Wazuh SIEM & File Integrity Monitoring Deployment

Adewole Adebayo

25\02\2026

1. Introduction

Wazuh is a free, open-source security platform that unifies Security Information and Event Management (SIEM) and Extended Detection and Response (XDR) capabilities. It provides real-time threat detection, monitoring, and incident response for endpoints, cloud workloads, and on-premises environments.

2. Importance

- **Early Breach Detection:** FIM acts as an early warning system. Unauthorized changes to system files, registry keys, or log files are often the first signs of a malware infection, ransomware, or an active hacker attempting to hide their tracks.
- **Regulatory Compliance:** Many industry standards mandate FIM to protect sensitive data. Key regulations include:

    - **PCI DSS:** Required for organizations handling credit card data.
    - **HIPAA:** Essential for protecting electronic health records.

    - **GDPR:** Necessary for ensuring the integrity of personal data.
- **Detection of Insider Threats:** While firewalls and antivirus tools focus on external threats, FIM tracks changes made by internal users. This identifies accidental or malicious modifications made by individuals who already have legitimate access to the system.
- **Incident Response & Forensics:** When a breach occurs, FIM provides a detailed "who, what, when, and how" audit trail. This data is vital for forensic investigators to determine the scope of the damage and restore systems to a trusted state.

3. Goal

- To Detect Unauthorized File Changes
- Alert Tuning & Investigation

1. Documenting VirtualBox Setup

1.1 Virtualization Platform

Platform: VirtualBox
Host OS: Windows 11

Host RAM: 16GB
Purpose: Create isolated environment for SIEM deployment and agent testing, and file monitoring.

1.2 Virtual Machine Configuration

VM 1 – Wazuh Manager (Ubuntu Server)

- OS: Ubuntu Server 22.04 LTS
- RAM: 6 GB
- CPU: 4 cores
- Storage: 70 GB dynamically allocated
- Network Mode: Bridged Adapter

VM 2 – Windows Server Agent

- OS: Windows Server 2025
- RAM: 4 GB
- CPU: 2 cores
- Storage: 50 GB
- Network Mode: Bridged Adapter

1.3 Why Bridged Network?

Bridged networking was selected to:

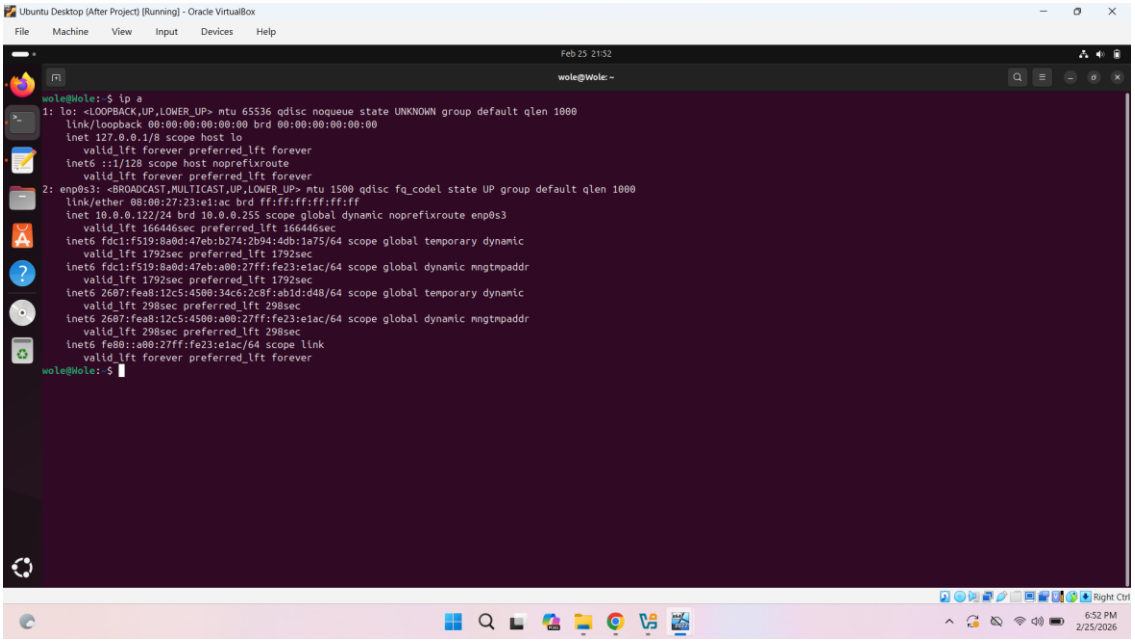
- Allow both VMs to obtain IP addresses from the same LAN
- Enable direct communication between manager and agent
- Simulate real enterprise network behavior

