Endpoint Threat Monitoring and Response Using Wazuh

Prepared by: Mythresh Sai Mahadev

Masters in Cybersecurity Operations

Webster University

Project Overview

The "Endpoint Threat Monitoring and Response Using Wazuh" project explores the deployment and use of Wazuh as a comprehensive endpoint security solution. It involves configuring Wazuh Manager, connecting endpoint agents, and monitoring logs to detect and respond to threats effectively.

The project demonstrates key features of Wazuh through practical implementations, including File Integrity Monitoring (FIM), detecting and blocking SSH brute-force attacks, and integrating Suricata IDS for network intrusion detection. These activities highlight the importance of endpoint monitoring in a layered cybersecurity approach, providing actionable insights and automated responses to potential threats.

Table of Contents

Introduction

Endpoint security is vital in combating the growing sophistication of cyber threats. This project explores the implementation of **Wazuh**, an open-source security platform, to monitor, detect, and respond to endpoint threats effectively.

The project involves setting up the Wazuh environment, integrating endpoint agents, and configuring monitoring rules. Practical use cases, such as detecting brute-force attacks, monitoring file integrity, and leveraging Suricata for network intrusion detection, demonstrate Wazuh's capabilities in enhancing endpoint security and building a robust defense against cyber risks.

Objectives

- Deploy and configure Wazuh Manager and Dashboard on an Ubuntu server.
- Connect endpoint agents (Windows/Linux) to the Wazuh Manager for centralized monitoring.
- Implement File Integrity Monitoring (FIM) and detect system changes in critical directories.
- Detect and block SSH brute-force attacks using automated response mechanisms.
- Integrate Suricata IDS with Wazuh for network intrusion detection.
- Monitor system logs to identify potential threats and generate actionable alerts.

Step 1: Setting Up Wazuh Environment

1.1 System Requirements

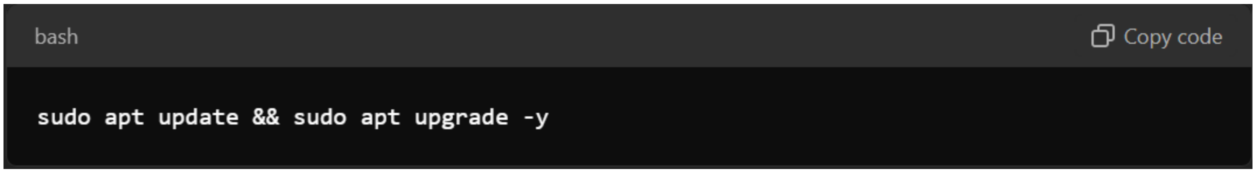
Before proceeding, ensure:

- **Ubuntu Version**: 20.04 or later.
- **RAM**: At least 4GB.
- **Disk Space**: 20GB or more.
- **Static IP Address**: The server should have a consistent IP.

1.2 Prepare Your Ubuntu Server

Step 1: Update and Upgrade the System

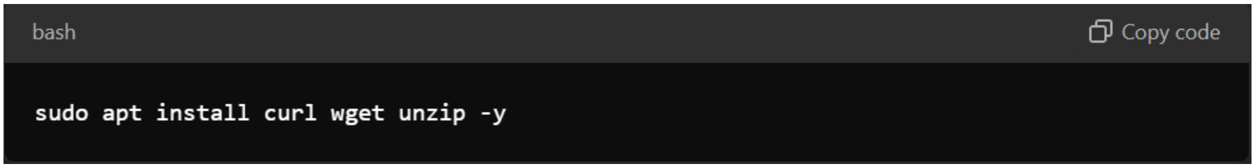
Always ensure the system is up-to-date:



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

Step 2: Install Required Tools

Install essential tools for downloading packages and troubleshooting:



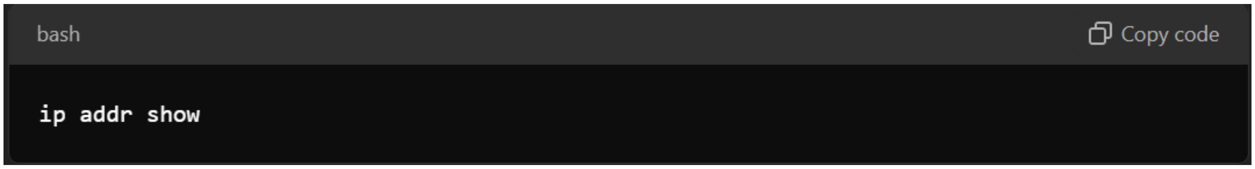
```bash
sudo apt install curl wget unzip -y
```

- curl: Used for downloading data from URLs.

- wget: Similar to curl, often useful in scripts.
- unzip: Needed for extracting compressed files.

Step 3: Verify Your Network Configuration

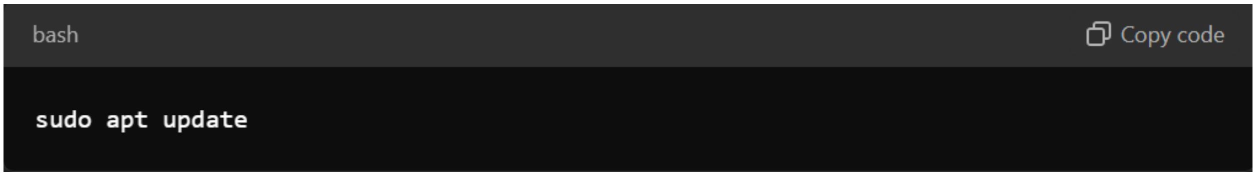
Check your server’s IP address:



```bash
ip addr show
```

- **Look for**: inet under your network interface (e.g., enp0s3).
- Note the IP address for use later (e.g., <manager-ip>).

Step 4: Update the package list



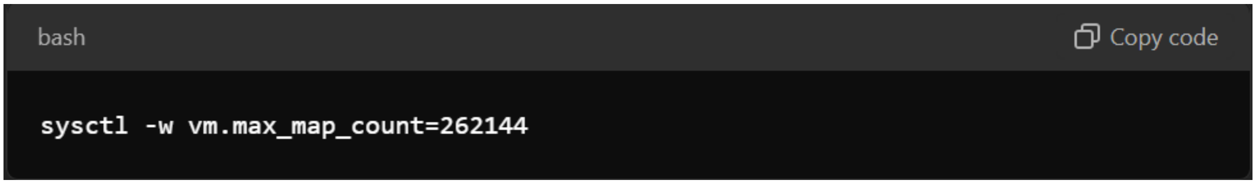
```bash
sudo apt update
```

1.3 Installing Wazuh Manager Using Docker

Open your web browser and visit the official Wazuh documentation: Navigate to the Docker Installation section:

Step 1: Configuring

1. Increase max_map_count on Your Host:
    - Run the following command to set the value temporarily:

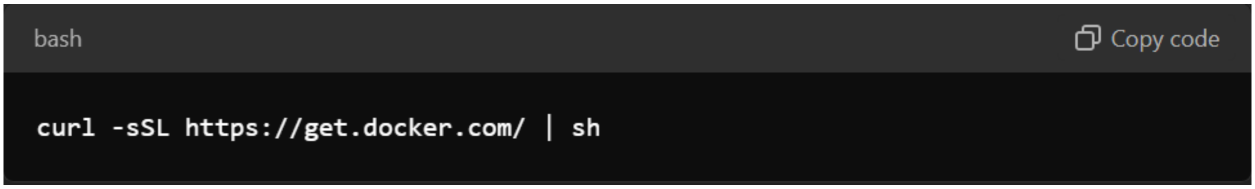


```bash
sysctl -w vm.max_map_count=262144
```

Step 2: Installing Docker

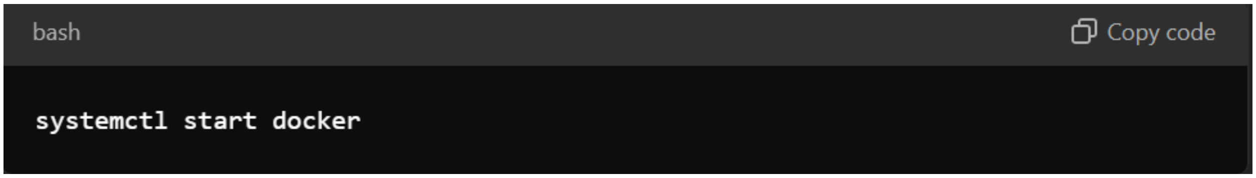
1. Download and Install Docker:

    - Run the Docker installation script:



```bash
curl -sSL https://get.docker.com/ | sh
```

2. Start the Docker Service using:



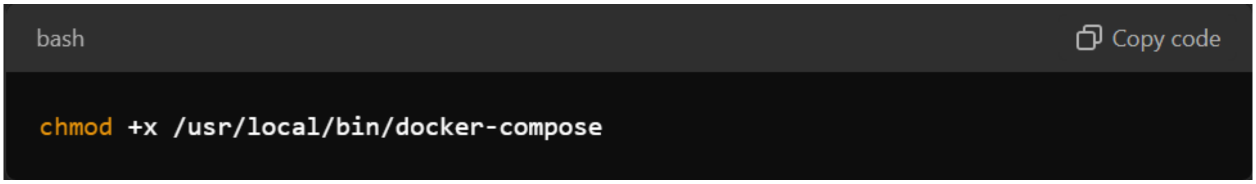
```bash
systemctl start docker
```

Step 2: Installing Docker Compose

1. **Download Docker Compose:**
   - Use the following command to download the binary:

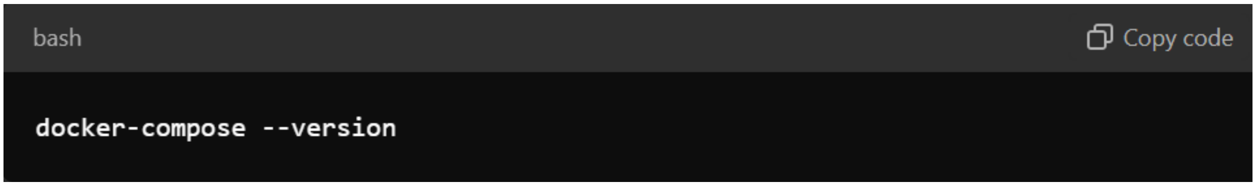
   curl -L "https://github.com/docker/compose/releases/download/v2.12.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

2. **Grant Execution Permissions:**
   - Make the binary executable



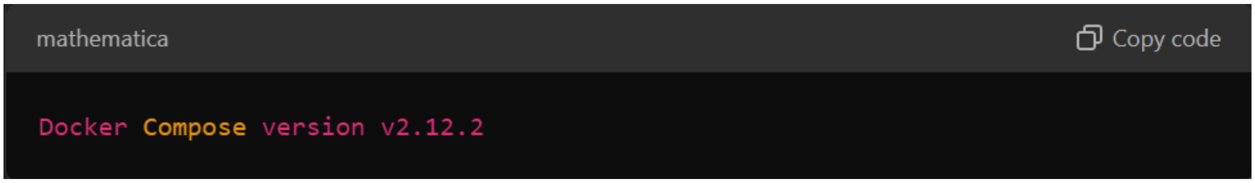
```bash
chmod +x /usr/local/bin/docker-compose
```

3. **Verify the Installation**:
   - Test the installation to ensure Docker Compose is installed



```bash
docker-compose --version
```

   - Example Output:



```mathematica
Docker Compose version v2.12.2
```

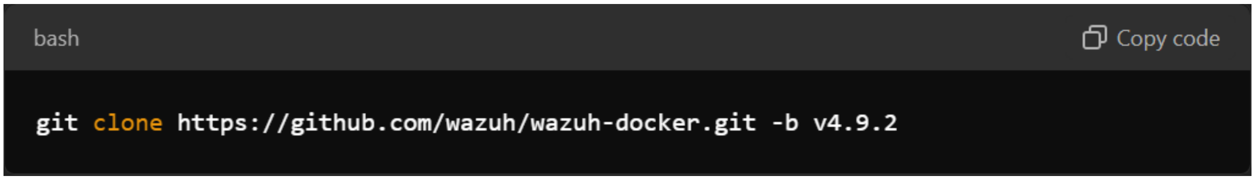
Step 3: Wazuh Docker Deployment

- You can deploy Wazuh as a **single-node** or **multi-node stack**.
- **Single-node deployment:** Deploys one Wazuh manager, indexer, and dashboard node.
- **Multi-node deployment:** Deploys two Wazuh manager nodes (one master and one worker), three Wazuh indexer nodes, and a Wazuh dashboard node.

