WAZUH-VT-THREATCHECK

AN INTEGRATED FRAMEWORK FOR AUTOMATED THREAT INTELLIGENCE INCIDENT RESPONSE

UNDER THE GUIDANCE OF

Mr. ASAD BABU

Faculty at TechByHeart

ABSTRACT

This project focuses on the deployment and configuration of **Wazuh**, an open-source Security Information and Event Management (SIEM) platform, integrated with multiple threat intelligence and monitoring tools — **VirusTotal**, **AlienVault OTX**, **MITRE ATT&CK**, **File Integrity Monitoring (FIM)**, and **Active Response**.

The objective of this project is to establish a comprehensive and automated security monitoring environment capable of detecting, analyzing, and responding to threats in real time.

A **single-node Wazuh environment** was deployed using Docker on a Windows platform and connected to a Windows agent for active monitoring. **VirusTotal integration** enabled automatic analysis of file hashes to identify malware, while **AlienVault OTX** provided global threat intelligence to verify suspicious IPs. The **MITRE ATT&CK integration** mapped security events to attacker tactics and techniques, enhancing incident visibility. **FIM** ensured the integrity of system files and registry entries by detecting unauthorized changes, and **Active Response** automated remediation actions upon detection of malicious activities.

Testing was conducted using simulated threats such as the **EICAR test file**, registry modifications, and malicious IP lookups to validate alerting and response mechanisms. The results confirmed that all integrations functioned as intended, providing enriched threat intelligence, improved detection accuracy, and automated mitigation.

This implementation demonstrates how Wazuh, when integrated with external intelligence sources and automated defense mechanisms, can serve as an efficient, scalable, and proactive solution for real-time threat detection and response — thereby strengthening the overall cybersecurity posture of the monitored environment.

TABLE OF CONTENTS:

INTRODUCTION

1.1 Wazuh

Wazuh is an open-source security monitoring and threat detection platform designed for enterprise environments. It provides real-time log analysis, intrusion detection, file integrity monitoring, vulnerability detection, and compliance reporting. Wazuh helps organizations maintain security visibility across their infrastructure, allowing administrators to quickly identify and respond to potential threats. Its modular architecture and integration capabilities make it suitable for both small-scale and large-scale deployments.

1.2 File Integrity Monitoring (FIM)

File Integrity Monitoring (FIM) in Wazuh continuously checks for unauthorized changes in system files, configurations, and registries. It helps detect potential security breaches, configuration tampering, or malware infections by monitoring file creation, modification, and deletion events.

When a file or registry change is detected, Wazuh generates alerts specifying the affected file path, type of change, and user responsible. FIM is crucial for compliance and forensic analysis, ensuring system integrity is maintained at all times. It plays a vital role in early detection of malicious activity, particularly in critical system directories.

1.3 VirusTotal Integration

VirusTotal is an online service that analyzes files, URLs, and IP addresses to detect viruses, malware, and other malicious content using multiple antivirus engines and threat intelligence tools. By providing detailed reports and file reputation data, VirusTotal enables users to identify potential security risks quickly. The service also offers an API, which allows automated integration with security platforms for real-time threat analysis.

Integrating Wazuh with VirusTotal enhances the security monitoring capabilities of Wazuh by automatically analyzing file hashes against VirusTotal's database. This integration allows Wazuh to generate alerts for potentially malicious files detected on monitored endpoints. By combining Wazuh's comprehensive monitoring with VirusTotal's threat intelligence, administrators gain actionable insights, improving incident response times and overall cybersecurity posture.

1.4 AlienVault OTX Integration

AlienVault Open Threat Exchange (OTX) is a collaborative threat intelligence platform that enables users to share and receive information about emerging cyber threats. OTX provides community-driven data known as "pulses," which contain indicators of compromise (IOCs) such as malicious IPs, domains, and file hashes.

Integrating **Wazuh with AlienVault OTX** allows automatic reputation checks of detected indicators against the OTX threat database. When Wazuh encounters a suspicious IP or file hash, it queries the OTX API to verify if the indicator is associated with any known malicious activities. This integration enhances Wazuh's detection capabilities by adding real-time, community-sourced threat intelligence to the SIEM environment.

1.5 MITRE ATT&CK Integration

The **MITRE ATT&CK** framework is a globally recognized knowledge base that categorizes adversarial tactics, techniques, and procedures (TTPs) observed in real-world cyberattacks. It helps analysts understand the behavior of attackers and the progression of security incidents.

Integrating MITRE ATT&CK with **Wazuh** maps generated alerts to specific MITRE techniques and tactics, such as execution, persistence, or privilege escalation. For instance, detections related to suspicious command executions or reconnaissance activities are automatically classified under corresponding MITRE IDs (e.g., T1059 – Command and Scripting Interpreter). This integration improves incident analysis, enabling security teams to quickly identify attack stages and respond effectively.

1.6 Active Response

Active Response in Wazuh provides automated remediation actions in response to specific alerts. Once a suspicious event or malicious activity is detected, Active Response scripts can execute predefined countermeasures such as blocking IP addresses, terminating processes, or isolating compromised endpoints.

This automation reduces the response time and limits the impact of attacks by containing threats immediately. Integrating Active Response with threat intelligence sources like VirusTotal and OTX allows Wazuh to not only detect but also respond dynamically, making it a proactive defense mechanism within the SOC environment.

.

OBJECTIVES

The main objectives of this project are to:

1. Set up a **Wazuh SIEM environment** using Docker Desktop on Windows to monitor and analyze security events.

2. Install and configure a **Wazuh Agent** on Windows and Linux systems for endpoint monitoring.

3. Establish secure communication between the **Wazuh Agent and Wazuh Manager** to ensure accurate event collection and reporting.

4. Integrate **VirusTotal** for enhanced file and hash-based malware detection.

5. Integrate **AlienVault OTX** for community-driven threat intelligence and real-time indicator reputation checks.

6. Enable **MITRE ATT&CK mapping** to classify detected threats based on adversarial tactics and techniques.

7. Implement **File Integrity Monitoring (FIM)** and **Active Response** to detect unauthorized system changes and automatically respond to security incidents.

8. Verify communication, detection accuracy, and automated alerting functionalities across all configured integrations.

PREREQUISITES.