2. Documenting Network Configuration

2.1 IP Address Verification

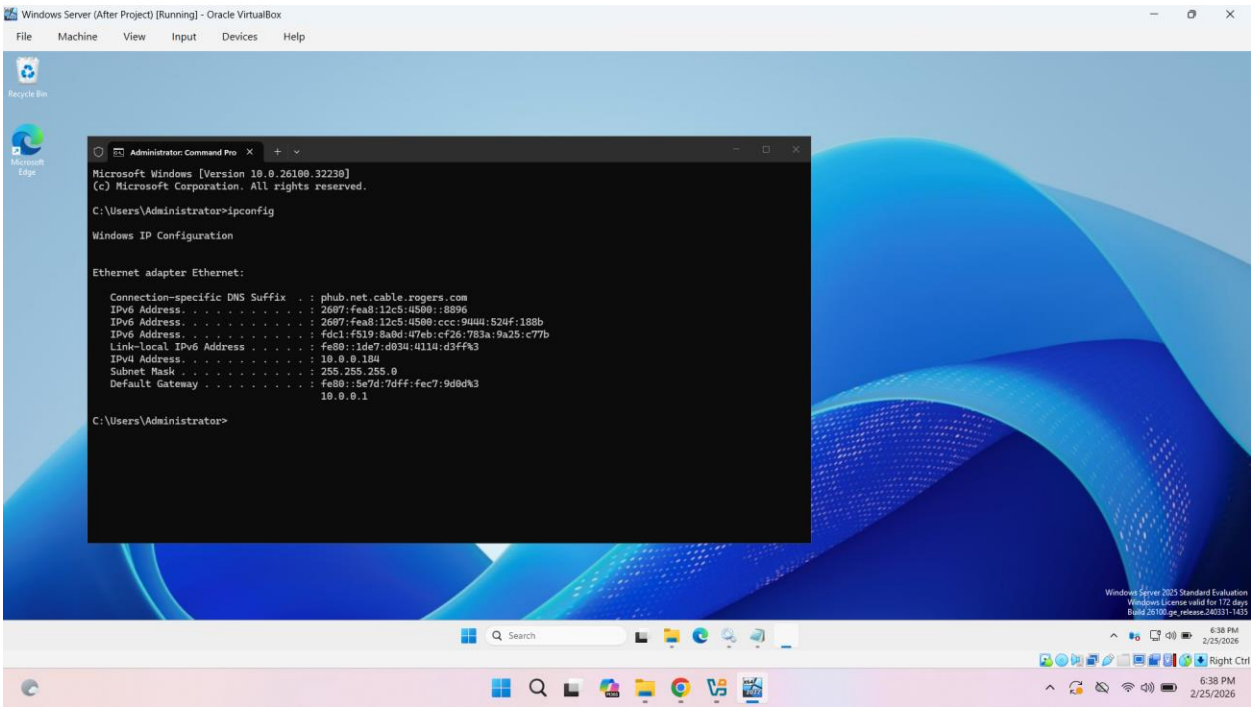
On Ubuntu:

ip a



On Windows:

Ipconfig



- Ubuntu Manager: 10.0.0.122
- Windows Agent: 10.0.0.184
- Both must be on the same subnet mask to communicate

2.2 Connectivity Testing

Tested communication using **ping**:

On Windows:

ping 10.0.0.122

On Ubuntu:

ping 10.0.0.184

Verified port 1514 (Wazuh agent communication port) was reachable.