Single-node Deployment Steps

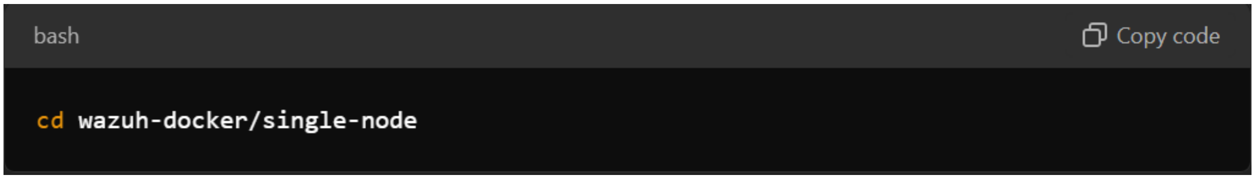
Step 3.1:

1. Open a terminal.
2. Clone the Wazuh Docker repository:



```bash
git clone https://github.com/wazuh/wazuh-docker.git -b v4.9.2
```

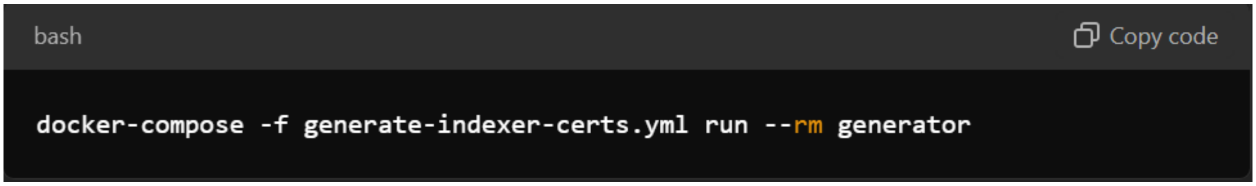
3. Navigate into the single-node directory:



```bash
cd wazuh-docker/single-node
```

Step 3.2: Provide Certificates for Secure Communication

1. Certificates are required to secure communication between Wazuh components.
   - Use the Wazuh certificate generator tool:



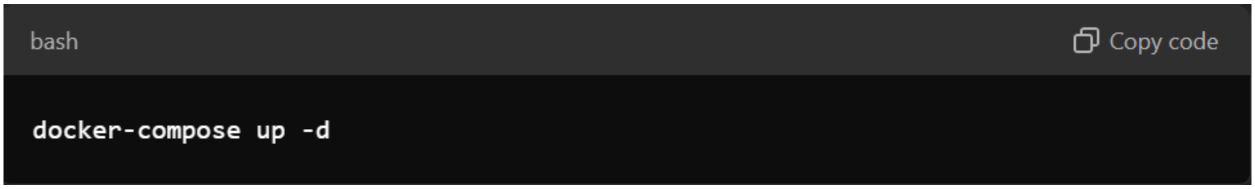
```bash
docker-compose -f generate-indexer-certs.yml run --rm generator
```

   - The generated certificates will be saved in the config/wazuh_indexer_ssl_certs directory.

Step 3.3: Start the Wazuh Single-Node Deployment

1. **Run the deployment**:

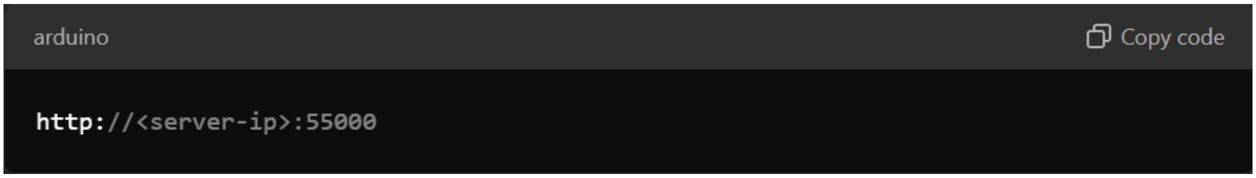
   - To run in the background:



```bash
docker-compose up -d
```

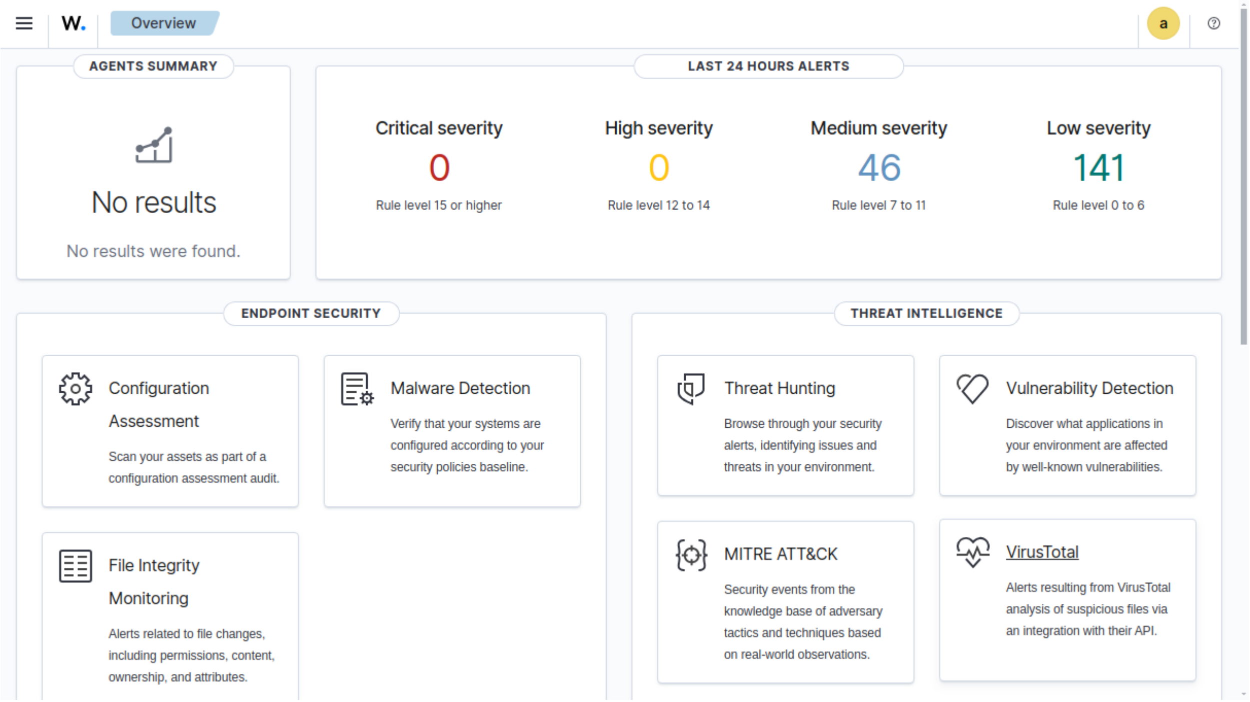
Step 4: Access the Dashboard

1. Open your browser and go to:



```
http://<server-ip>:55000
```

2. Replace <server-ip> with the IP address of your Ubuntu server.
3. Log in with default credentials:
    - **Username**: admin
    - **Password**: SecretPassword



No results

No results were found.

Critical severity	High severity	Medium severity	Low severity
0	0	46	141
Rule level 15 or higher	Rule level 12 to 14	Rule level 7 to 11	Rule level 0 to 6

Configuration Assessment

Scan your assets as part of a configuration assessment audit.

Malware Detection

Verify that your systems are configured according to your security policies baseline.

File Integrity Monitoring

Alerts related to file changes, including permissions, content, ownership, and attributes.

Threat Hunting

Browse through your security alerts, identifying issues and threats in your environment.

Vulnerability Detection

Discover what applications in your environment are affected by well-known vulnerabilities.

MITRE ATT&CK

Security events from the knowledge base of adversary tactics and techniques based on real-world observations.

VirusTotal

Alerts resulting from VirusTotal analysis of suspicious files via an integration with their API.

Step 2: Adding and Installing Wazuh Agent on Ubuntu Server

Step 2.1: Navigate to the Endpoint Summary

- On the dashboard, click on **Endpoint Summary** in the left-hand menu.

- This section displays a list of all existing agents.

Step 2.2: Add a New Agent

1. **Click on "Add Agent"**:
   - Locate and click the **Add Agent** button to start the process.
2. **Select the Agent Package**:
   - Choose the appropriate agent package for your operating system:
     - **Linux**: Select `DEB` (amd64) for Debian-based distributions.
     - **Windows**: Select the Windows installer.
     - **macOS**: Select the macOS installer.
3. **Set the Server Address**:
   - Enter the IP address or hostname of your Wazuh Manager.
4. **Assign an Agent Name**:
   - Provide a unique name for the agent. This will help identify it in the Wazuh Manager.
5. **Select a Group**:
   - Choose a **Default Group** or an **Existing Group** based on your deployment.

Step 2.3: Download and Install the Agent

1. **Run the Command to Download and Install**:
   - After selecting the options above, the Wazuh Dashboard generates a command.
   - Copy the generated command and run it on the endpoint machine (Linux terminal, Windows PowerShell, or macOS terminal).

Example for Linux (DEB):

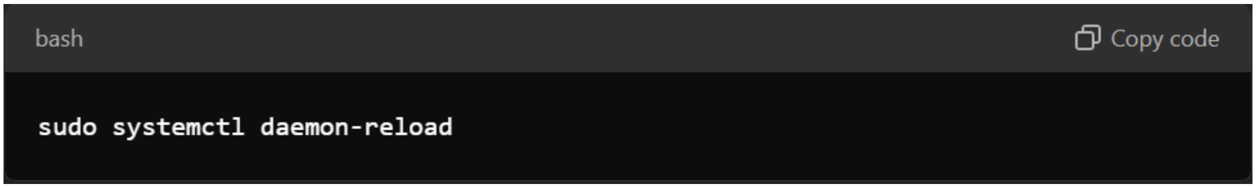
```
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.9.2-1_amd64.deb && sudo WAZUH_MANAGER='192.168.1.45' WAZUH_AGENT_NAME='UbuntuWazuh' dpkg -i ./wazuh-agent_4.9.2-1_amd64.deb
```

Replace <manager-ip> with the Wazuh Manager's IP address.