System Requirements:

Single-node stack deployment:

- Operating system: Linux or Windows
- Architecture: AMD64
- CPU: At least 4 cores
- Memory: At least 8 GB of RAM for the Docker host
- Disk space: At least 50 GB storage for Docker images and data volumes

Wazuh agent deployment:

- Operating system: Linux or Windows
- Architecture: AMD64
- CPU: At least 2 cores
- Memory: At least 1 GB of RAM for the Docker host
- Disk space: At least 10 GB storage for Docker images and logs

RESEARCH METHODOLOGY

Required Software:

Install Docker Desktop (Windows) or Docker Engine (Linux)

Here we used windows version.

Wazuh agent installation:

- Download wazuh agent from https://documentation.wazuh.com/current/installationguide/wazuh-agent/wazuh-agent-package-windows.html

- Run the installer and complete setup.
- Edit configuration file: C:\Program Files (x86)\ossec-agent\ossec.conf Modify manager address to: <address>127.0.0.1</address>

```
<ossec_config>

  <client>
    <server>
      <address>127.0.0.1</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>windows, windows10</config-profile>
    <crypto_method>aes</crypto_method>
    <notify_time>20</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <enrollment>
      <enabled>yes</enabled>
      <groups>default</groups>
    </enrollment>
  </client>
```

Docker Desktop installation:

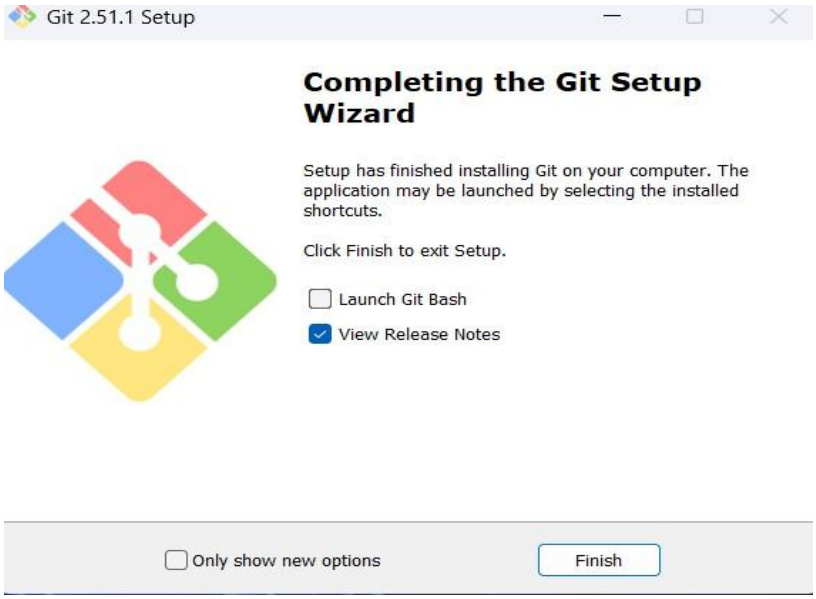
- Download “Docker Desktop for windows x86_64” from the given site https://docs.docker.com/desktop/setup/install/windows-install/
- Enable WSL 2 backend during installation.
- Start Docker Desktop after installation.
- Verify Docker installation using PowerShell:

docker version
docker info

```
PS C:\Windows\system32> docker version
Client:
 Version:           28.5.1
 API version:       1.51
 Go version:        go1.24.8
 Git commit:        e180ab8
 Built:             Wed Oct  8 12:19:16 2025
 OS/Arch:           windows/amd64
 Context:           desktop-linux

Server: Docker Desktop 4.48.0 (207573)
 Engine:
  Version:          28.5.1
  API version:      1.51 (minimum version 1.24)
  Go version:       go1.24.8
  Git commit:       f8215cc
  Built:            Wed Oct  8 12:17:24 2025
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.7.27
  GitCommit:        05044ec0a9a75232cad458027ca83437aae3f4da
 runc:
  Version:          1.2.5
  GitCommit:        v1.2.5-0-g59923ef
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
PS C:\Windows\system32>
```

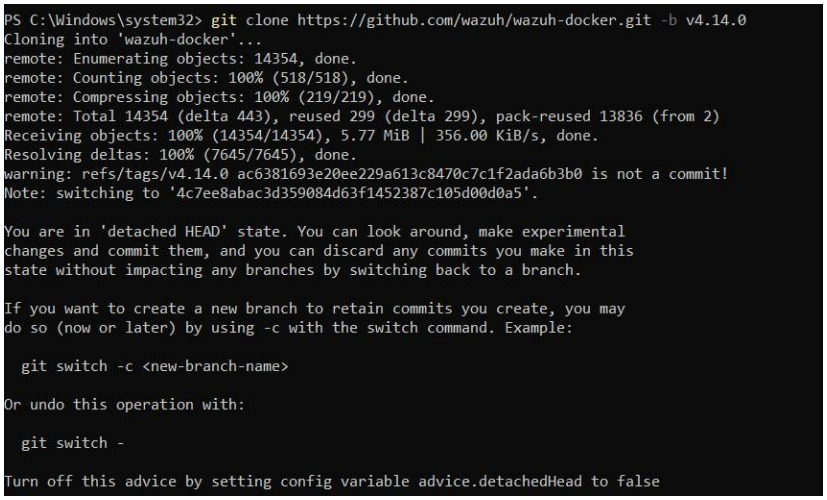
Git installation:

- Download and install “Git for Windows/x64 Setup” For cloning the Wazuh docker repository from https://git-scm.com/install/windows



Deploy Wazuh environment using Docker:

1. Clone the Wazuh docker repository to your system: git clone https://github.com/wazuh/wazuh-docker.git -b v4.14.0

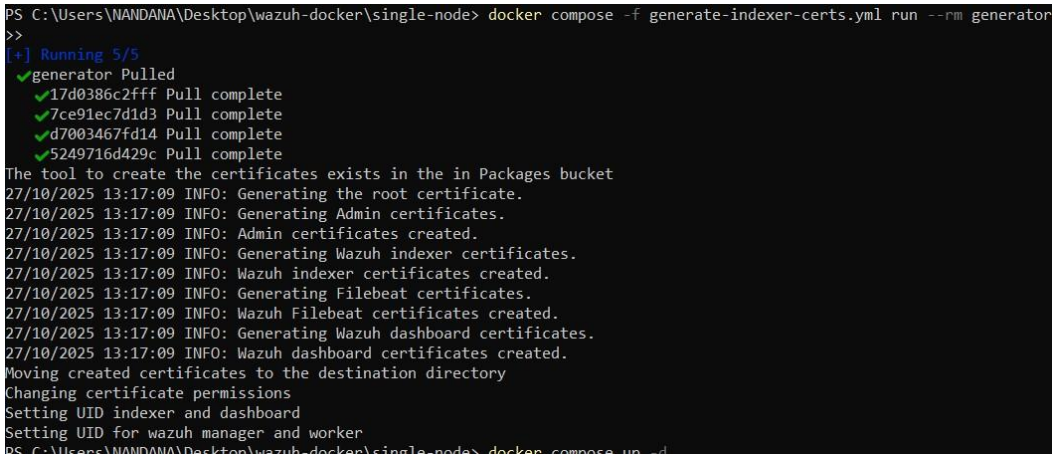


2. Navigate to the single-node directory to execute all the following commands.

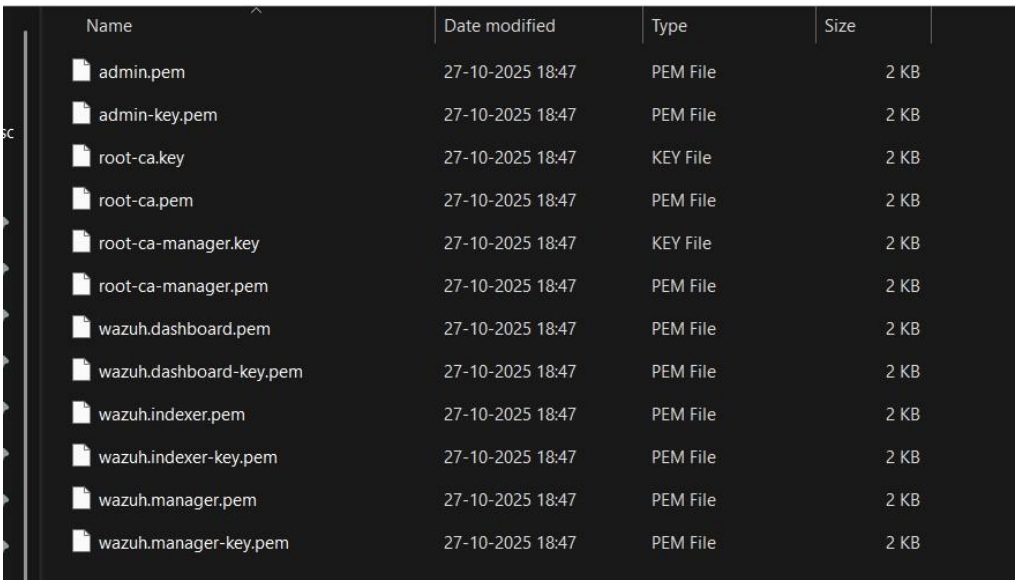
cd wazuh-docker/single-node/

Certificate generation:

1. Run the following command to generate the desired certificates: docker compose -f generate-indexer-certs.yml run --rm generator



2. The generated certificates will be stored in the wazuh-docker/singlenode/config/wazuh_indexer_ssl_certs directory.

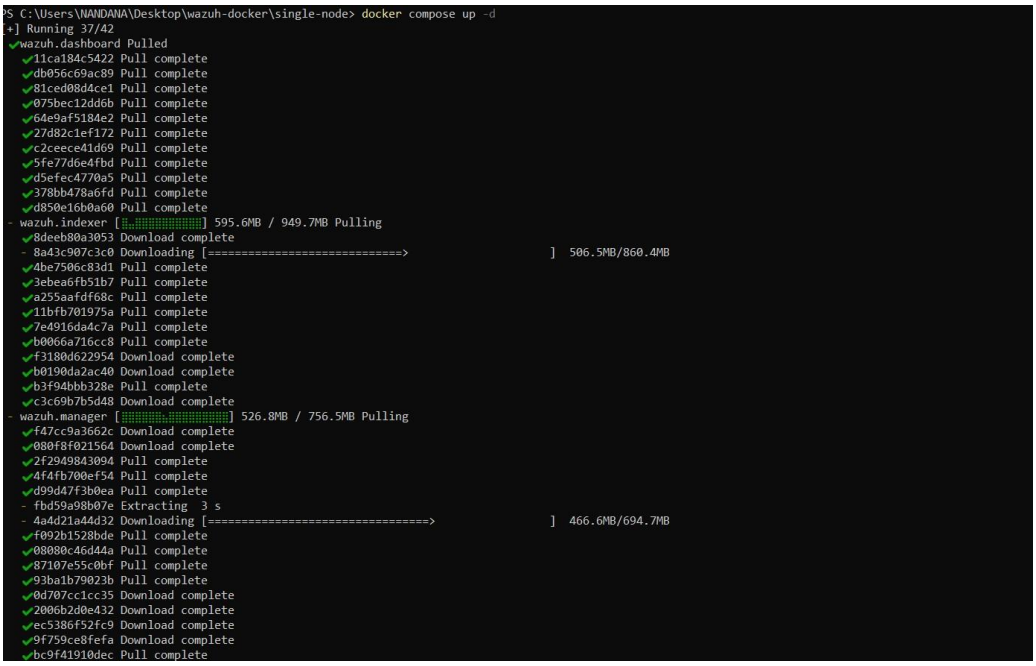


Name	Date modified	Type	Size
admin.pem	27-10-2025 18:47	PEM File	2 KB
admin-key.pem	27-10-2025 18:47	PEM File	2 KB
root-ca.key	27-10-2025 18:47	KEY File	2 KB
root-ca.pem	27-10-2025 18:47	PEM File	2 KB
root-ca-manager.key	27-10-2025 18:47	KEY File	2 KB
root-ca-manager.pem	27-10-2025 18:47	PEM File	2 KB
wazuh.dashboard.pem	27-10-2025 18:47	PEM File	2 KB
wazuh.dashboard-key.pem	27-10-2025 18:47	PEM File	2 KB
wazuh.indexer.pem	27-10-2025 18:47	PEM File	2 KB
wazuh.indexer-key.pem	27-10-2025 18:47	PEM File	2 KB
wazuh.manager.pem	27-10-2025 18:47	PEM File	2 KB
wazuh.manager-key.pem	27-10-2025 18:47	PEM File	2 KB

Deployment:

1. Start the Wazuh Docker deployment using the docker compose command:

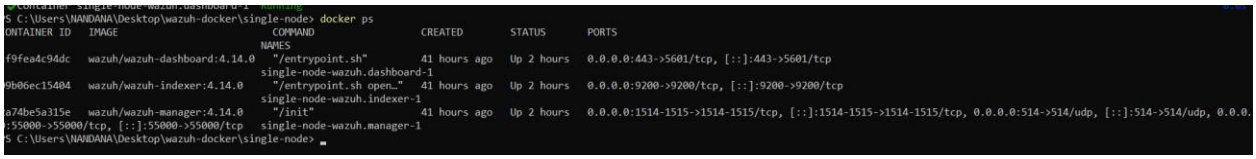
   docker compose up -d



```
PS C:\Users\NANDANA\Desktop\wazuh-docker\single-node> docker compose up -d
[+] Running 37/42
 ✔ wazuh.dashboard Pulled
   ✔ 11ca184c5422 Pull complete
   ✔ db056c69ac89 Pull complete
   ✔ 81ced08d4ce1 Pull complete
   ✔ 075bec12dd6b Pull complete
   ✔ 64e9af5184e2 Pull complete
   ✔ 27d82c1ef172 Pull complete
   ✔ c2ceece41d69 Pull complete
   ✔ 5fe77d6e4fbd Pull complete
   ✔ d5efec4770a5 Pull complete
   ✔ 378bb478a6fd Pull complete
   ✔ d850e16b0a60 Pull complete
 - wazuh.indexer [⣿⣀⣀⣀⣀⣀⣀⣀⣀⣀] 595.6MB / 949.7MB Pulling
   ✔ 8deeb80a3053 Download complete
   - 8a43c907c3c0 Downloading [============================>                  ]  506.5MB/860.4MB
   ✔ 4be7506c83d1 Pull complete
   ✔ 3ebea6fb51b7 Pull complete
   ✔ a255aafdf68c Pull complete
   ✔ 11bfb701975a Pull complete
   ✔ 7e4916da4c7a Pull complete
   ✔ b0066a716cc8 Pull complete
   ✔ f3180d622954 Download complete
   ✔ b0190da2ac40 Download complete
   ✔ b3f94bbb328e Pull complete
   ✔ c3c69b7b5d48 Download complete
 - wazuh.manager [⣿⣿⣿⣿⣿⣿⣿⣿⣀⣀⣀⣀⣀⣀⣀⣀⣀⣀] 526.8MB / 756.5MB Pulling
   ✔ f47cc9a3662c Download complete
   ✔ 080f8f021564 Download complete
   ✔ 2f2949843094 Pull complete
   ✔ 4f4fb700ef54 Pull complete
   ✔ d99d47f3b0ea Pull complete
   - fbd59a98b07e Extracting  3 s
   - 4a4d21a44d32 Downloading [================================>              ]  466.6MB/694.7MB
   ✔ f092b1528bde Pull complete
   ✔ 08080c46d44a Pull complete
   ✔ 87107e55c0bf Pull complete
   ✔ 93ba1b79023b Pull complete
   ✔ 0d707cc1cc35 Download complete
   ✔ 2006b2d0e432 Download complete
   ✔ ec5386f52fc9 Download complete
   ✔ 9f759ce8fefa Download complete
   ✔ bc9f41910dec Pull complete
```

2. Check whether the containers are running by using the given command

Docker ps

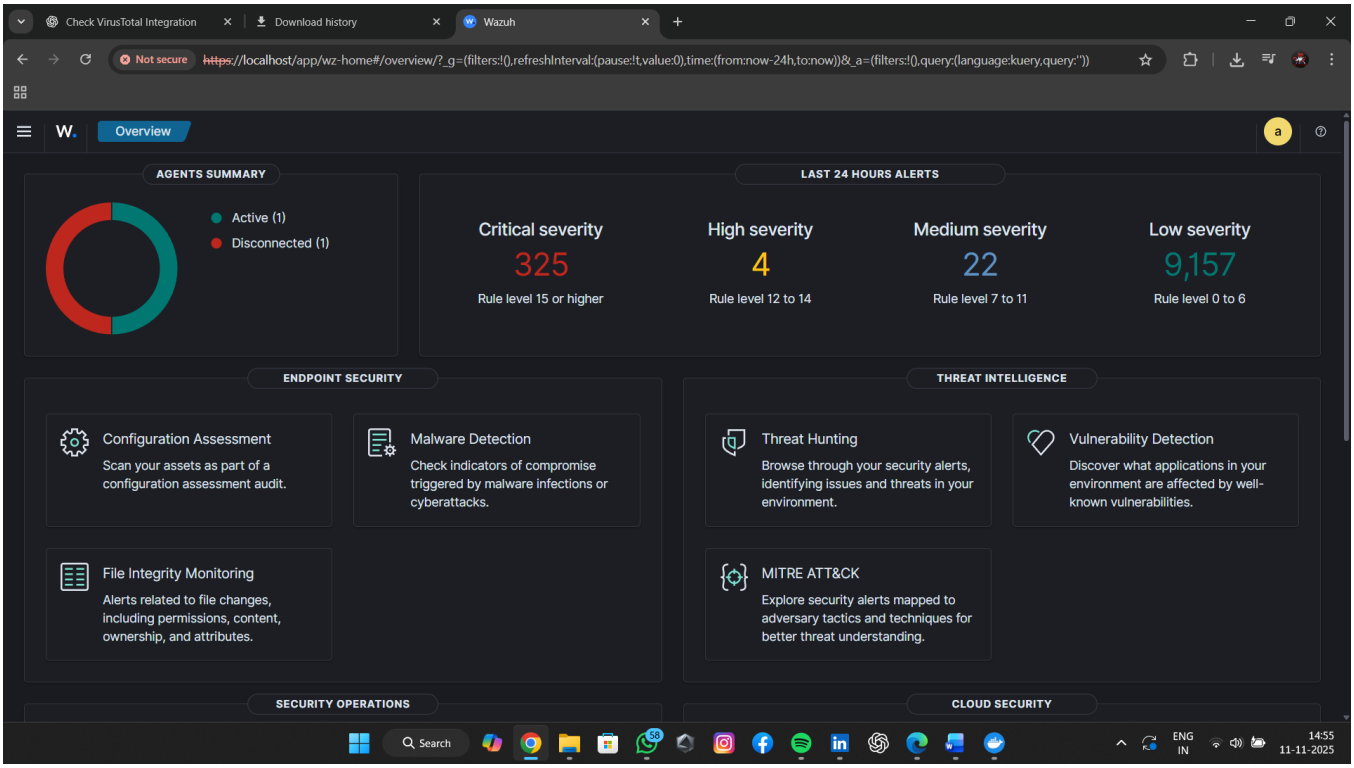


Accessing the Wazuh dashboard:

After deploying the single-node stack, you can access the Wazuh dashboard using your Docker host's IP address or localhost. https://<DOCKER_HOST_IP>

This is the default username and password to access the Wazuh dashboard:

- Username: admin
- Password: SecretPassword



File Integrity Monitoring (FIM)

Purpose:
FIM detects unauthorized changes in files, directories, or registry keys.

