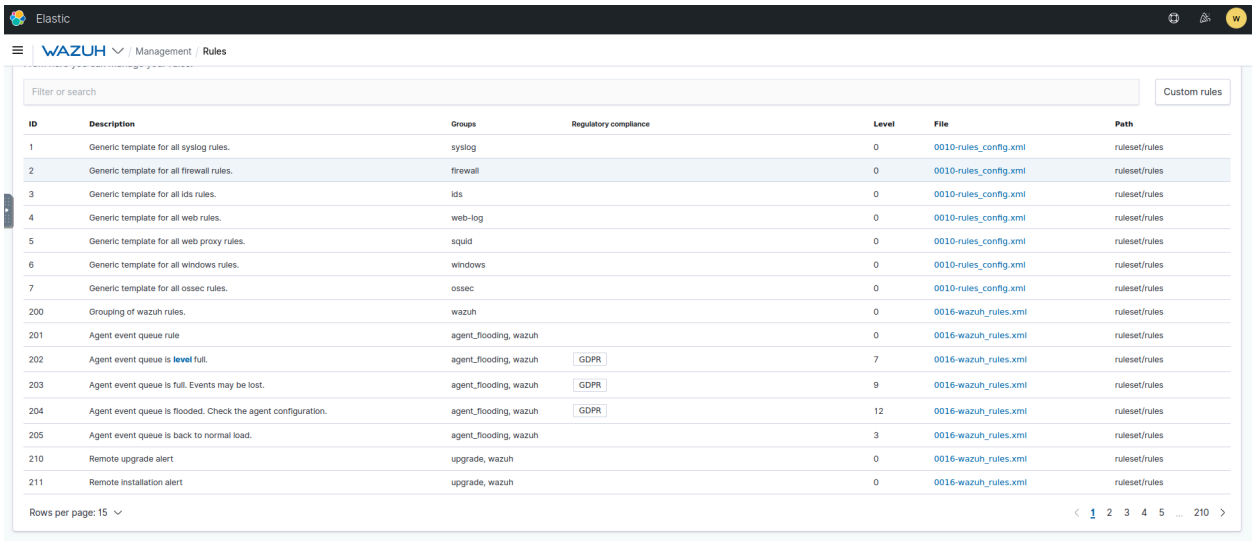
These activities were captured in real-time and presented in the **Security Events** section of the Wazuh dashboard, along with detailed metadata about the changes.



6.3 Process Monitoring

Wazuh also supports **process monitoring**, which enables visibility into processes initiated on endpoints. This is crucial for detecting malicious behavior such as unauthorized script execution or abnormal usage of administrative tools.

In the TryHackMe lab, simulated command-line executions were tracked, and Wazuh generated alerts for commands commonly associated with post-exploitation activities.