3. Documenting Ubuntu Installation

3.1 ISO Installation

- Downloaded Ubuntu Server 22.04 ISO on official website
- Attached ISO in VirtualBox
- Booted VM and completed installation wizard

3.2 Initial Configuration

- Created admin user
- Configured network (automatic DHCP)

3.3 System Update

- **sudo apt update && sudo apt upgrade –y**
- **sudo apt install curl**

*Ensured system packages were fully updated before Wazuh installation.

4. Documenting Wazuh Manager Installation

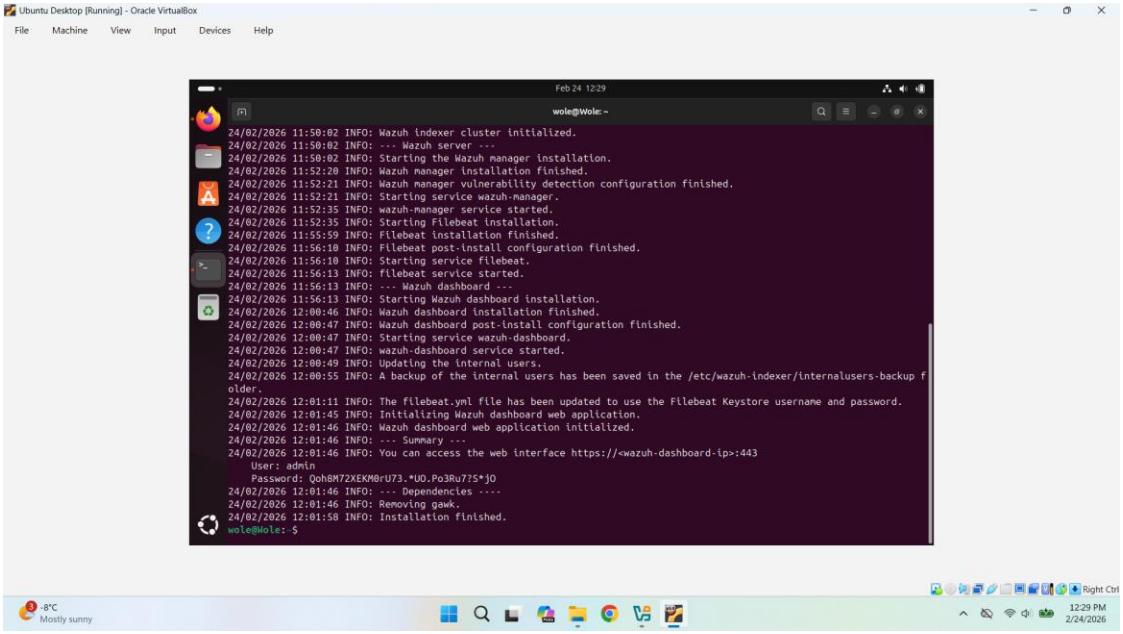
4.1 Installation Method

Installed Wazuh manager using official installation script from Wazuh repository.

curl -sO https://packages.wazuh.com/4.x/wazuh-install.sh
sudo bash wazuh-install.sh -a

This installed:

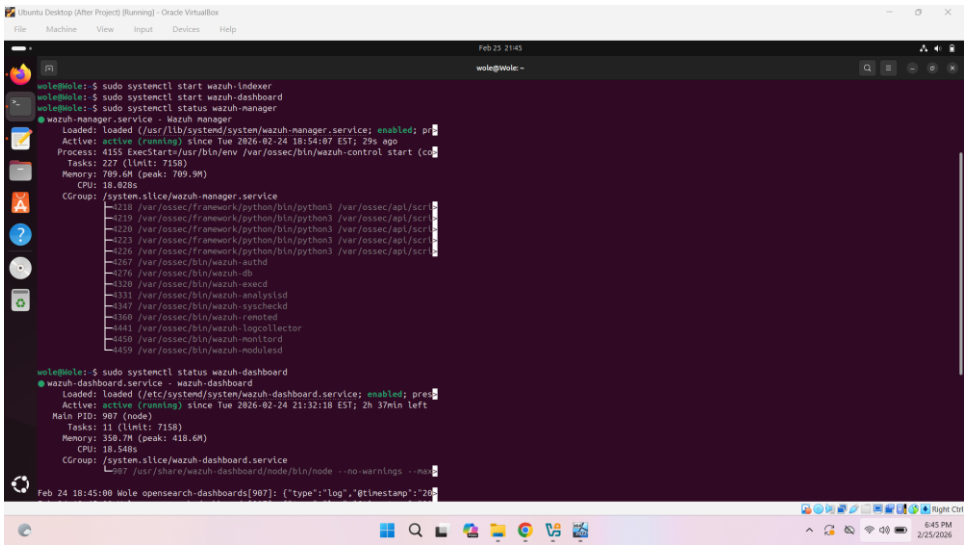
- Wazuh Manager
- Wazuh Indexer
- Wazuh Dashboard
  *After installation, login credentials are generated and these are used to login on the dashboard

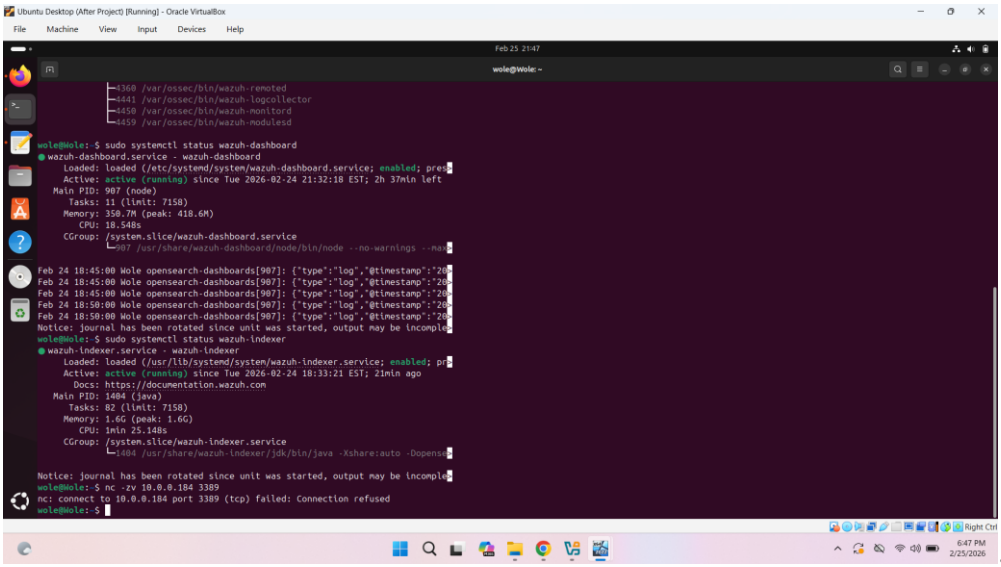


4.2 Service Verification

Checked services:

sudo systemctl status wazuh-manager
sudo systemctl status wazuh-indexer
sudo systemctl status wazuh-dashboard