Steps:

1. **Edit the Configuration File**
   Open /var/ossec/etc/ossec.conf and add:

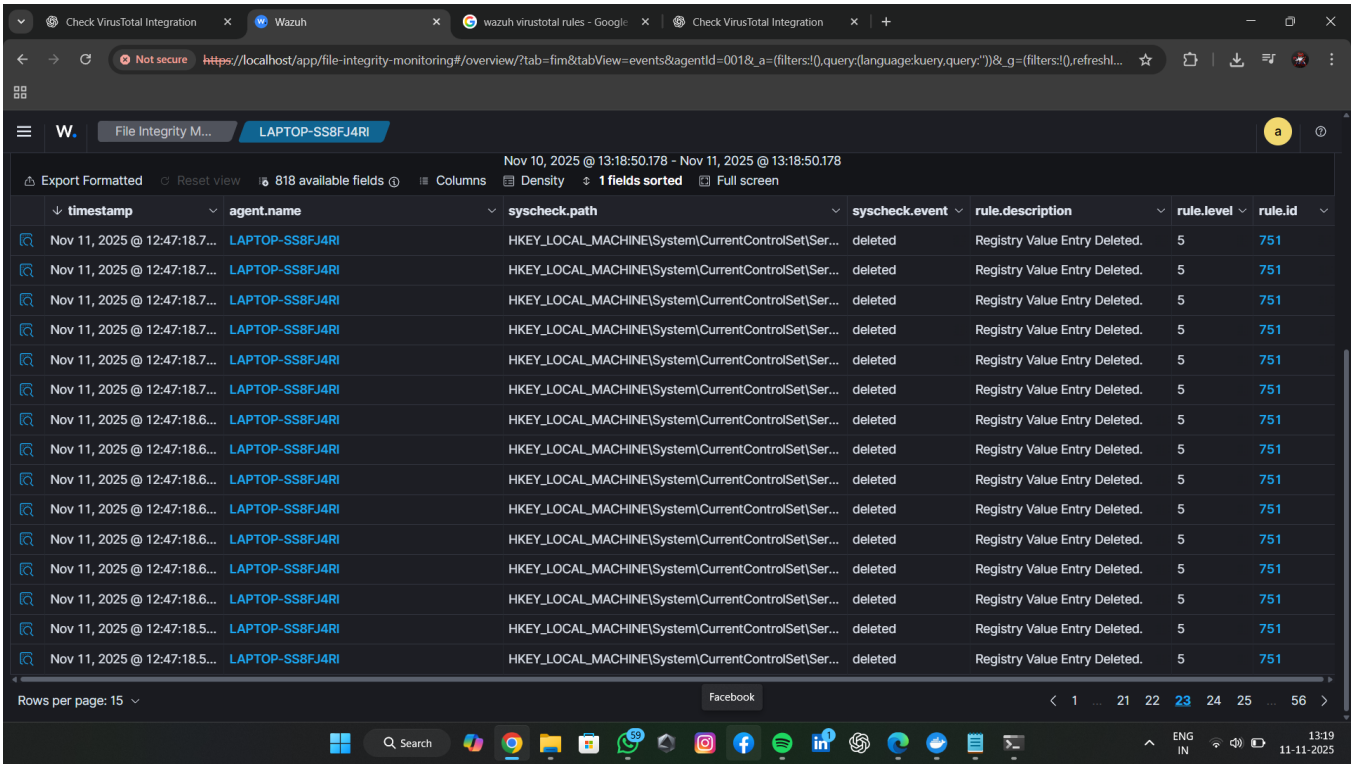
```
<syscheck>
  <directories check_all="yes">C:\Windows\System32</directories>
  <directories check_all="yes">C:\Users\Public</directories>
  <windows_registry>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services</windows_registry>
</syscheck>
```

2. **Restart the Wazuh Agent**

   #net stop wazuh

#net start wazuh

3. **Modify a File or Registry Key**
    - **Delete or modify a registry entry to trigger a detection.**
4. **Verify Alerts**
    - **Go to File Integrity Monitoring → Events**
    - **You'll see alerts like:**
    - **Registry Value Entry Deleted**



VirusTotal Integration:

- Login to VirusTotal account from https://www.virustotal.com/gui/home/upload
- Obtain your VirusTotal API Key
- Edit wazuh-manager.conf and add the integration block:

<integration>

```
  <name>virustotal</name>
  <api_key>YOUR_API_KEY_HERE</api_key>
  <group>syscheck</group>
  <alert_format>json</alert_format>
</integration>
```