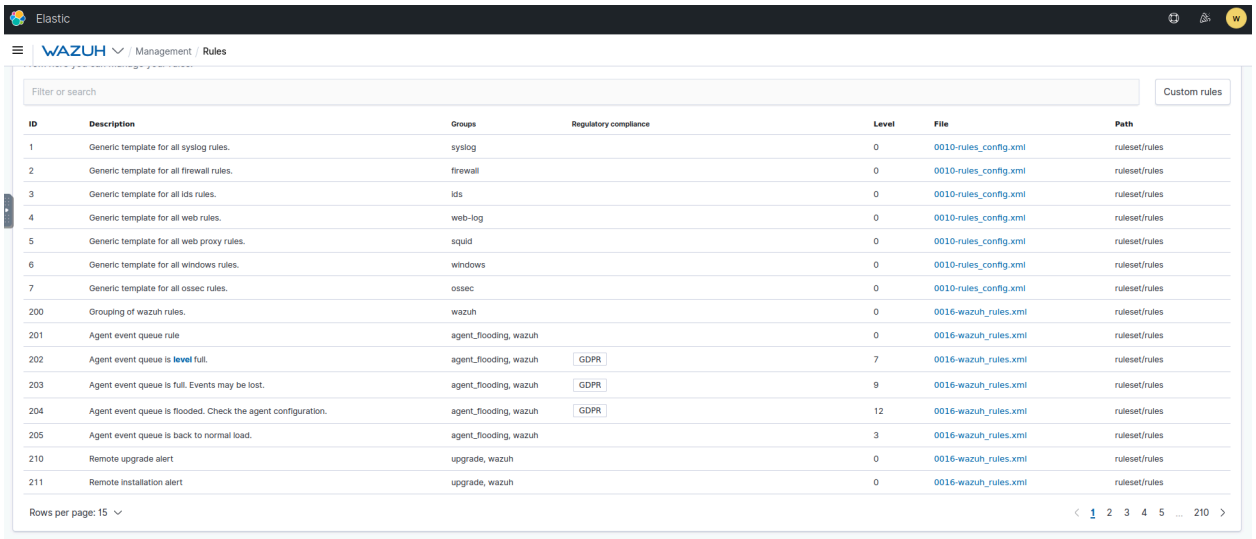


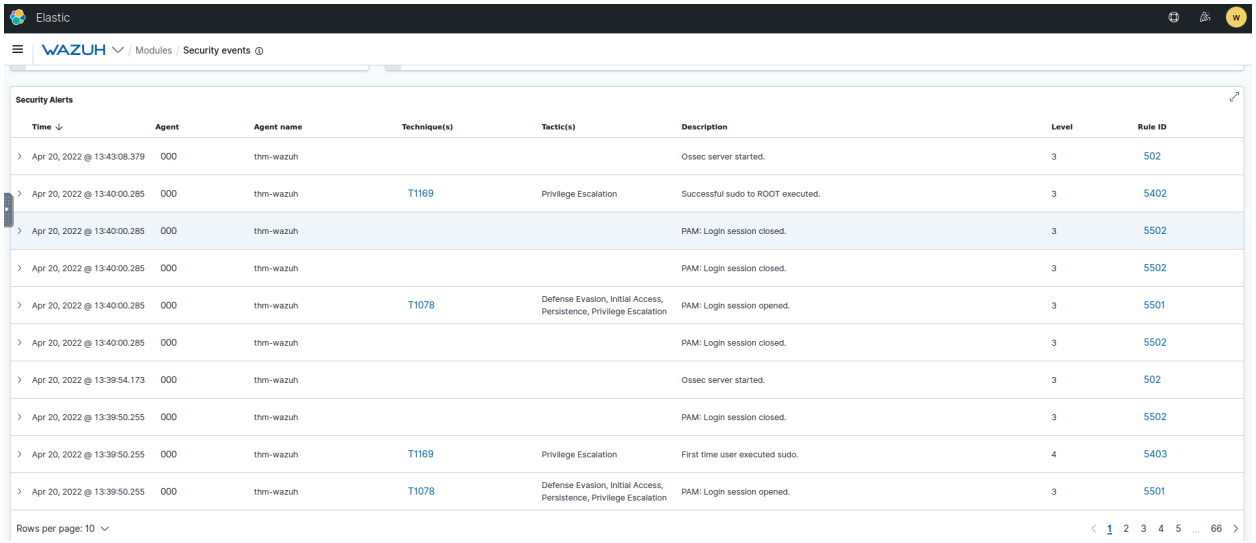
6.4 Generating Reports

The Wazuh dashboard allows users to generate visual summaries and detailed logs of detected events. During the lab, the following types of reports were generated and reviewed:

- **Overview Reports** summarizing the number and severity of recent alerts
- **Agent-specific Reports** detailing all events associated with a given monitored endpoint
- **Rule-based Reports** focusing on specific categories like authentication, file access, or malware detection

These reports are valuable for performing post-incident analysis and identifying recurring threats or patterns.





6.5 Key Findings

From the generated reports in the TryHackMe Wazuh Lab, the following insights were observed:

- High frequency of failed SSH login attempts (brute-force simulation)
- Unauthorized file modifications indicative of privilege escalation attempts
- Execution of potentially harmful scripts flagged as critical alerts
- Effective correlation of logs across multiple modules (e.g., FIM + Auth logs)

These findings demonstrate Wazuh's effectiveness in offering full-stack endpoint visibility and real-time response capabilities.

Despite hardware limitations preventing a local deployment on Apple Silicon, the simulated lab environment provided by TryHackMe successfully showcased Wazuh's ability to monitor endpoint behavior and generate actionable reports. This section has confirmed that Wazuh is a capable and transparent EDR solution, empowering security teams with insights necessary for proactive defense and incident response.