Step 2.4: Start and Enable Agent Service

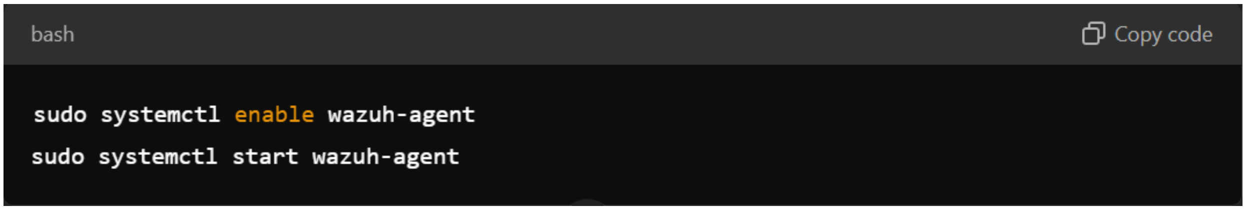
1. **Reload Systemd Daemon**:

- Run the following command:



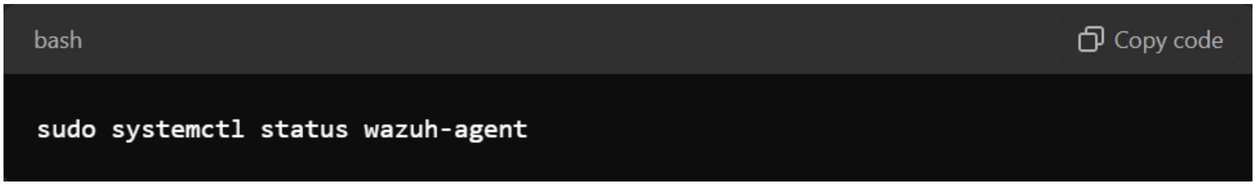
```bash
sudo systemctl daemon-reload
```

2. **Enable the Agent and start the service with:**



```bash
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
```

3. **Verify Agent Status**:
   - Check if the agent service is running:

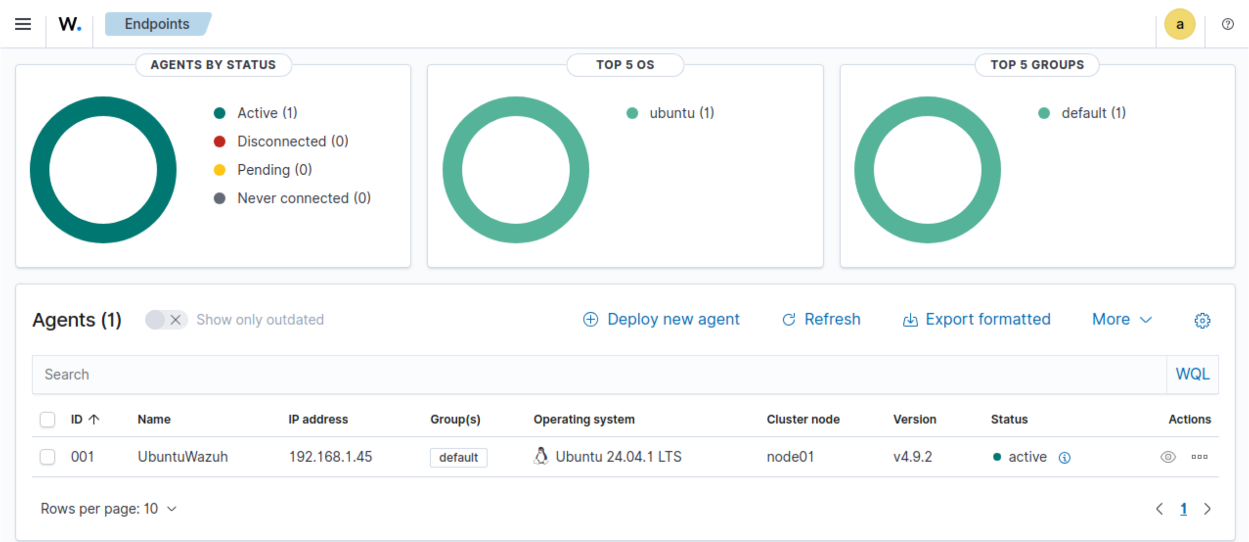


```bash
sudo systemctl status wazuh-agent
```

   - Look for active (running).

Step 2.5: Confirm Agent Registration

1. **Check the Wazuh Dashboard**:
   - Navigate back to the **Endpoint Summary** section



ID ↑	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	UbuntuWazuh	192.168.1.45	default	Ubuntu 24.04.1 LTS	node01	v4.9.2	● active	

   - Confirm that the newly added agent appears in the list with an **Active** status.

Step 3: Configuring Threat Monitoring and Rules

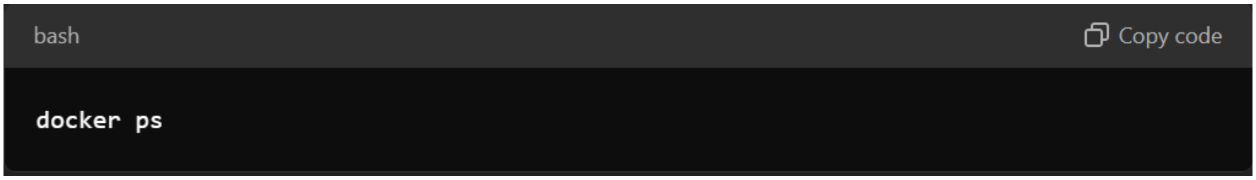
Lab 1 : To configure File Integrity Monitoring (FIM) using Wazuh to detect and generate alerts for changes in critical directories and files. This includes configuring both the Wazuh Manager (running in Docker) and the Wazuh Agent (on an Ubuntu server).

Step 1: Modifying Wazuh Manager Configuration

1. **Access the Wazuh Manager Container**

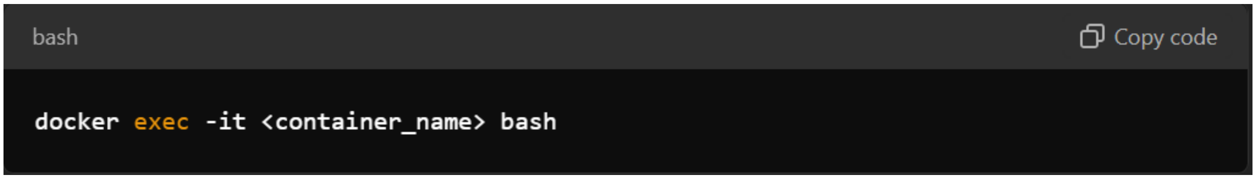
To modify the Wazuh Manager's configuration, first, access the Docker container:

- Identify the running Wazuh Manager container:



```bash
docker ps
```

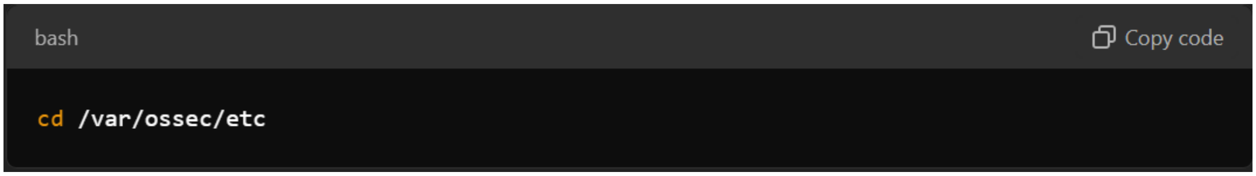
- Access the container’s shell:
- Note the container name or ID.



```bash
docker exec -it <container_name> bash
```

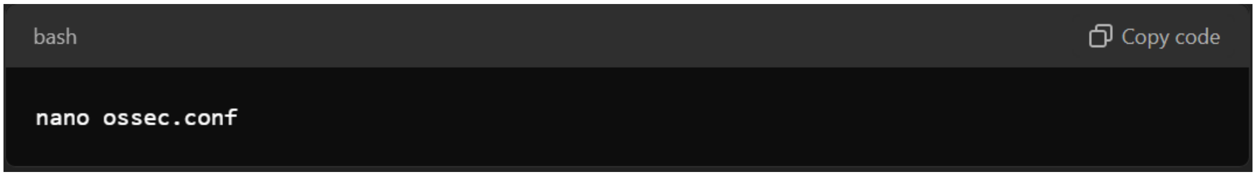
- Replace <container_name> with the name or ID of the Wazuh Manager container.

2. **Navigate to the Configuration Directory**
   - You’re now inside the container. Navigate to the directory containing the ossec.conf file:



```bash
cd /var/ossec/etc
```

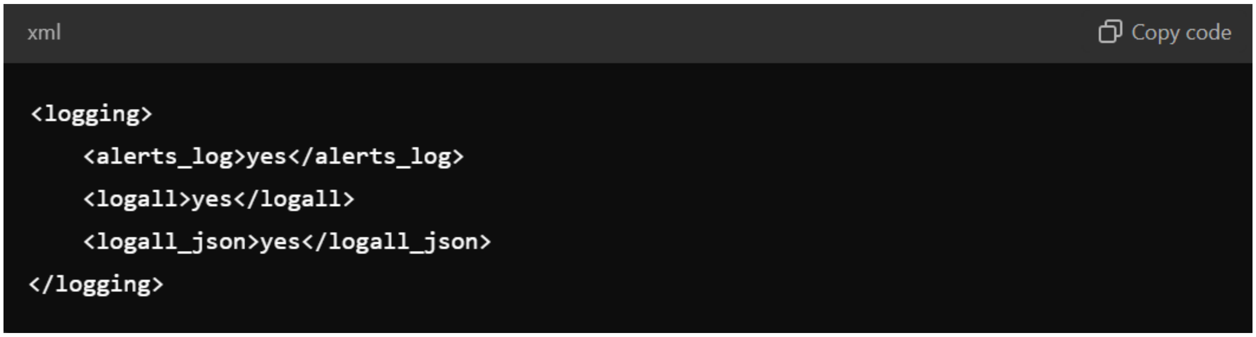
3. **Edit the ossec.conf File**
   - Open the ossec.conf file using a text editor:



```bash
nano ossec.conf
```

4. **Update the <logging> Section**

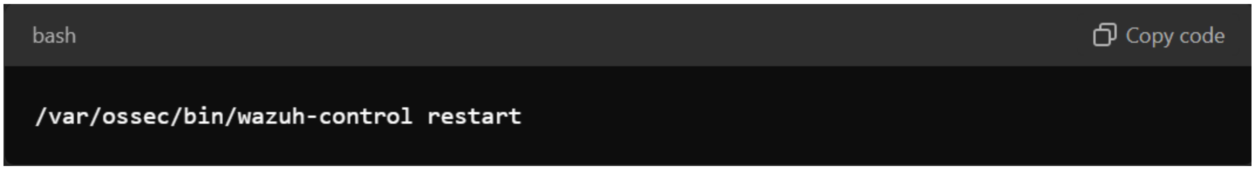
- Ensure logging is enabled to capture alerts and logs in various formats:



```xml
<logging>
    <alerts_log>yes</alerts_log>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
</logging>
```

5. **Save and Exit**
    - Press Ctrl+O, then Enter to save the file.
    - Press Ctrl+X to exit the editor.

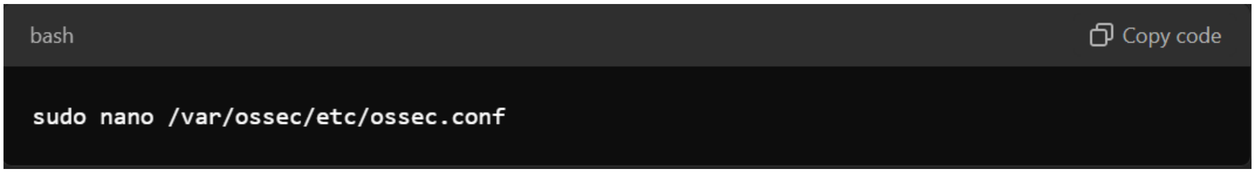
6. **Restart the Wazuh Manager**
    - Restart the Wazuh Manager to apply the changes:



```bash
/var/ossec/bin/wazuh-control restart
```

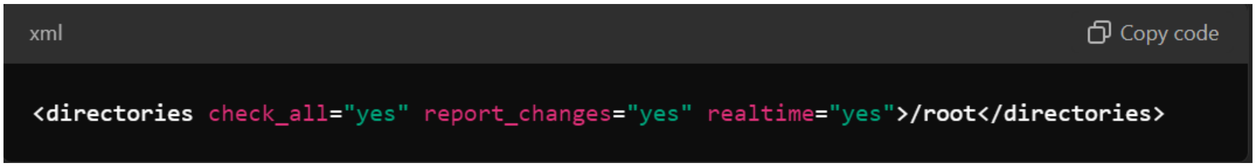
Step 2: Modifying Wazuh Agent Configuration

1. **Open the Agent Configuration File**
    - On the Ubuntu server where the Wazuh Agent is installed, open the ossec.conf file:



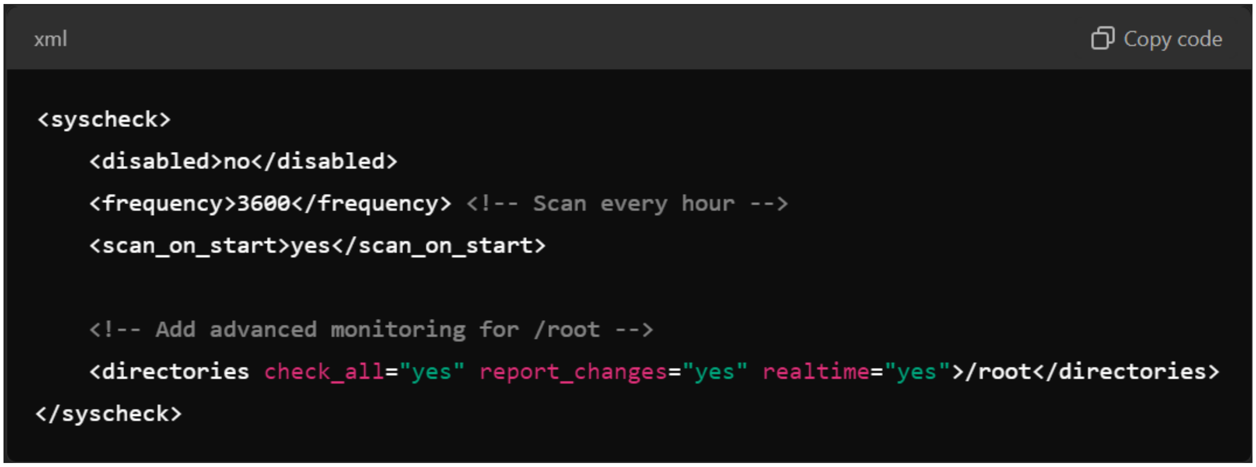
```bash
sudo nano /var/ossec/etc/ossec.conf
```

2. **Update the <syscheck> Section**
    - Add advanced monitoring options for the /root directory and other critical paths:



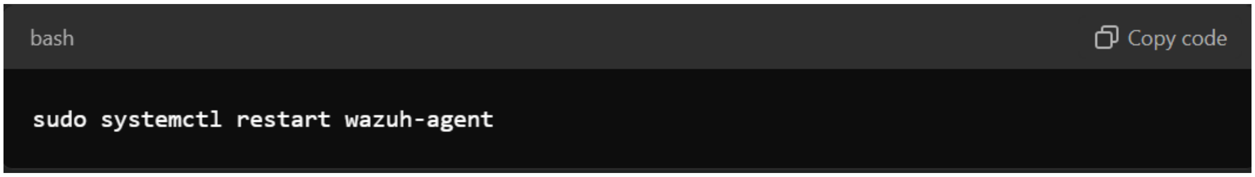
```xml
<directories check_all="yes" report_changes="yes" realtime="yes">/root</directories>
```

3. **Save and Exit**
   - Press Ctrl+O, then Enter to save.
   - Press Ctrl+X to exit the editor.



```xml
<syscheck>
    <disabled>no</disabled>
    <frequency>3600</frequency> <!-- Scan every hour -->
    <scan_on_start>yes</scan_on_start>

    <!-- Add advanced monitoring for /root -->
    <directories check_all="yes" report_changes="yes" realtime="yes">/root</directories>
</syscheck>
```

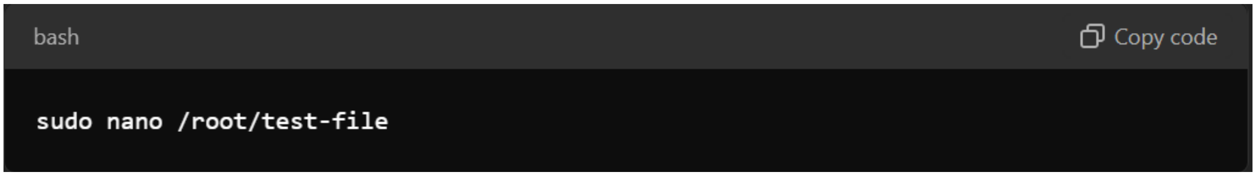
4. **Restart the Wazuh Agent**
   - Apply the changes by restarting the Wazuh Agent service:



```bash
sudo systemctl restart wazuh-agent
```

Step 3: Testing File Integrity Monitoring

1. **Create or modify a file in the /root directory:**



```bash
sudo nano /root/test-file
```

- Add some text and save the file.

2. Verify Alerts in the Wazuh Dashboard (4.9):

Step 3.1: Log into the Wazuh Dashboard

- Open the Wazuh Dashboard in your browser:
- Log in with your credentials.

Step 3.2: Navigate to the Wazuh Agent Endpoint Summary

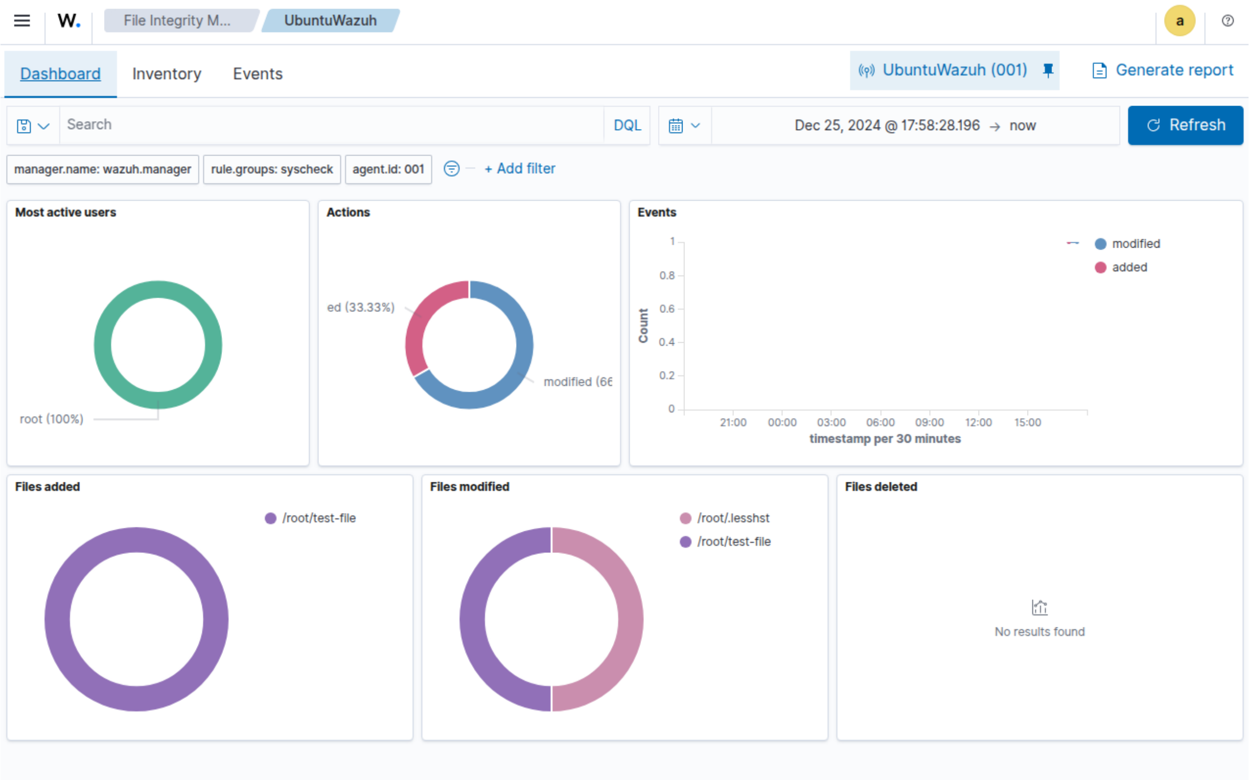
- In the Dashboard menu, go to **"Agents"**.

- Select the specific Agent where the modification was made (e.g., the hostname of your Wazuh Agent).
- This will open the **Agent Endpoint Summary**.

Step 3.3: View File Integrity Events

- In the Agent Endpoint Summary, locate and click on the **"File Integrity"** section.
- Here, you will see **Monitored events**: A list of recent file integrity events detected by the Agent.

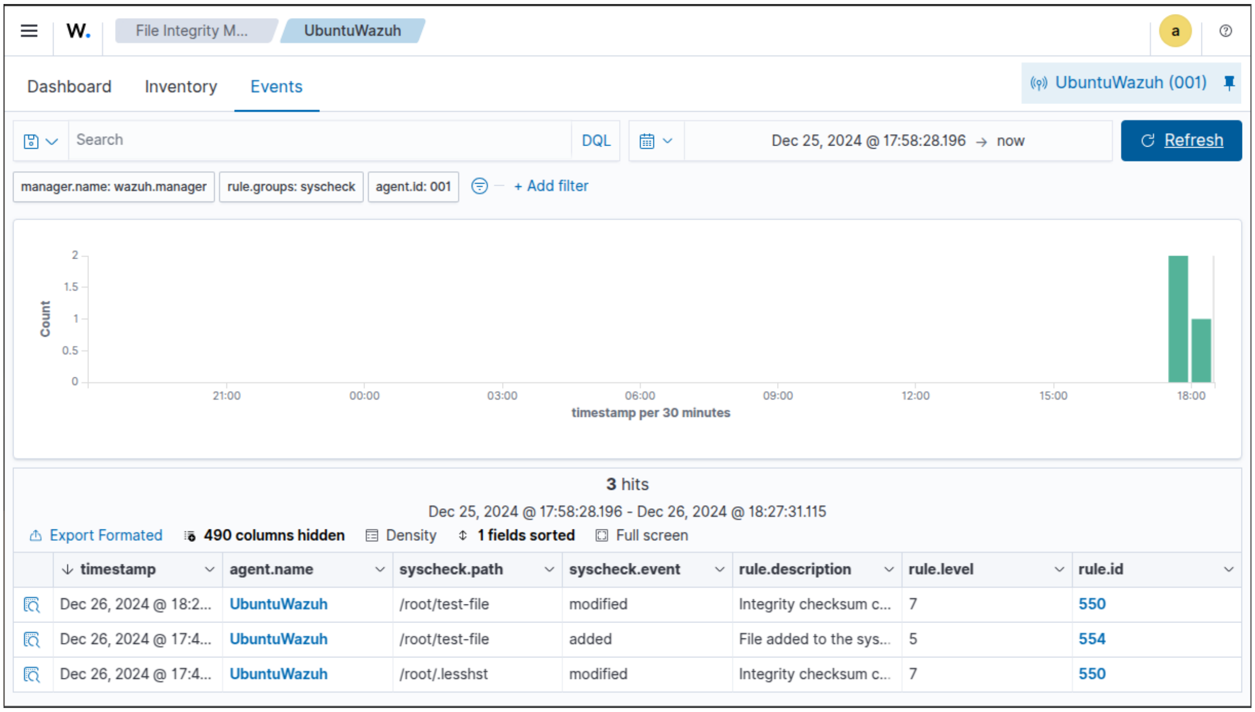
Dashboard View:



Dec 25, 2024 @ 17:58:28.196 → now

manager.name: wazuh.manager | rule.groups: syscheck | agent.id: 001

Events View:



3 hits

Dec 25, 2024 @ 17:58:28.196 - Dec 26, 2024 @ 18:27:31.115

timestamp	agent.name	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
Dec 26, 2024 @ 18:2...	UbuntuWazuh	/root/test-file	modified	Integrity checksum c...	7	550
Dec 26, 2024 @ 17:4...	UbuntuWazuh	/root/test-file	added	File added to the sys...	5	554
Dec 26, 2024 @ 17:4...	UbuntuWazuh	/root/.lesshst	modified	Integrity checksum c...	7	550

Lab 2: Detecting and Blocking SSH Brute-Force Attacks

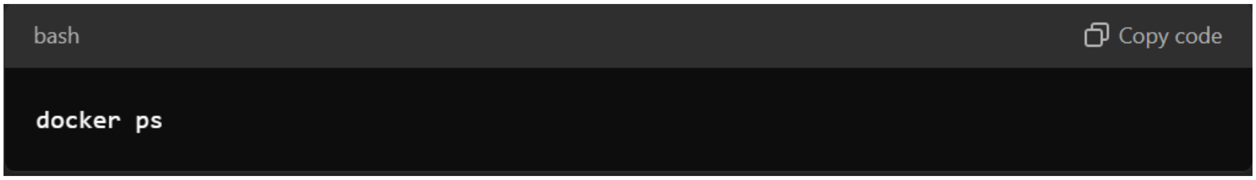
To configure Wazuh to monitor SSH authentication logs, detect brute-force attacks using **Rule ID 5763**, and block the attacking IP address automatically using active response mechanisms.