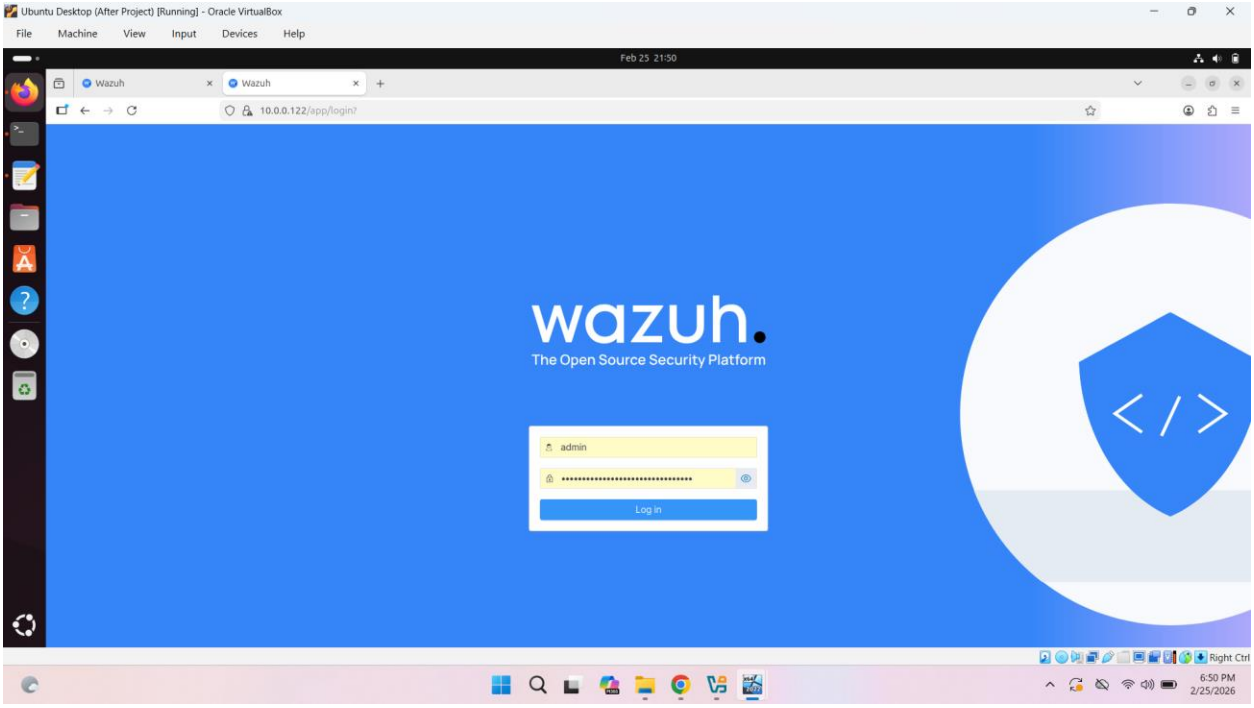


* Confirmed that all services were running.

4.3 Dashboard Access

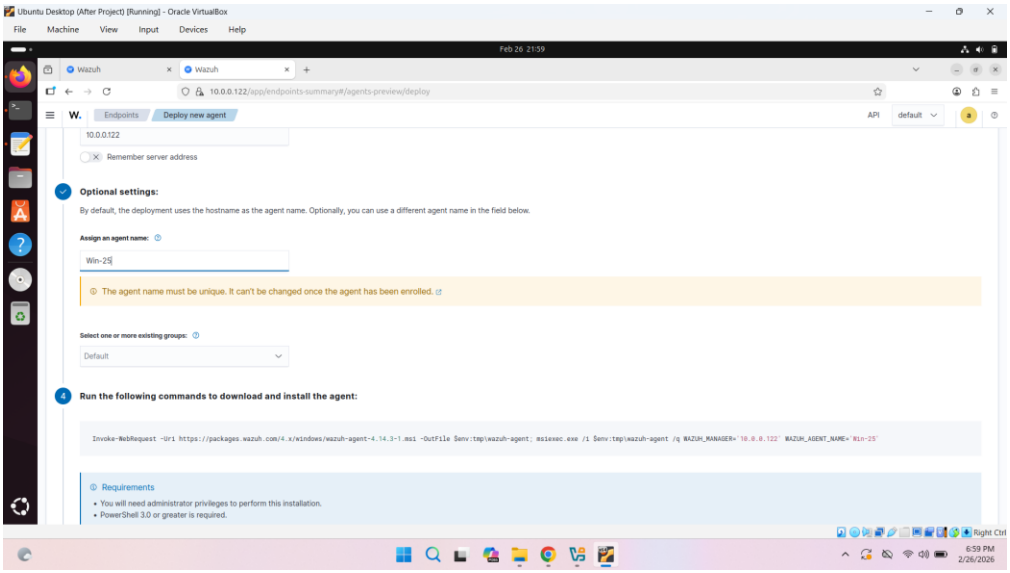
Accessed dashboard via:

https://10.0.0.122



Logged in using generated credentials.