7.0 Hands-on Labs

This section aims to solidify practical knowledge of endpoint security through guided hands-on labs. By completing the assigned TryHackMe rooms, interns gain real-world experience in securing endpoints, detecting malicious activities, and configuring tools like Wazuh for threat detection and response.

To fulfill this requirement, two dedicated labs were completed using the **TryHackMe** platform:

7.1 TryHackMe Lab: Intro to Endpoint Security

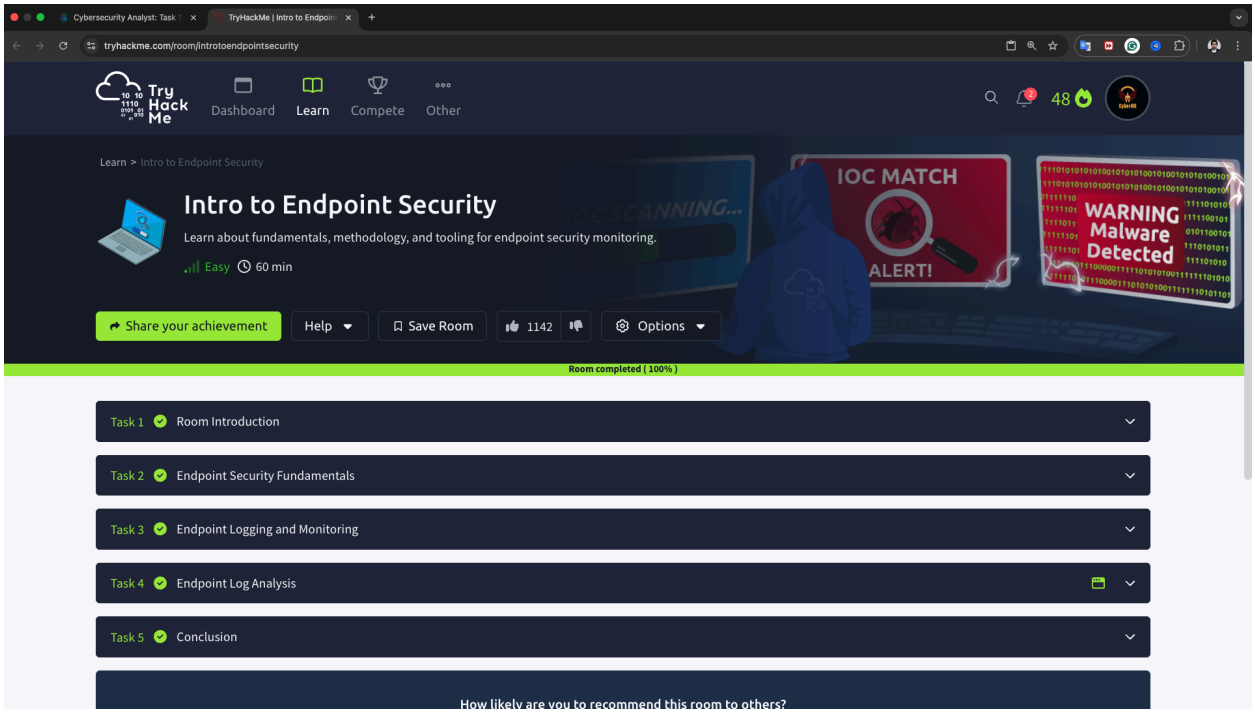
Room: *Intro to Endpoint Security*

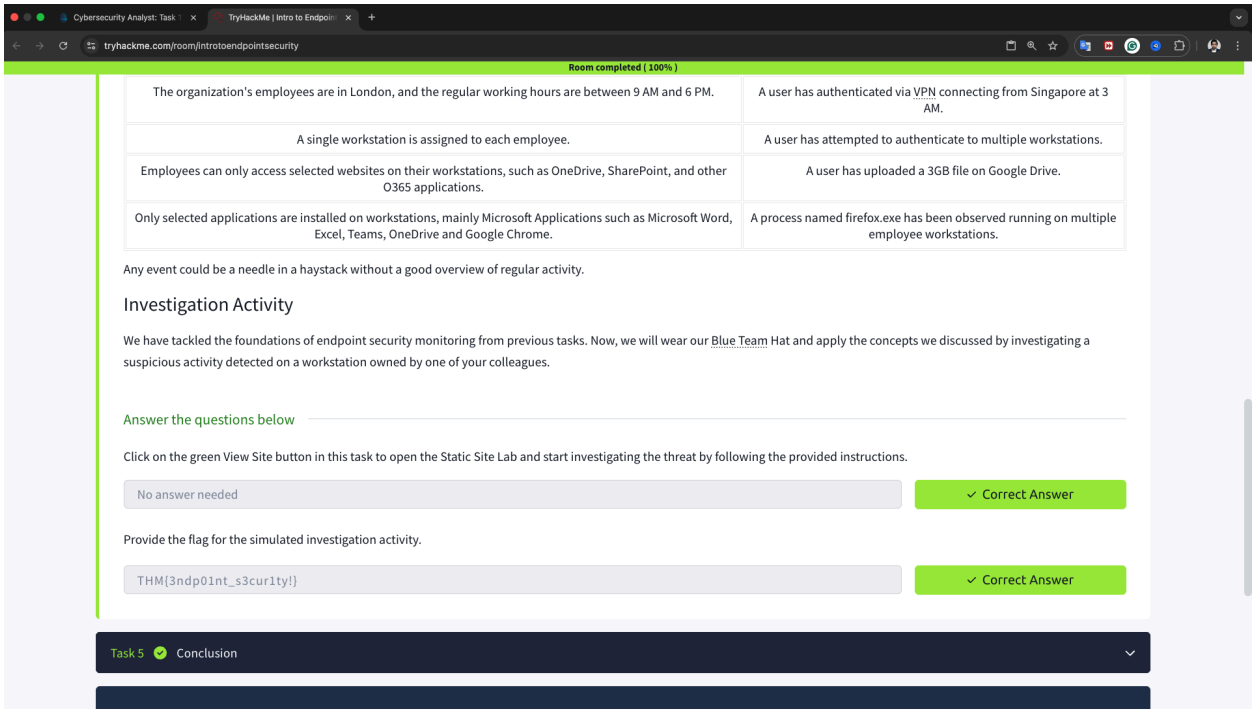
Description:
This lab provides a foundational understanding of endpoint security, including the key principles and practices necessary to protect endpoints from evolving cyber threats. It walks through:

- Common endpoint vulnerabilities
- The role of endpoint protection platforms (EPP)
- Basic hardening measures (e.g., password policies, firewall settings, antivirus software)
- Hands-on tasks using virtual endpoints to simulate attacks and apply defenses

Lab Highlights:

- Configured basic endpoint protections using pre-installed tools
- Observed the impact of disabling protection mechanisms and the security gaps they created
- Learned the importance of monitoring processes, network activity, and system changes


tryhackme.com/room/introtoendpointsecurity
Try Hack Me
Dashboard
Learn
Compete
Other
48
Learn > Intro to Endpoint Security
Intro to Endpoint Security
Learn about fundamentals, methodology, and tooling for endpoint security monitoring.
Easy
60 min
IOC MATCH
ALERT!
WARNING
Malware
Detected
Share your achievement
Help
Save Room
1142
Options
Room completed (100%)