Step 1: Configuring Wazuh Manager for SSH Brute-Force Detection

1. Access the Wazuh Manager Container

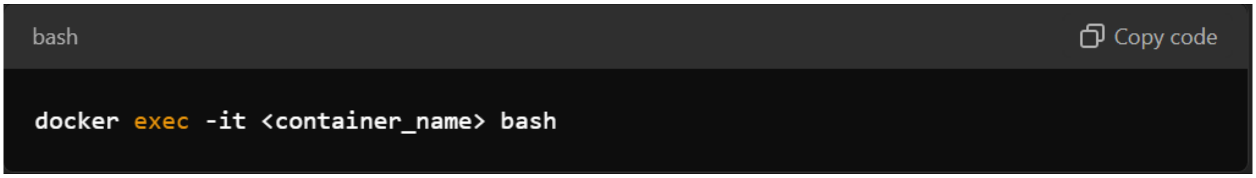
To modify the Wazuh Manager's configuration, first, access the Docker container:

- Identify the running Wazuh Manager container:



```
docker ps
```

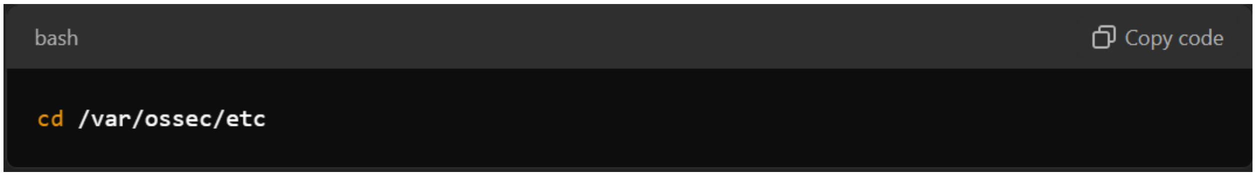
- Access the container’s shell:
- Note the container name or ID.



```
docker exec -it <container_name> bash
```

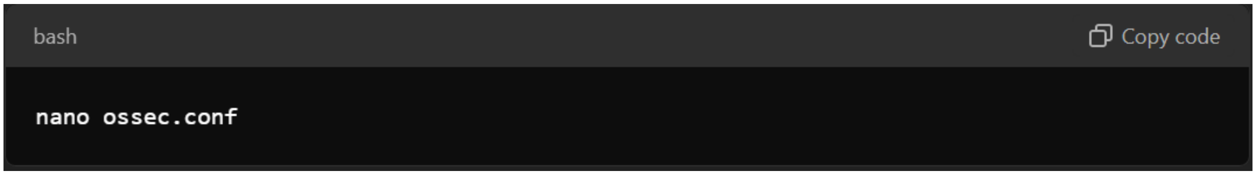
- Replace <container_name> with the name or ID of the Wazuh Manager container.

2. **Navigate to the Configuration Directory**
    - You’re now inside the container. Navigate to the directory containing the ossec.conf file:



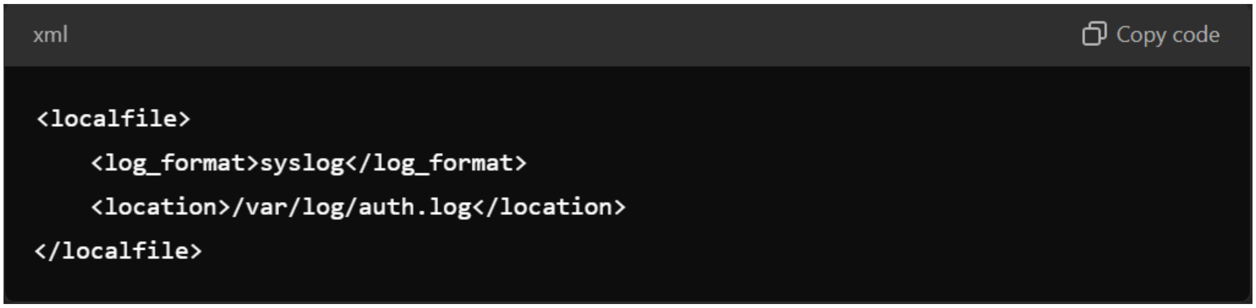
```bash
cd /var/ossec/etc
```

3. **Edit the ossec.conf File**
    - Open the ossec.conf file using a text editor:



```bash
nano ossec.conf
```

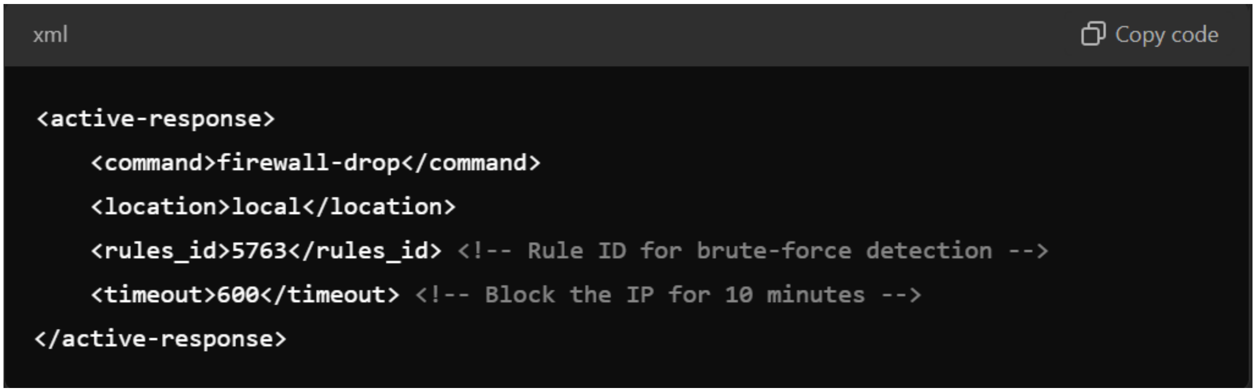
- Add the following <localfile> section to monitor SSH authentication logs:



```xml
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/auth.log</location>
</localfile>
```

- Add an <active-response> section to enable automatic blocking for Rule ID 5763:

4. **Save and Exit**
    - Press Ctrl+O, then Enter to save.

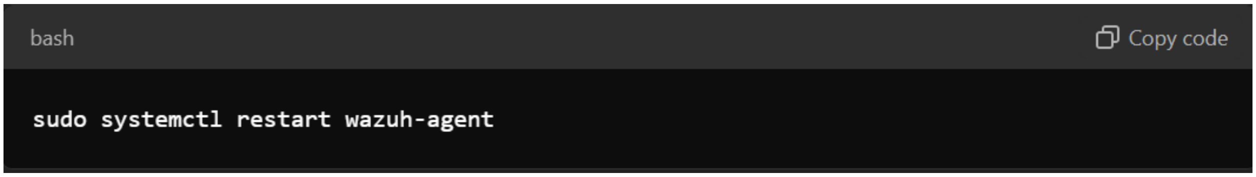


```xml
<active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>5763</rules_id> <!-- Rule ID for brute-force detection -->
    <timeout>600</timeout> <!-- Block the IP for 10 minutes -->
</active-response>
```

- Press Ctrl+X to exit the editor.

5. **Restart the Wazuh Agent**

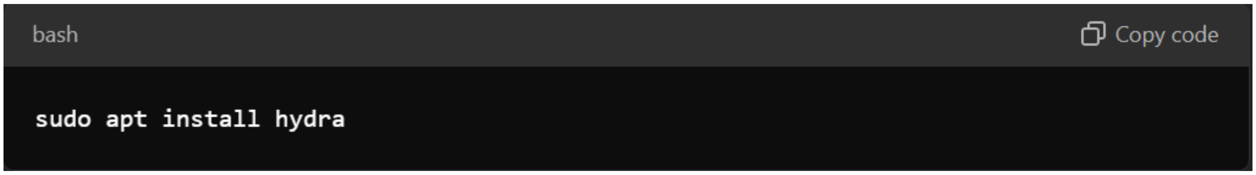
- Apply the changes by restarting the Wazuh Agent service:



```bash
sudo systemctl restart wazuh-agent
```

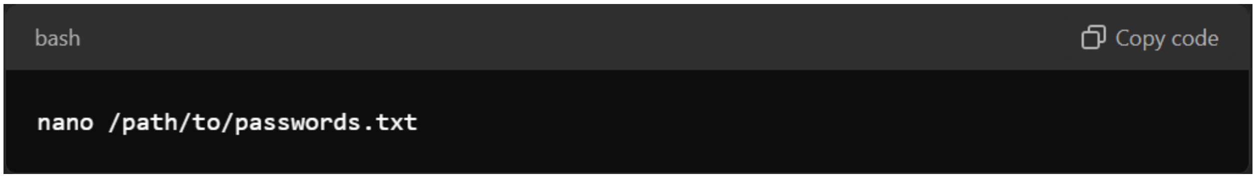
Step 2: Simulating an SSH Brute-Force Attack

1. **Set Up Hydra on the Attacker Machine**
    - Install Hydra on a separate machine (e.g., Kali Linux):



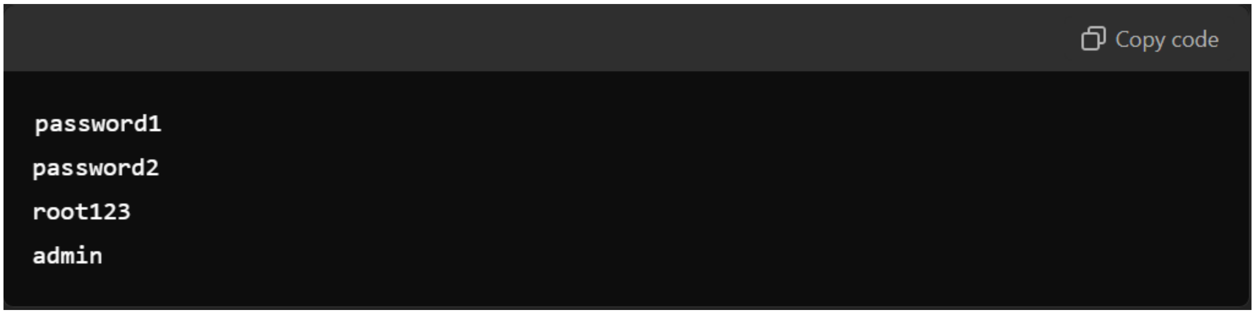
```bash
sudo apt install hydra
```

2. **Launch a Brute-Force Attack**
    - Create or use your custom password list. For example:



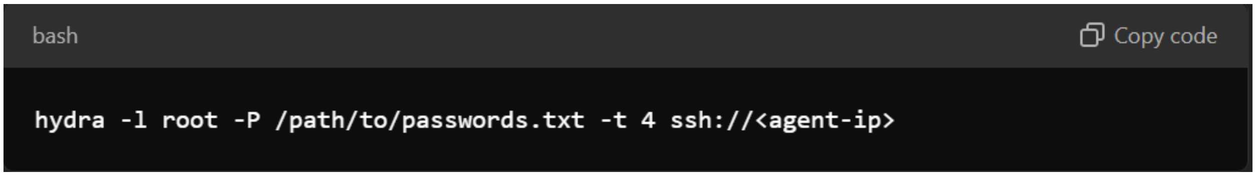
```bash
nano /path/to/passwords.txt
```

- Add a few sample passwords:



```
password1
password2
root123
admin
```

3. Run Hydra to simulate a brute-force attack targeting the SSH service of the Wazuh Agent:



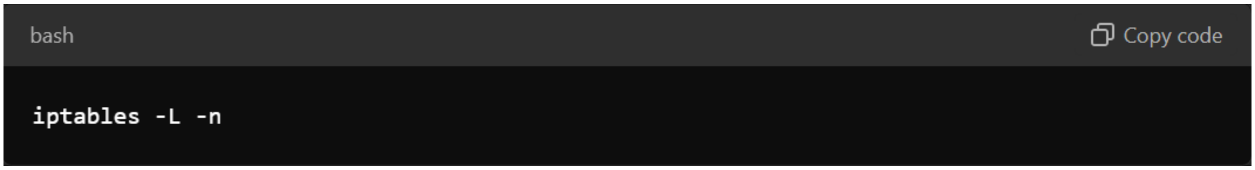
```bash
hydra -l root -P /path/to/passwords.txt -t 4 ssh://<agent-ip>
```

- Replace <agent-ip> with the IP address of the Wazuh Agent.
- Replace /path/to/passwords.txt with the actual path to your password file.
- -t 4 specifies the number of parallel tasks.