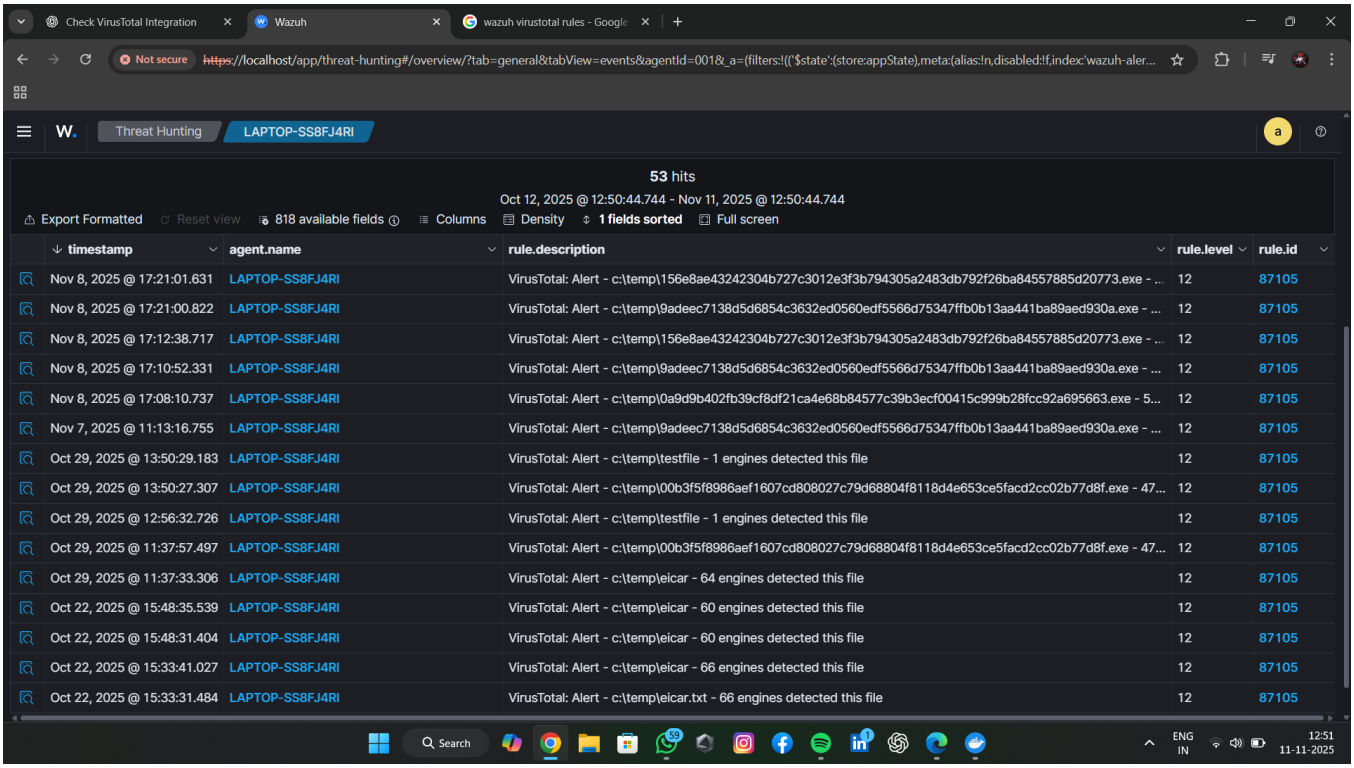
```
<ossec_config>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/ossec/logs/active-responses.log</location>
  </localfile>


<integration>
<name>virustotal</name>
<api_key>974e643bb23a92bb464d0d7df43e4f86f72bd9122ee94d547443bd55bf082aa1</api_key>
<group>syscheck</group>
<alert_format>json</alert_format>
</integration>


</ossec_config>
```

- Download EICAR File, is a safe test file used to check whether antivirus software is working properly without using any real malware using the given command

  Invoke-WebRequest -Uri "https://secure.eicar.org/eicar.com.txt" -OutFile "C:\Users\Public\eicar.com.txt"

```
PS C:\wazuh-docker\single-node> Invoke-WebRequest -Uri "https://secure.eicar.org/eicar.com.txt" -OutFile "C:\Users\Public\eicar.com.txt"
PS C:\wazuh-docker\single-node>
```

- Check for any alerts in Wazuh dashboard



AlienVault OTX Integration

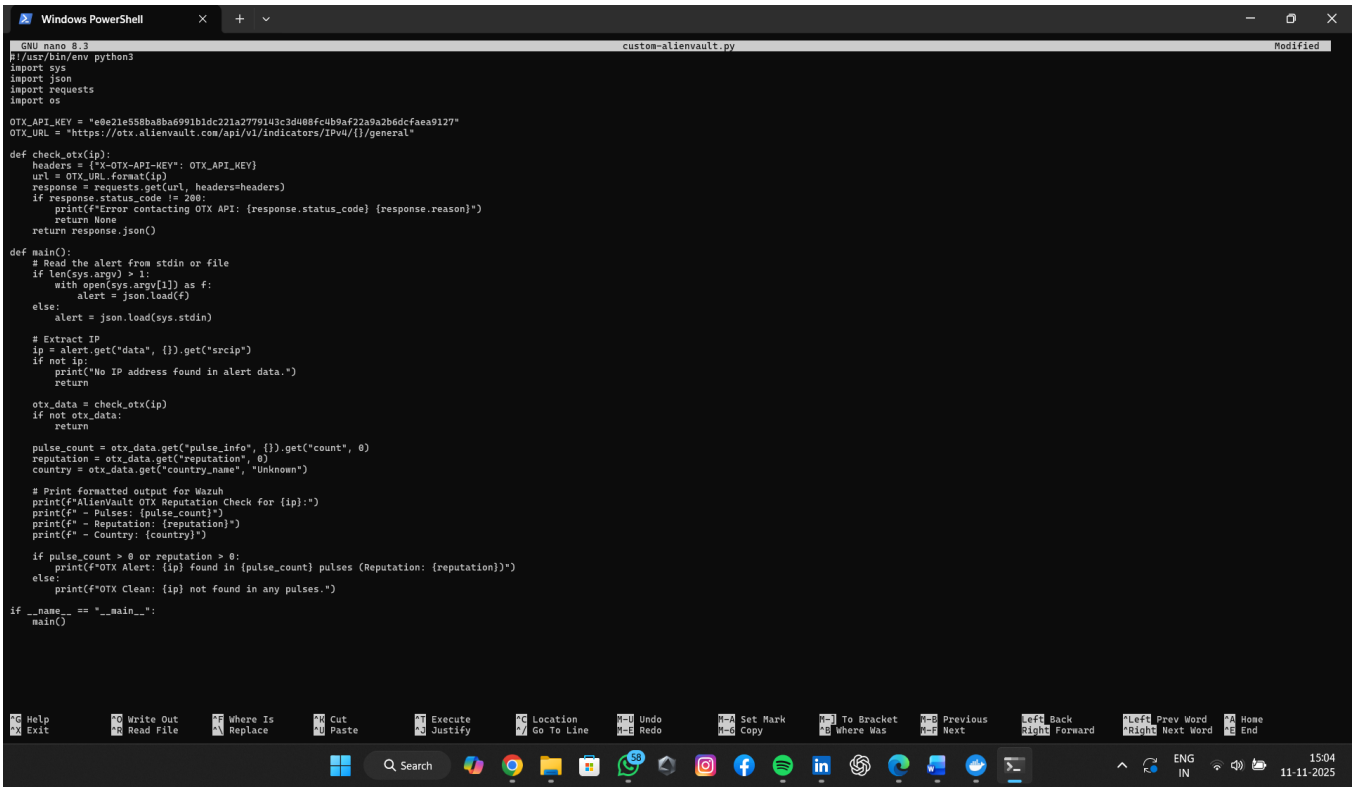
Purpose:
AlienVault OTX provides real-time threat intelligence. Integrating it with Wazuh allows reputation checks of suspicious IPs and domains against global threat data.