Task 1 Room Introduction
Task 2 Endpoint Security Fundamentals
Task 3 Endpoint Logging and Monitoring
Task 4 Endpoint Log Analysis
Task 5 Conclusion
How likely are you to recommend this room to others?


tryhackme.com/room/introtoendpointsecurity
Room completed (100%)
The organization's employees are in London, and the regular working hours are between 9 AM and 6 PM.
A user has authenticated via VPN connecting from Singapore at 3 AM.
A single workstation is assigned to each employee.
A user has attempted to authenticate to multiple workstations.
Employees can only access selected websites on their workstations, such as OneDrive, SharePoint, and other O365 applications.
A user has uploaded a 3GB file on Google Drive.
Only selected applications are installed on workstations, mainly Microsoft Applications such as Microsoft Word, Excel, Teams, OneDrive and Google Chrome.
A process named firefox.exe has been observed running on multiple employee workstations.
Any event could be a needle in a haystack without a good overview of regular activity.
Investigation Activity
We have tackled the foundations of endpoint security monitoring from previous tasks. Now, we will wear our Blue Team Hat and apply the concepts we discussed by investigating a suspicious activity detected on a workstation owned by one of your colleagues.
Answer the questions below
Click on the green View Site button in this task to open the Static Site Lab and start investigating the threat by following the provided instructions.
No answer needed
Correct Answer
Provide the flag for the simulated investigation activity.
THM{3ndp01nt_s3cur1ty!}
Correct Answer
Task 5 Conclusion