4. Observe the attack output to confirm that attempts are being made.

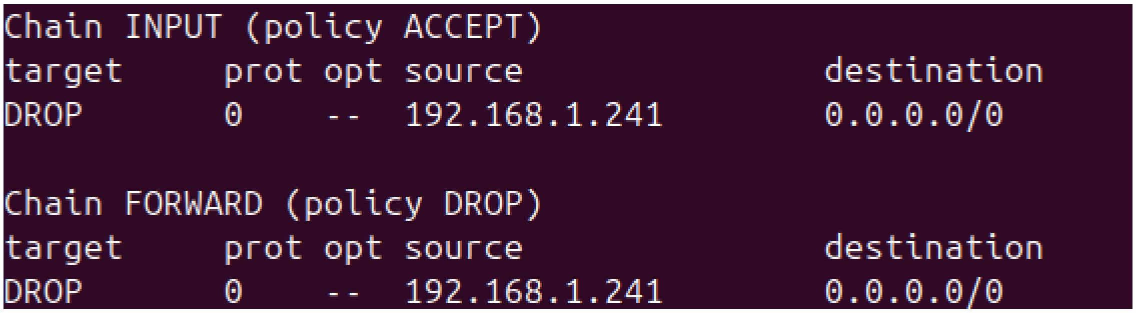
Step 3: Verifying Detection and Blocking

1. **Verify IP Blocking:**
   - Check if the attacking IP has been blocked using the system's firewall:



```bash
iptables -L -n
```

   - You should see an entry like this in the **INPUT** or **FORWARD** chain:



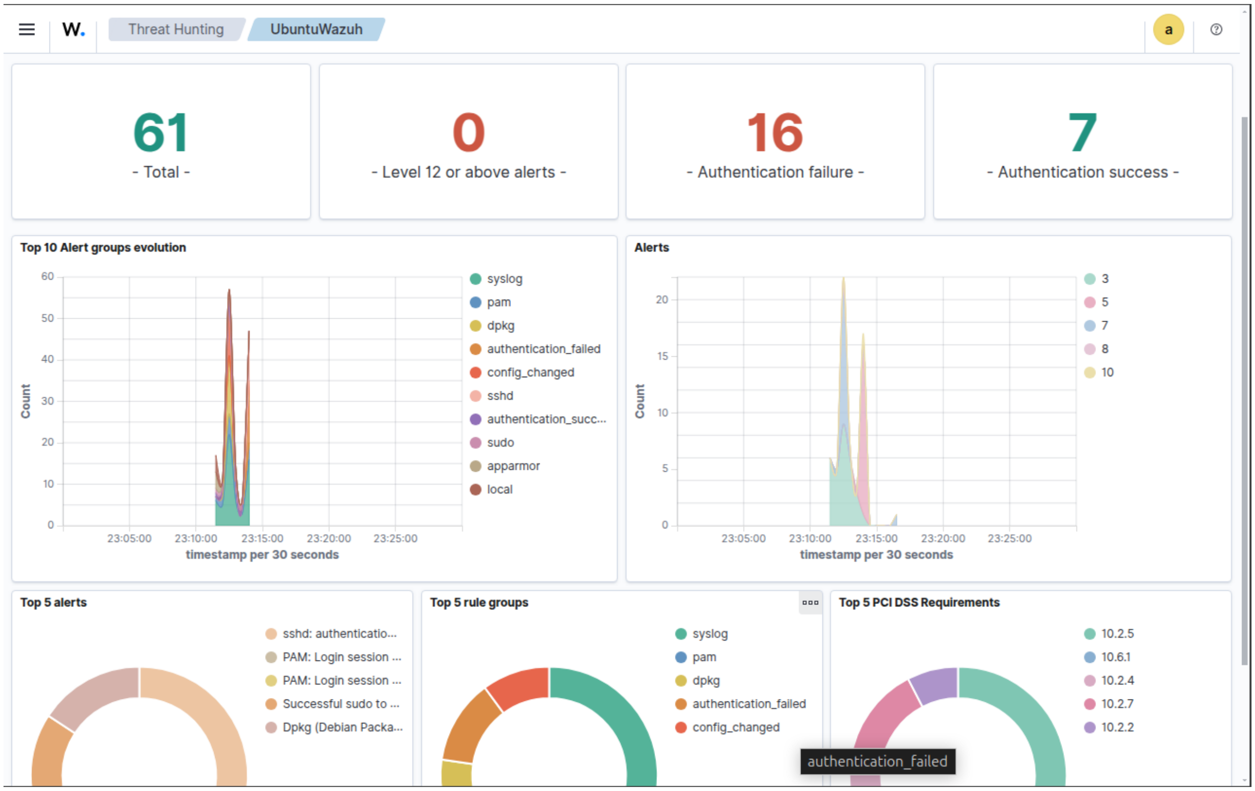
```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       0    --  192.168.1.241        0.0.0.0/0

Chain FORWARD (policy DROP)
target     prot opt source               destination
DROP       0    --  192.168.1.241        0.0.0.0/0
```

2. **Confirm in the Wazuh Dashboard**

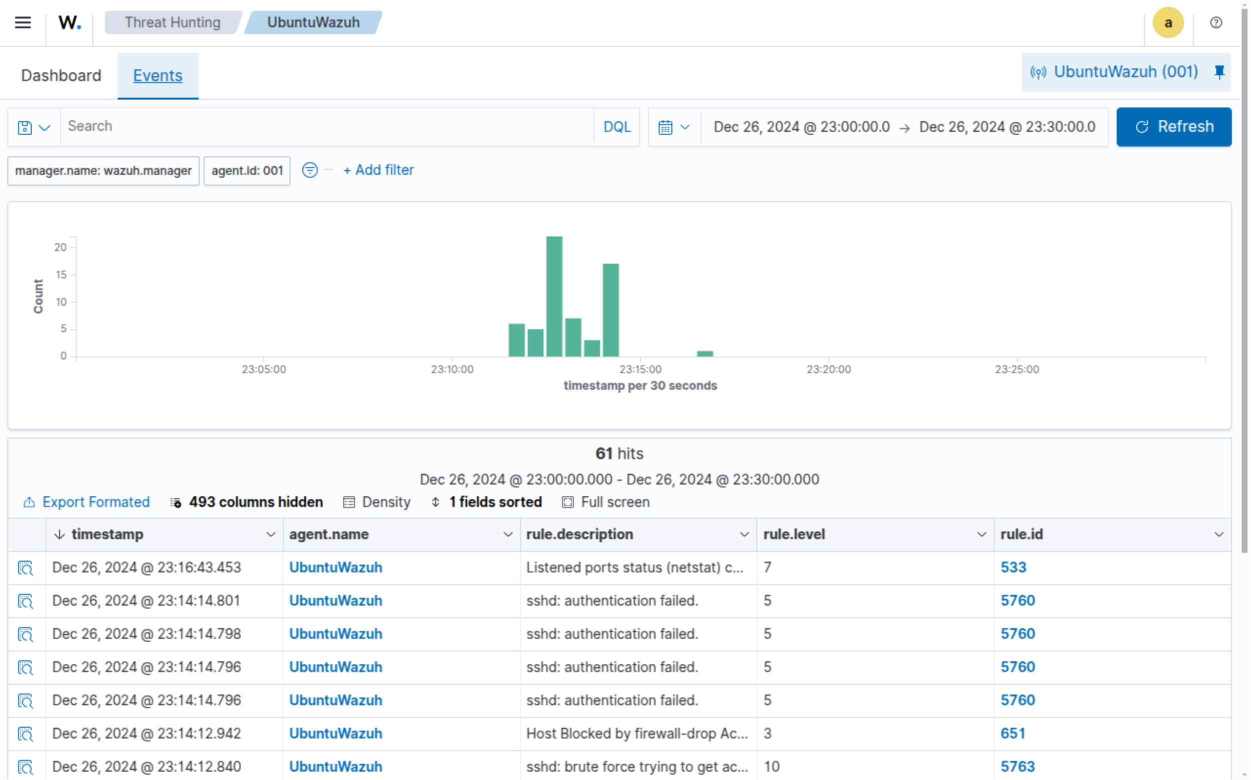
   1. Open the Wazuh Dashboard in your browser:

   2. Navigate to:
      - **Agents** > Select the relevant Agent > **Threat Hunting**.
      - Filter by **Rule ID 5763** to view events related to SSH brute-force detection.

Dashboard view:



Event View:



Dec 26, 2024 @ 23:00:00.0 → Dec 26, 2024 @ 23:30:00.0

61 hits

Dec 26, 2024 @ 23:00:00.000 - Dec 26, 2024 @ 23:30:00.000

timestamp	agent.name	rule.description	rule.level	rule.id
Dec 26, 2024 @ 23:16:43.453	UbuntuWazuh	Listened ports status (netstat) c...	7	533
Dec 26, 2024 @ 23:14:14.801	UbuntuWazuh	sshd: authentication failed.	5	5760
Dec 26, 2024 @ 23:14:14.798	UbuntuWazuh	sshd: authentication failed.	5	5760
Dec 26, 2024 @ 23:14:14.796	UbuntuWazuh	sshd: authentication failed.	5	5760
Dec 26, 2024 @ 23:14:14.796	UbuntuWazuh	sshd: authentication failed.	5	5760
Dec 26, 2024 @ 23:14:12.942	UbuntuWazuh	Host Blocked by firewall-drop Ac...	3	651
Dec 26, 2024 @ 23:14:12.840	UbuntuWazuh	sshd: brute force trying to get ac...	10	5763

Lab 3: Detecting Network Intrusion using Suricata IDS

To configure and integrate Suricata, an Intrusion Detection System (IDS), with Wazuh to monitor network traffic, detect suspicious activities, and centralize alerts.

Step 1: Install Suricata on Wazuh Agent

1. **Add the Suricata Repository**
   - Add the stable Suricata repository:

```bash
sudo add-apt-repository ppa:oisf/suricata-stable
```

   - Update the package list:

```bash
sudo apt-get update
```

2. **Installing Suricata**

```bash
sudo apt-get install suricata -y
```

   - Verify the installation:

```bash
suricata --version
```

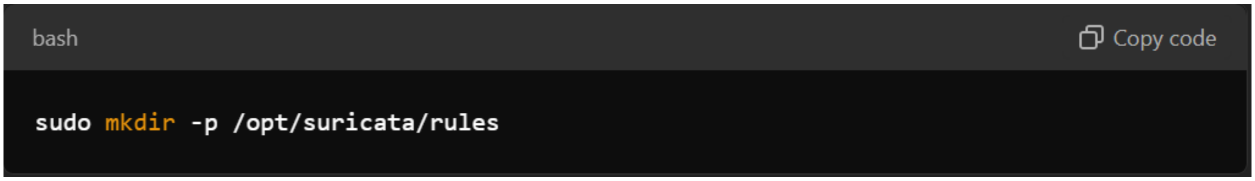
Step 2: Add the Emerging Threats Ruleset

1. **Check Existing Directory List:**
   - List the directories under /etc/suricata to check if the default rules directory exists:

```bash
ls -l /etc/suricata
```

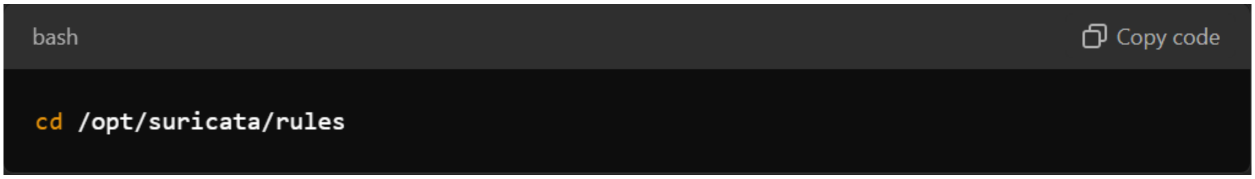
- Look for a rules directory. If it exists, you can proceed to step 3. If it doesn’t exist, create a new custom directory.