Steps:

1. **Obtain your OTX API Key**
    - **Login to https://otx.alienvault.com.**
    - **Go to Profile → API Key and copy it.**
2. **Create a Custom Integration Script**
   Run the following commands inside your Wazuh manager container:

   #cd /var/ossec/integrations/

   #nano custom-alienvault.py



3. **Paste your AlienVault OTX integration Python script.**

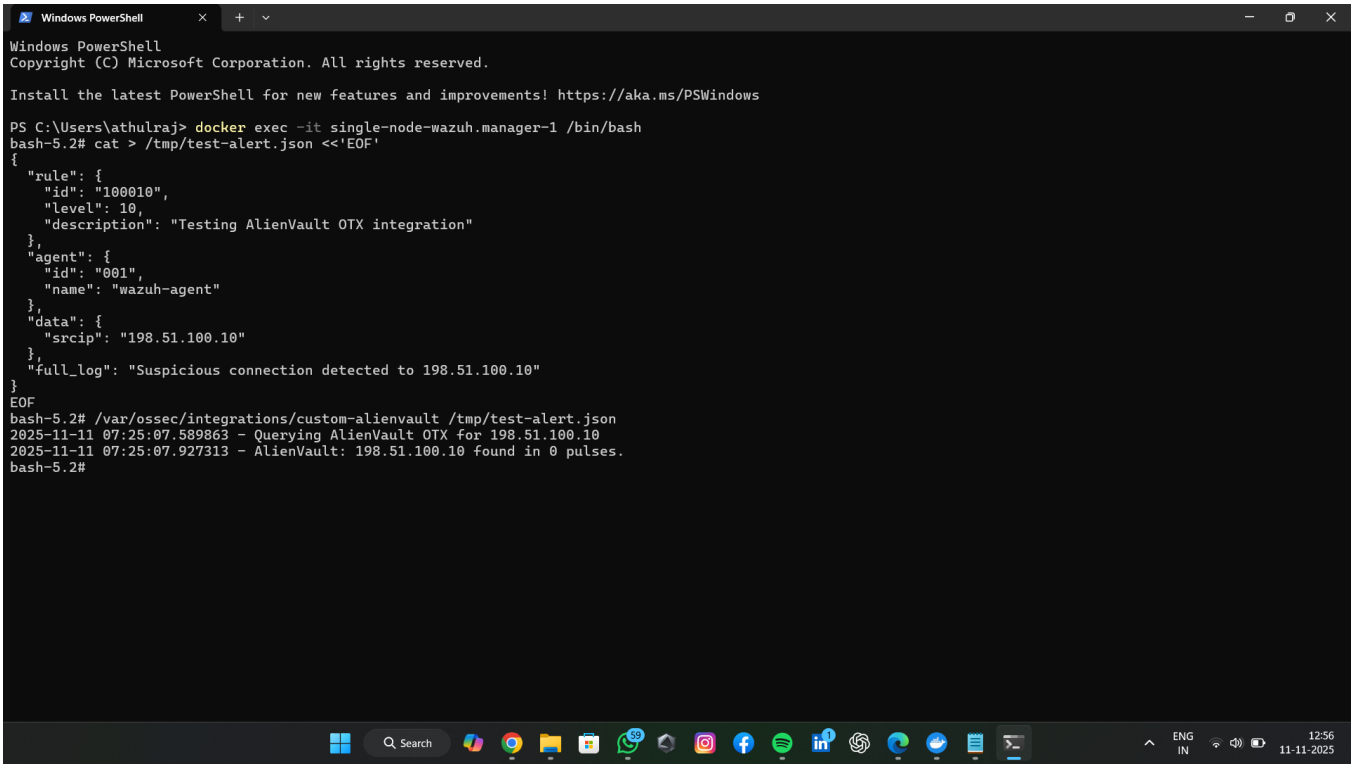
4. **Edit the Configuration File**
   Add the integration block inside /var/ossec/etc/ossec.conf:

```
  <integration>
  <name>custom-alienvault</name>
  <hook_url></hook_url>
  <level>3</level>
  <alert_format>json</alert_format>
  </integration>

</ossec_config>
```

5. **Restart the Wazuh Manager**

   #/var/ossec/bin/wazuh-control restart

6. **Test the Integration**



🧠 MITRE ATT&CK Integration

Purpose:
MITRE ATT&CK maps alerts to real-world attacker tactics and techniques for better context.

Steps:

1. **Ensure MITRE Module Is Enabled**
   Open the configuration file:

   #nano /var/ossec/etc/ossec.conf

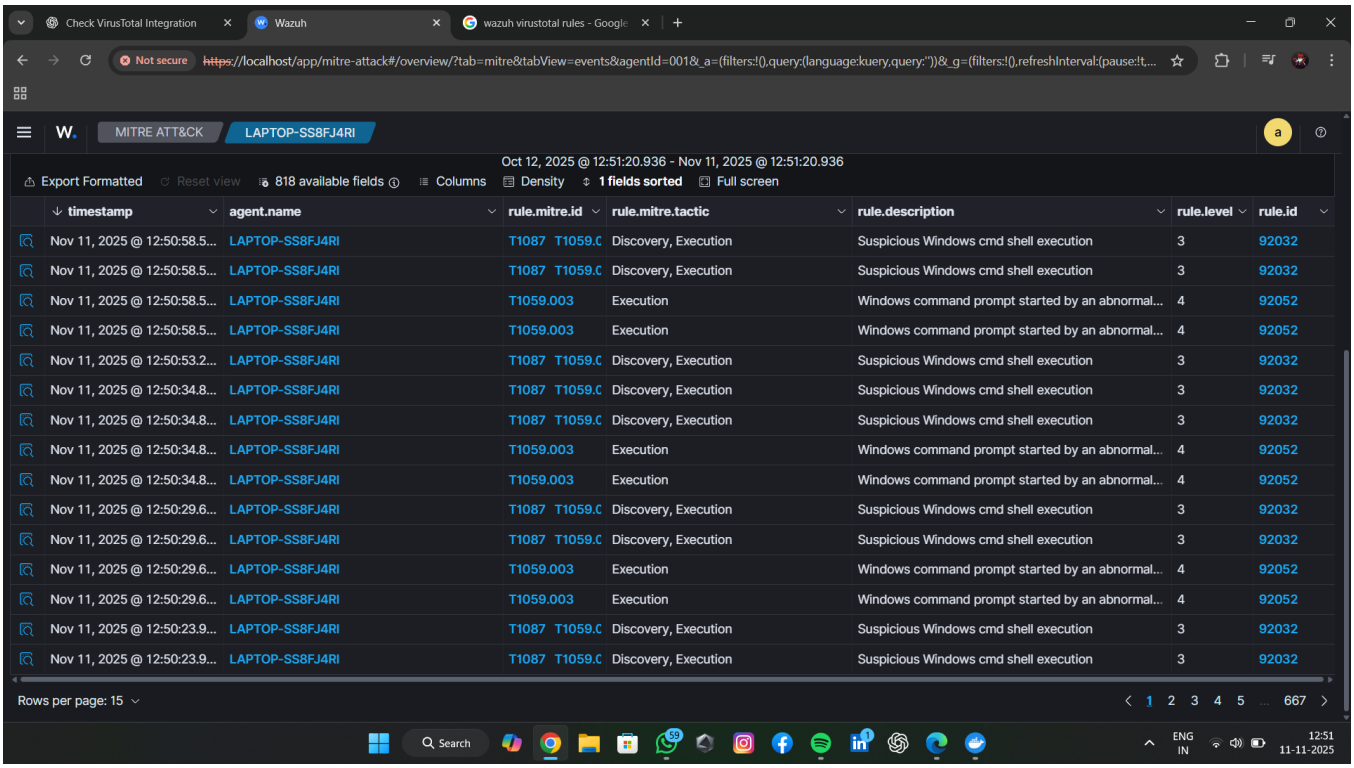
Verify that the following section exists:

```
<mitre>
  <enabled>yes</enabled>
</mitre>
```

2. **Restart Wazuh Manager**

#/var/ossec/bin/wazuh-control restart

3. **Generate MITRE Alerts**
   Perform suspicious activities such as:
   - Executing abnormal PowerShell commands
   - Running suspicious Windows CMD operations
4. **View Results in Dashboard**
   - Navigate to **MITRE ATT&CK → Overview**
   - Filter by **Agent Name** or **Technique ID (e.g., T1059, T1087)**



⚙ Active Response

Purpose:
Active Response allows Wazuh to automatically react to malicious events by running pre-configured scripts.

Steps:

1. **Enable Active Response**
   Add this to /var/ossec/etc/ossec.conf:

```
<active-response>
  <command>firewall-drop</command>
  <location>local</location>
  <rules_id>87105</rules_id>
  <timeout>600</timeout>
</active-response>
```

2. **Ensure Command Is Defined**
   Check /var/ossec/etc/shared/agent.conf:

```
<command>
  <name>firewall-drop</name>
  <executable>firewall-drop</executable>
  <expect>srcip</expect>
</command>
```

3. **Restart Wazuh**

   #/var/ossec/bin/wazuh-control restart

4. **Trigger a Malicious Event**
   - Run a malware file or trigger a VirusTotal alert.
   - Check logs:
   - tail -f /var/ossec/logs/active-responses.log

5. **View Result in Dashboard**
   - Go to **Threat Hunting → Events**
   - Filter for “firewall-drop” or “Active Response executed.”

RESULT AND OBSERVATION

After completing the configuration and integration process, the Wazuh environment successfully monitored the connected Windows and Linux agents. The Wazuh Dashboard displayed real-time event data, enabling efficient visualization of system activity and threat alerts.

The **VirusTotal integration** functioned effectively by automatically analyzing file hashes and detecting malicious files downloaded from test sources such as EICAR and MalwareBazaar. Alerts were generated when a file matched known malware signatures, confirming that VirusTotal API integration was operational.

The **AlienVault OTX integration** provided real-time threat intelligence by cross-verifying suspicious IPs with the global OTX pulse database. When simulated malicious connections were executed, Wazuh queried OTX and generated corresponding log entries, validating successful communication with the AlienVault API.

The **MITRE ATT&CK integration** accurately mapped alerts to their respective tactics and techniques (e.g., Execution, Discovery). This mapping enhanced the contextual understanding of security incidents and allowed visualization of attacker behavior patterns in the dashboard's MITRE ATT&CK tab.

The **File Integrity Monitoring (FIM)** feature effectively detected unauthorized registry and file changes on the Windows system. Whenever files or registry keys were modified or deleted, Wazuh generated alerts (Rule ID: 751), confirming FIM's role in maintaining system integrity.

Finally, the **Active Response module** demonstrated Wazuh's ability to perform automated defensive actions. On detection of specific malicious events, Wazuh triggered predefined active response scripts such as firewall-drop, enabling immediate mitigation of potential threats.

Overall, all integrations operated as expected. Wazuh's centralized monitoring, combined with these external intelligence feeds and automated responses, significantly enhanced the environment's detection accuracy and response capability.

CONCLUSION

The successful deployment and integration of **Wazuh with VirusTotal, AlienVault OTX, MITRE ATT&CK, File Integrity Monitoring (FIM), and Active Response** demonstrate a comprehensive and practical approach to real-time threat detection, analysis, and remediation.

Through **VirusTotal**, Wazuh effectively enriched file-based alerts with global malware intelligence. **AlienVault OTX** added collaborative threat intelligence, ensuring detection of emerging global threats. The **MITRE ATT&CK** integration provided structured visibility into attacker behavior, mapping events to known adversarial tactics and techniques for better incident analysis. **FIM** contributed by ensuring system integrity and compliance monitoring, while **Active Response** automated the reaction process, reducing manual intervention and response time.

This combination transformed Wazuh from a standard SIEM into a proactive, intelligence-driven security solution capable of identifying, correlating, and responding to threats efficiently. The project validates Wazuh's capability to serve as an essential component in a **Security Operations Center (SOC)** setup, providing scalability, automation, and deep visibility across endpoints.

In the future, this environment can be further enhanced by integrating **machine learning–based anomaly detection**, **custom dashboards**, and **automated playbooks**, thereby strengthening proactive defense, incident correlation, and overall cybersecurity resilience.