7.2 TryHackMe Lab: Wazuh Lab

Room: *Wazuh Lab*

Description:
This lab focuses specifically on Wazuh as an Endpoint Detection and Response (EDR) platform. Participants work within a simulated environment to configure and test Wazuh's core functionalities:

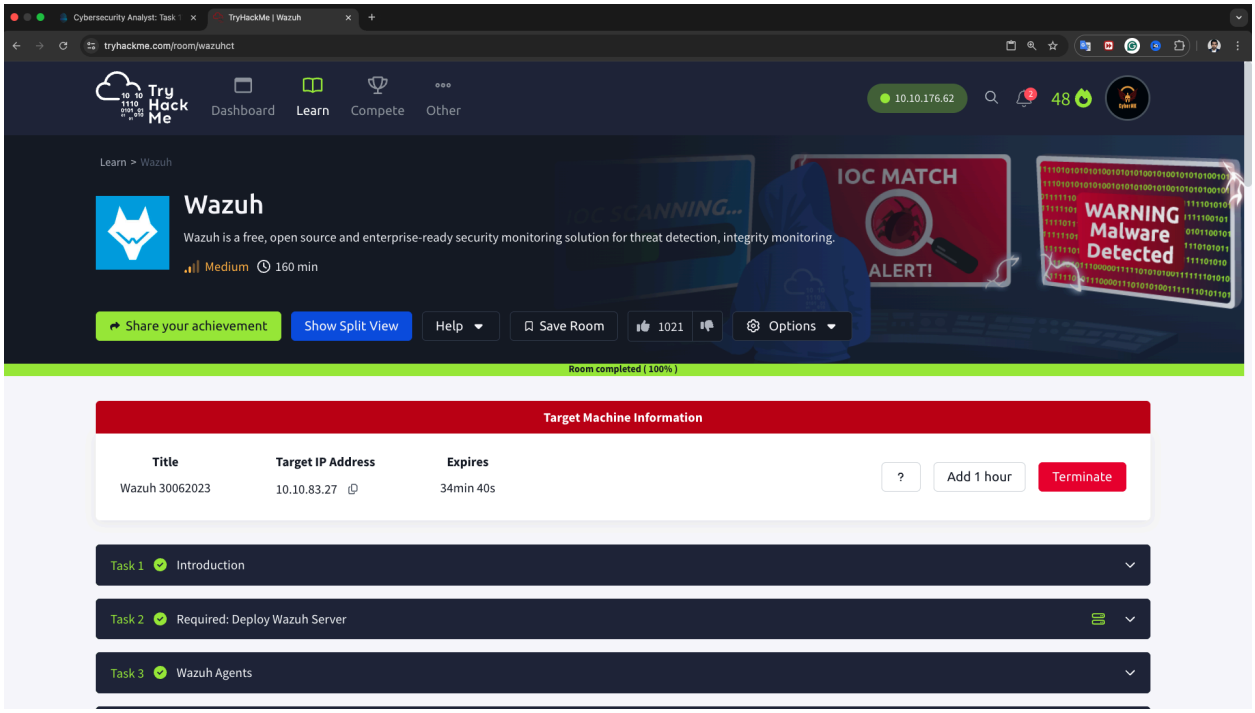
- Installing and configuring the Wazuh server and agents
- Collecting and analyzing logs from endpoints
- Detecting brute-force attempts and file integrity violations
- Reviewing triggered alerts and system responses