2. **Create a Custom Directory for Rules**
   - Create a new directory for storing Suricata rules:



```bash
sudo mkdir -p /opt/suricata/rules
```

3. **Navigate to the custom directory:**



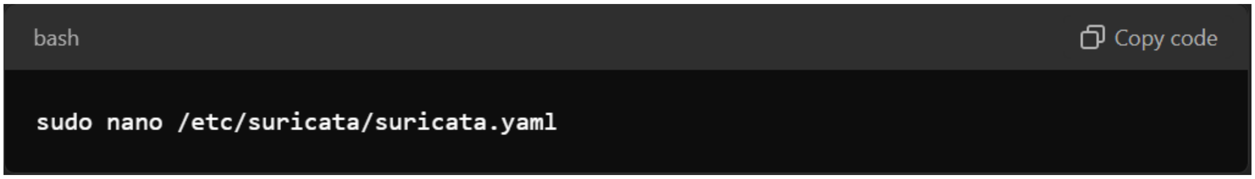
```bash
cd /opt/suricata/rules
```

4. **Download and Extract Rules**
   - cd /tmp/ && curl -LO https://rules.emergingthreats.net/open/suricata-6.0.8/emerging.rules.tar.gz

   - sudo tar -xvzf emerging.rules.tar.gz && sudo mv rules/*.rules /etc/suricata/rules/
   - sudo chmod 640 /etc/suricata/rules/*.rules

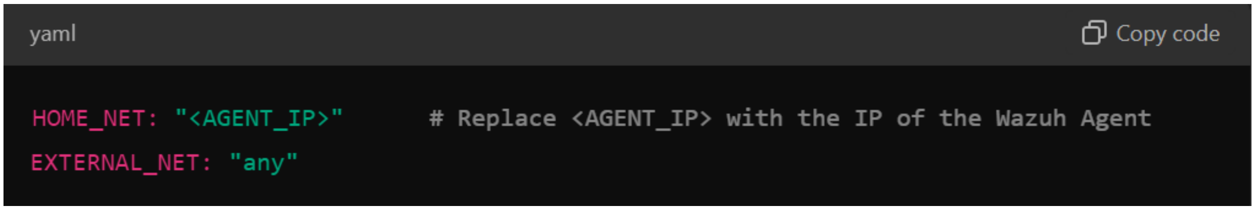
Step 3: Configure Suricata

1. **Modify the Suricata Configuration**
   - Open the suricata.yaml configuration file:



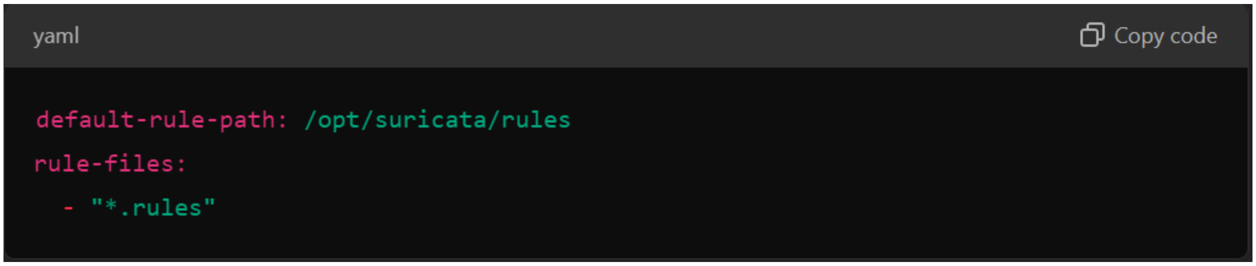
```bash
sudo nano /etc/suricata/suricata.yaml
```

- Update the HOME_NET and EXTERNAL_NET variables:



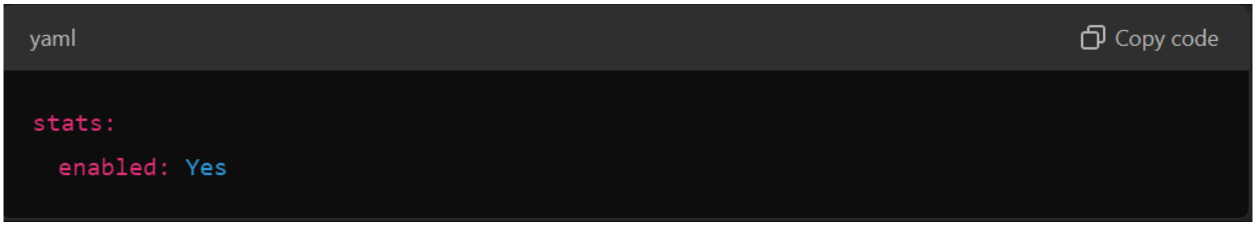
```yaml
HOME_NET: "<AGENT_IP>"      # Replace <AGENT_IP> with the IP of the Wazuh Agent
EXTERNAL_NET: "any"
```

- Configure the default rule path and rule files:



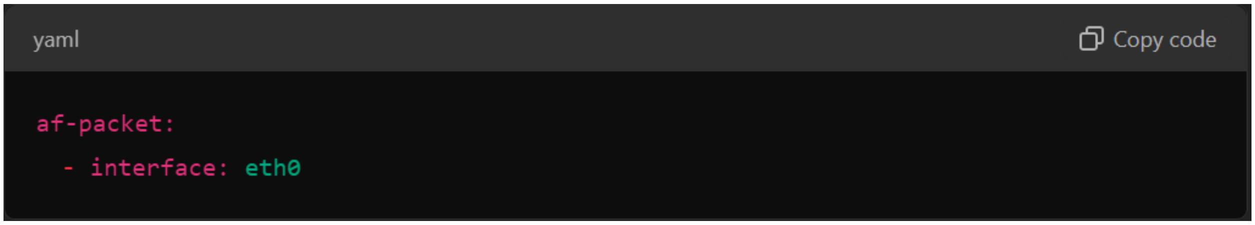
```yaml
default-rule-path: /opt/suricata/rules
rule-files:
  - "*.rules"
```

- Enable global statistics:



```yaml
stats:
  enabled: Yes
```

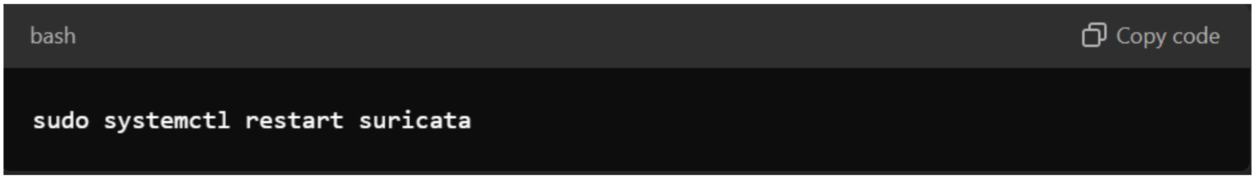
- Enable high-speed packet capture for the network interface:



```yaml
af-packet:
  - interface: eth0
```

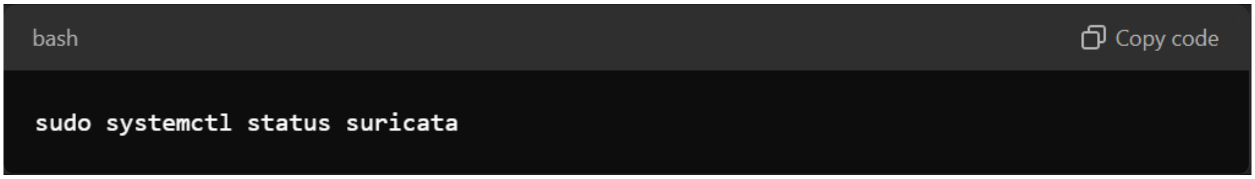
- Replace eth0 with your network interface (use ip a to identify the interface).

2. **Save and exit:**
   - Press Ctrl+O, then Enter to save.
   - Press Ctrl+X to exit.
3. **Restart the Suricata Service**
   - Restart Suricata to apply the changes:



```bash
sudo systemctl restart suricata
```

- Verify the service status:

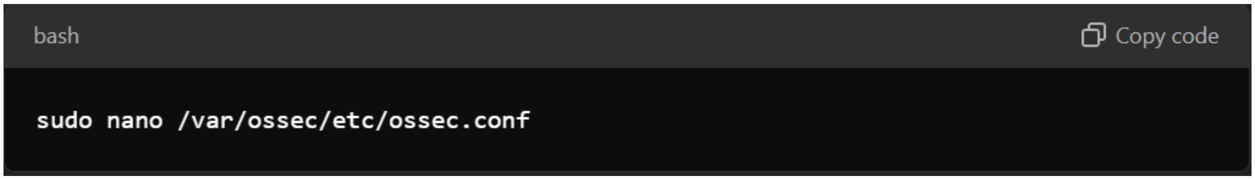


```bash
sudo systemctl status suricata
```

Step 5.4: Integrate Suricata with Wazuh

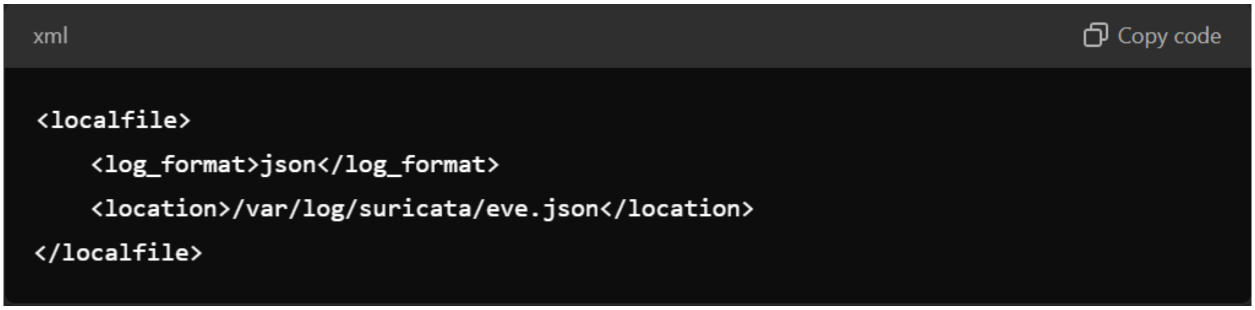
1. **Modify the Wazuh Agent Configuration**

- Open the Wazuh Agent configuration file:



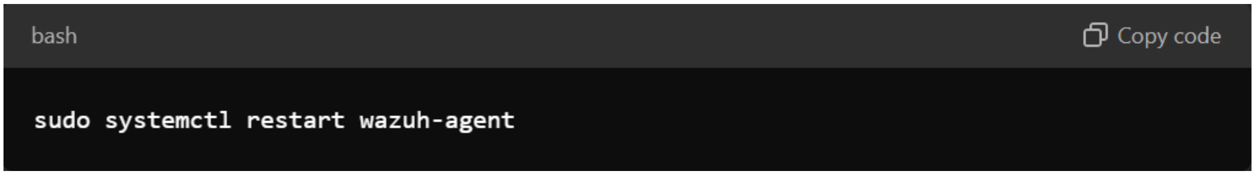
```bash
sudo nano /var/ossec/etc/ossec.conf
```