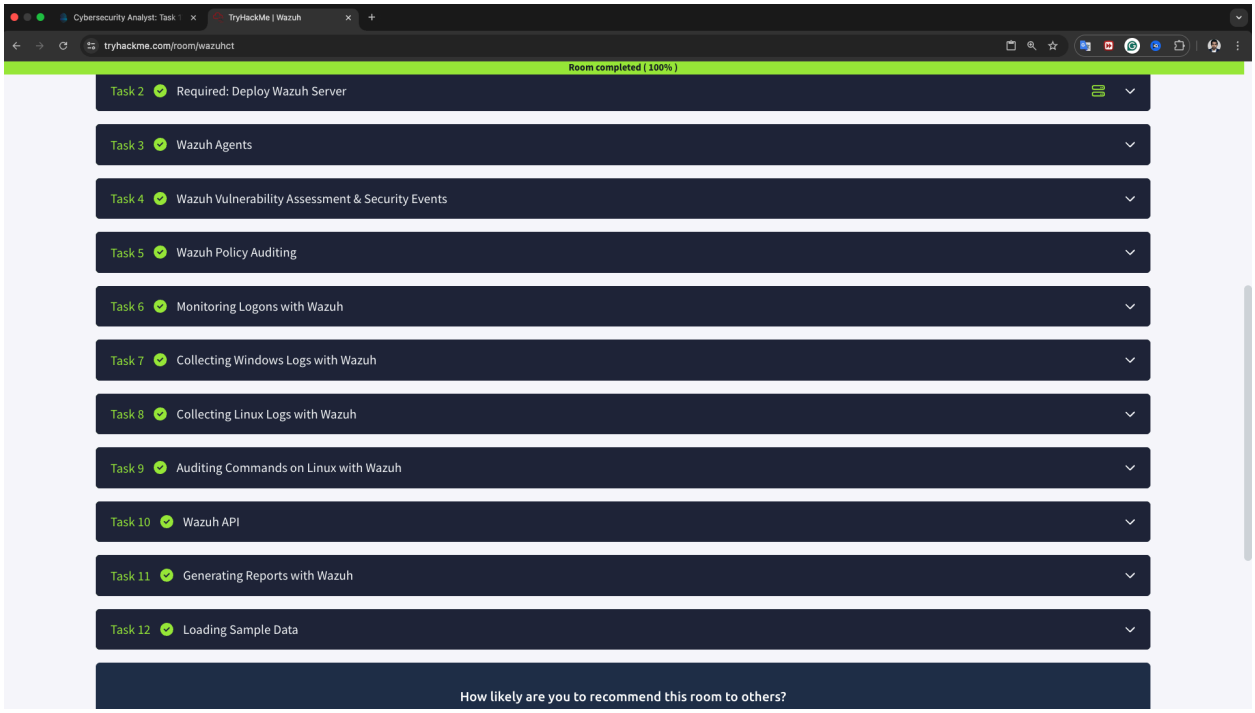
Lab Highlights:

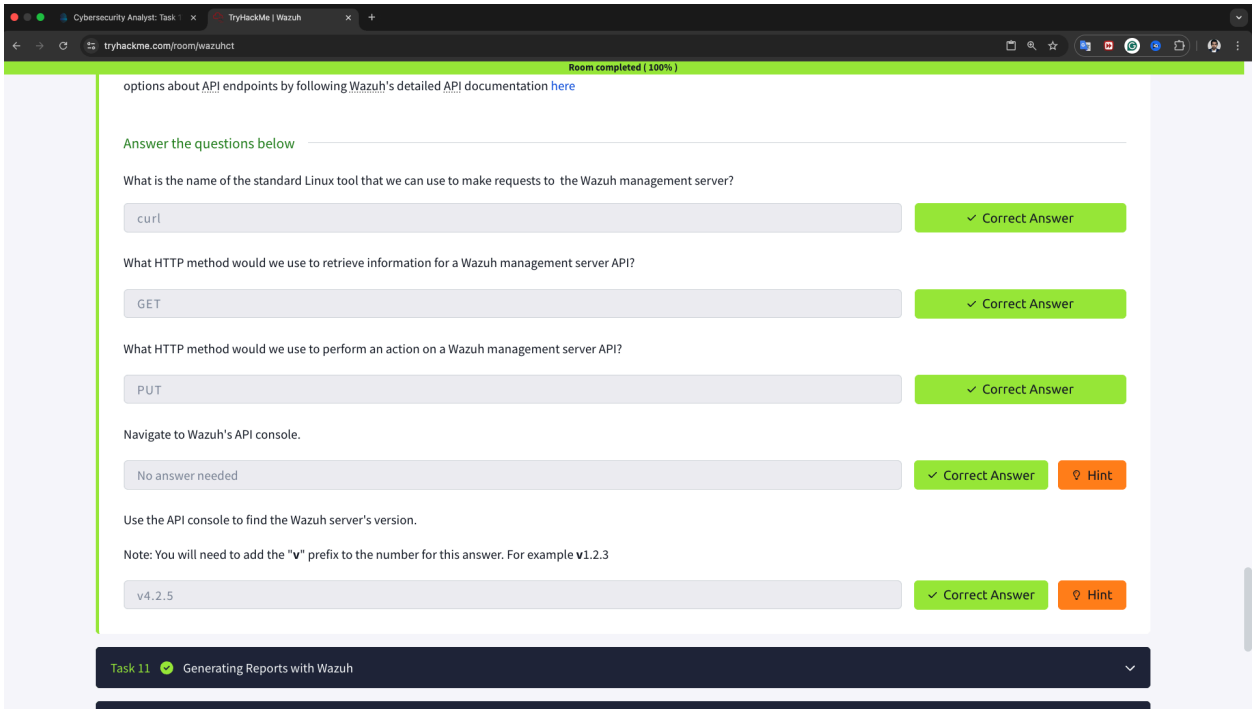
- Successfully configured Wazuh agent on a simulated endpoint
- Triggered and observed alerts for unauthorized login attempts
- Used the Wazuh dashboard to correlate and analyze incidents
- Explored modules such as File Integrity Monitoring (FIM) and process monitoring

These labs provided a safe and interactive platform to apply theoretical knowledge from the earlier sections. The ability to simulate incidents and view Wazuh's real-time responses helped reinforce understanding of endpoint visibility, behavioral monitoring, and automated alerting mechanisms.

Due to hardware limitations with Apple Silicon-based devices (MacStudio), performing these exercises in local virtual machines was not feasible. Therefore, the TryHackMe platform was essential in completing this task and demonstrating proficiency in endpoint security and Wazuh configuration.


tryhackme.com/room/wazuhct
Try Hack Me
Dashboard
Learn
Compete
Other
10.10.176.62
48
Learn > Wazuh
Wazuh
Wazuh is a free, open source and enterprise-ready security monitoring solution for threat detection, integrity monitoring.
Medium
160 min
IOC MATCH
ALERT!
WARNING
Malware Detected
Share your achievement
Show Split View
Help
Save Room
1021
Options
Room completed (100%)
Target Machine Information
Title
Wazuh 30062023
Target IP Address
10.10.83.27
Expires
34min 40s
?
Add 1 hour
Terminate
Task 1 Introduction
Task 2 Required: Deploy Wazuh Server
Task 3 Wazuh Agents


tryhackme.com/room/wazuhct
Room completed (100%)
Task 2 Required: Deploy Wazuh Server
Task 3 Wazuh Agents
Task 4 Wazuh Vulnerability Assessment & Security Events
Task 5 Wazuh Policy Auditing
Task 6 Monitoring Logons with Wazuh
Task 7 Collecting Windows Logs with Wazuh
Task 8 Collecting Linux Logs with Wazuh
Task 9 Auditing Commands on Linux with Wazuh
Task 10 Wazuh API
Task 11 Generating Reports with Wazuh
Task 12 Loading Sample Data
How likely are you to recommend this room to others?



8.0 Personal Reflection: Challenges Faced on Apple Silicon

Working with Wazuh on my **Apple Silicon (M1)** machine presented several technical challenges, which significantly impacted the time and effort required to complete this task.

Architecture Compatibility Issues

Wazuh components (especially the manager and Elastic Stack) are primarily built for **x86_64 (amd64)** architecture. However, Apple Silicon uses **ARM64 (aarch64)**, which caused multiple compatibility errors during installation:

- Many Docker images and precompiled binaries failed to run.
- Native installations often threw dependency errors or failed to compile from source.

Troubleshooting Virtual Environments

To bypass this, I experimented with several virtualization tools:

- **UTM** (QEMU-based): Resource-efficient but slow; networking and agent communication were unstable.
- **VMware Fusion (Tech Preview)**: Better performance, but again encountered issues when trying to bridge Wazuh agents and manager reliably.
- Each time I switched environments, I had to **reinstall Ubuntu or Kali Linux from scratch**, which added to the setup time and complexity.

Time Investment

Due to these technical hurdles, I spent almost **an entire week** testing, debugging, and switching environments just to get a working Wazuh setup. Despite the difficulties, I remained persistent and explored alternative solutions.

Final Solution

Eventually, I decided to use a **TryHackMe virtual machine**, which already had Wazuh set up for log analysis tasks. Although it was a **pre-configured older version**, it allowed me to simulate incident detection and continue with the rest of the internship tasks.

EncryptEdge Labs

This Internship Task report was developed on [April, 20, 2025]

By:

atalmamun@gmail.com