- Add a <localfile> section to monitor Suricata’s eve.json log file:



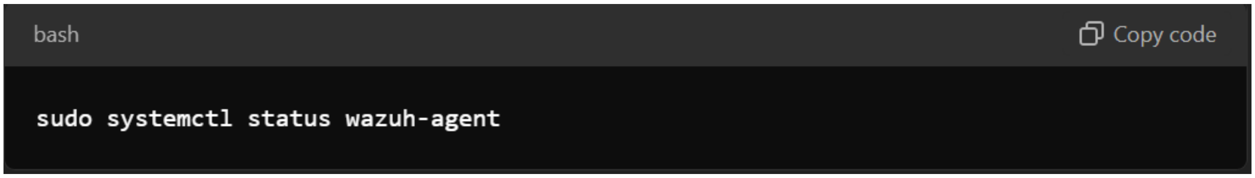
```xml
<localfile>
    <log_format>json</log_format>
    <location>/var/log/suricata/eve.json</location>
</localfile>
```

2. **Save and exit:**
    - Press Ctrl+O, then Enter to save.
    - Press Ctrl+X to exit.
3. **Restart the Wazuh Agent**
    - Restart the Agent to apply the changes:



```bash
sudo systemctl restart wazuh-agent
```

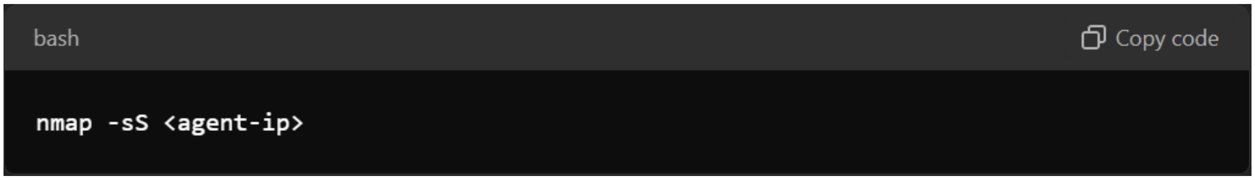
- Verify the status of the Agent:



```bash
sudo systemctl status wazuh-agent
```

Step 5: Simulate Intrusion Attempts

1. **Perform a Port Scan**
    - From an attacker machine (e.g., Kali Linux), use Nmap to perform a port scan:

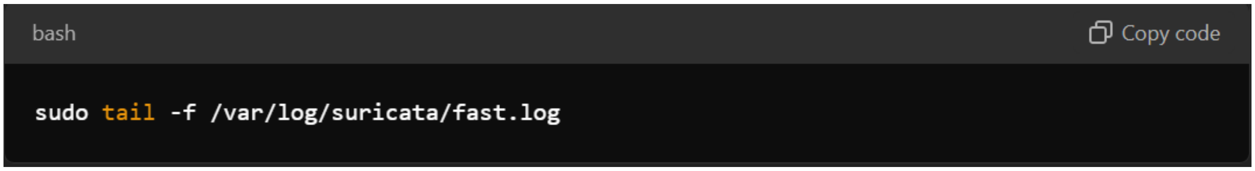


```bash
nmap -sS <agent-ip>
```

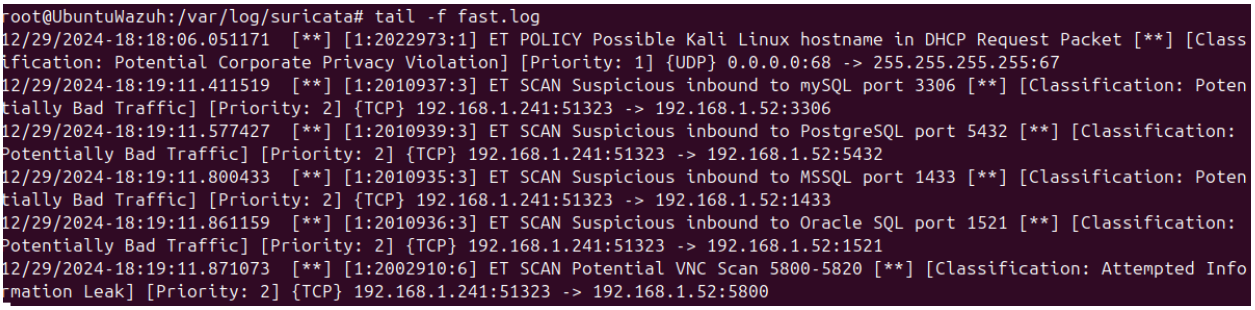
- Replace <agent-ip> with the IP address of the Wazuh Agent running Suricata.

Step 6: Verify Alerts

1. **Check Suricata Logs:**
   - Monitor Suricata's log using fast log on the Wazuh Agent:



```bash
sudo tail -f /var/log/suricata/fast.log
```



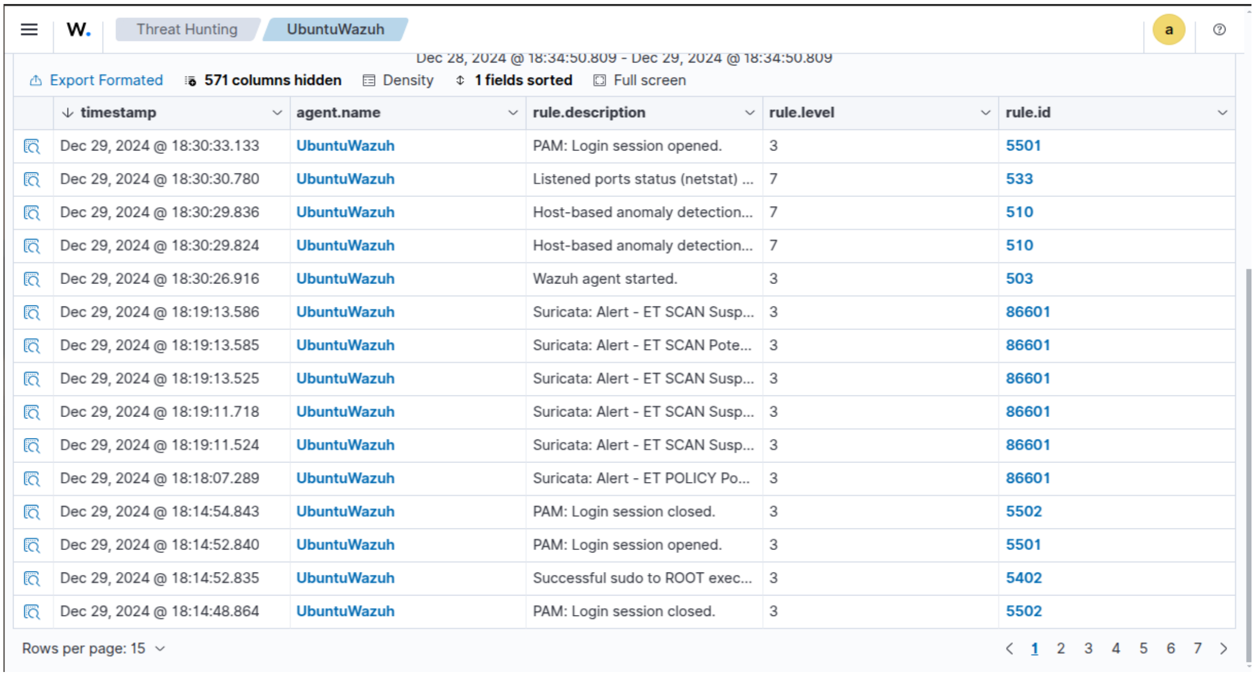
```
root@UbuntuWazuh:/var/log/suricata# tail -f fast.log
12/29/2024-18:18:06.051171  [**] [1:2022973:1] ET POLICY Possible Kali Linux hostname in DHCP Request Packet [**] [Class
ification: Potential Corporate Privacy Violation] [Priority: 1] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
12/29/2024-18:19:11.411519  [**] [1:2010937:3] ET SCAN Suspicious inbound to mySQL port 3306 [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {TCP} 192.168.1.241:51323 -> 192.168.1.52:3306
12/29/2024-18:19:11.577427  [**] [1:2010939:3] ET SCAN Suspicious inbound to PostgreSQL port 5432 [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.1.241:51323 -> 192.168.1.52:5432
12/29/2024-18:19:11.800433  [**] [1:2010935:3] ET SCAN Suspicious inbound to MSSQL port 1433 [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {TCP} 192.168.1.241:51323 -> 192.168.1.52:1433
12/29/2024-18:19:11.861159  [**] [1:2010936:3] ET SCAN Suspicious inbound to Oracle SQL port 1521 [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.1.241:51323 -> 192.168.1.52:1521
12/29/2024-18:19:11.871073  [**] [1:2002910:6] ET SCAN Potential VNC Scan 5800-5820 [**] [Classification: Attempted Info
rmation Leak] [Priority: 2] {TCP} 192.168.1.241:51323 -> 192.168.1.52:5800
```

2. **Verify Alerts in Wazuh Dashboard**

   - Open the Wazuh Dashboard.

   - Navigate to:
     - **Agents** > Select the relevant Agent > **Threat Hunting.**

Event View:



	timestamp	agent.name	rule.description	rule.level	rule.id
	Dec 29, 2024 @ 18:30:33.133	UbuntuWazuh	PAM: Login session opened.	3	5501
	Dec 29, 2024 @ 18:30:30.780	UbuntuWazuh	Listened ports status (netstat) ...	7	533
	Dec 29, 2024 @ 18:30:29.836	UbuntuWazuh	Host-based anomaly detection...	7	510
	Dec 29, 2024 @ 18:30:29.824	UbuntuWazuh	Host-based anomaly detection...	7	510
	Dec 29, 2024 @ 18:30:26.916	UbuntuWazuh	Wazuh agent started.	3	503
	Dec 29, 2024 @ 18:19:13.586	UbuntuWazuh	Suricata: Alert - ET SCAN Susp...	3	86601
	Dec 29, 2024 @ 18:19:13.585	UbuntuWazuh	Suricata: Alert - ET SCAN Pote...	3	86601
	Dec 29, 2024 @ 18:19:13.525	UbuntuWazuh	Suricata: Alert - ET SCAN Susp...	3	86601
	Dec 29, 2024 @ 18:19:11.718	UbuntuWazuh	Suricata: Alert - ET SCAN Susp...	3	86601
	Dec 29, 2024 @ 18:19:11.524	UbuntuWazuh	Suricata: Alert - ET SCAN Susp...	3	86601
	Dec 29, 2024 @ 18:18:07.289	UbuntuWazuh	Suricata: Alert - ET POLICY Po...	3	86601
	Dec 29, 2024 @ 18:14:54.843	UbuntuWazuh	PAM: Login session closed.	3	5502
	Dec 29, 2024 @ 18:14:52.840	UbuntuWazuh	PAM: Login session opened.	3	5501
	Dec 29, 2024 @ 18:14:52.835	UbuntuWazuh	Successful sudo to ROOT exec...	3	5402
	Dec 29, 2024 @ 18:14:48.864	UbuntuWazuh	PAM: Login session closed.	3	5502

Conclusion:

The "Endpoint Threat Monitoring and Response Using Wazuh" project demonstrates the importance of proactive endpoint security in protecting organizational assets. By setting up Wazuh Manager and Agents, implementing File Integrity Monitoring, and integrating Suricata IDS, the project showcases Wazuh's effectiveness in detecting and responding to security threats.

Real-world scenarios, such as blocking SSH brute-force attacks and monitoring network intrusions, validated the system's capabilities. This project serves as a solid foundation for further exploration, such as scaling deployments or integrating Wazuh with external SIEM tools.

In summary, this work highlights Wazuh as a reliable solution for enhancing endpoint security in a dynamic threat landscape.