4.4 Agent Deployment



Deployed a new agent on Windows server using the command from the Wazuh dashboard on Powershell

Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.14.3-1.msi -OutFile $env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='10.0.0.122' WAZUH_AGENT_NAME='Win-25'

Architecture Explanation

The Ubuntu Server hosts the Wazuh Manager, which receives logs and FIM alerts from the Windows Server agent over TCP port 1514. Alerts are indexed and visualized via the Wazuh Dashboard.

Design Considerations

- Allocated sufficient RAM and Storage to prevent indexer crash
- Used bridged networking to simulate enterprise deployment
- Reduced FIM scan interval to 60 seconds for rapid lab validation
- Segmented monitoring responsibilities between manager and agent

5. Configuration

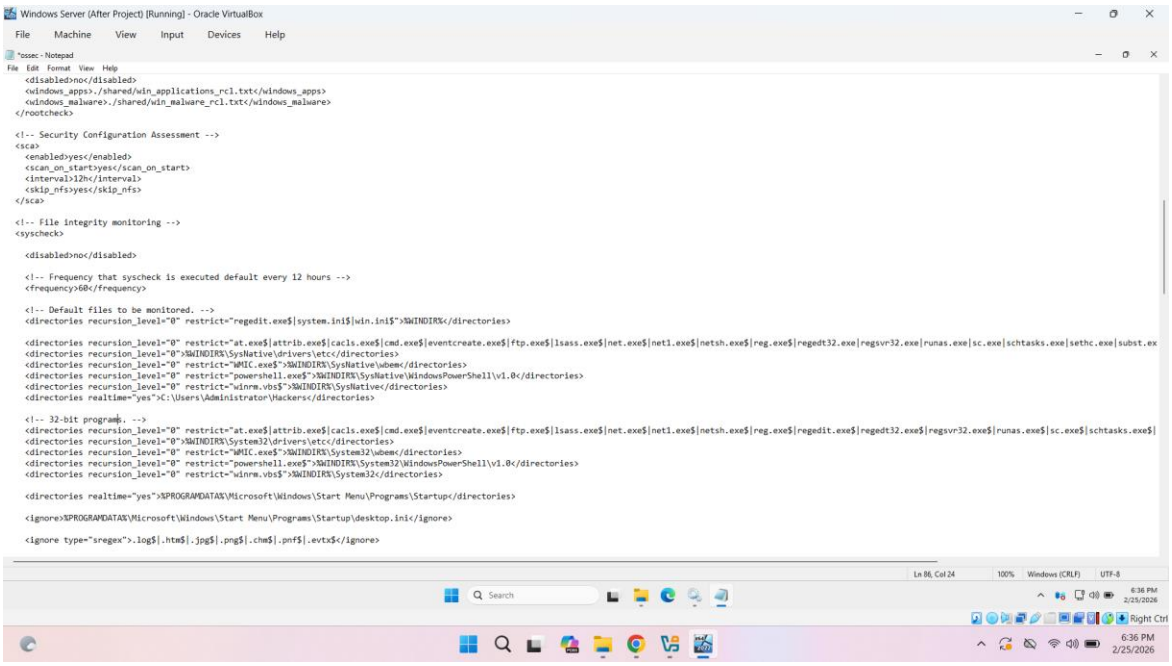
<syscheck>

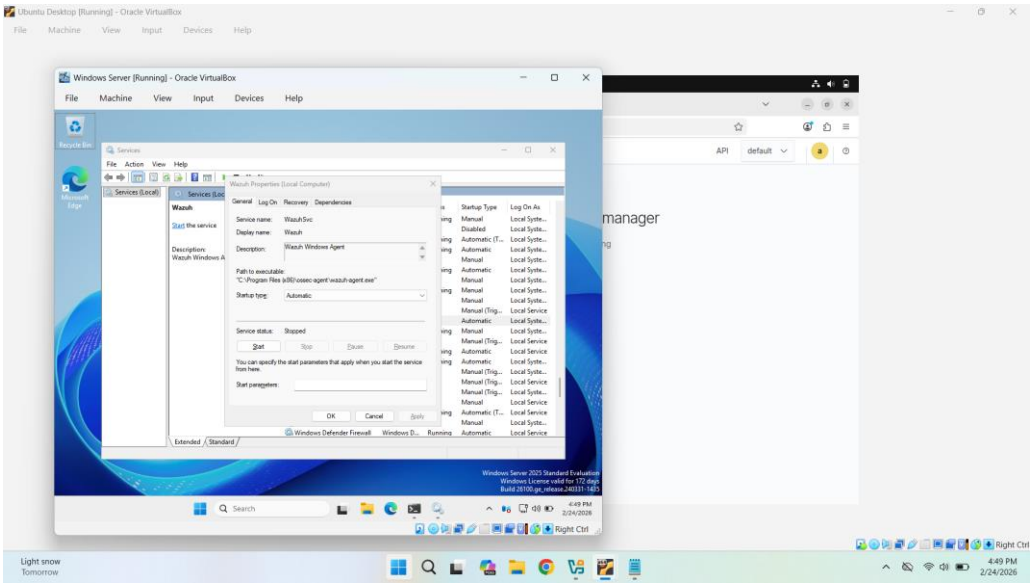
<disabled>no</disabled>

<frequency>60</frequency>

<directories realtime="yes">C:\Users\Administrator\Hackers</directories>

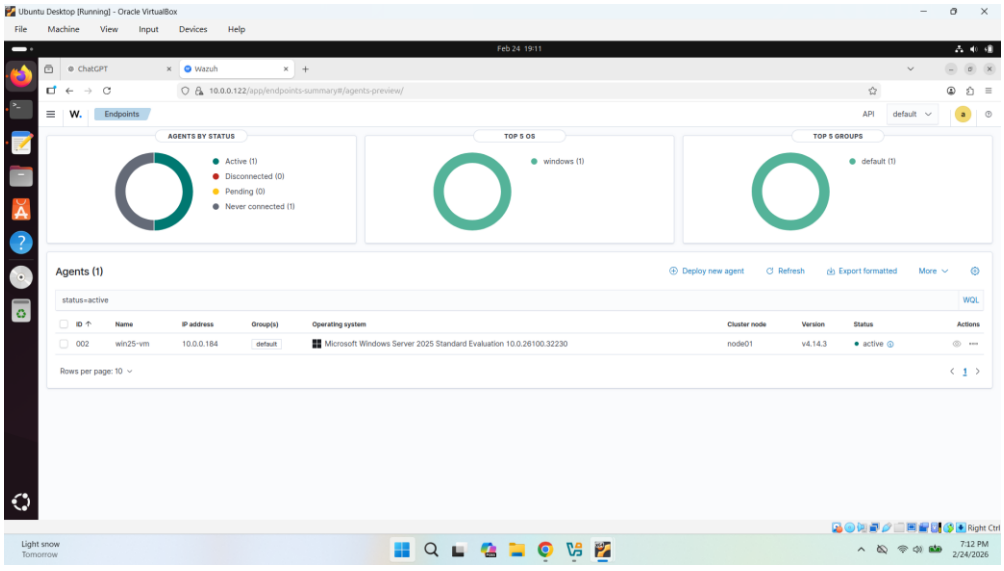
<syscheck>



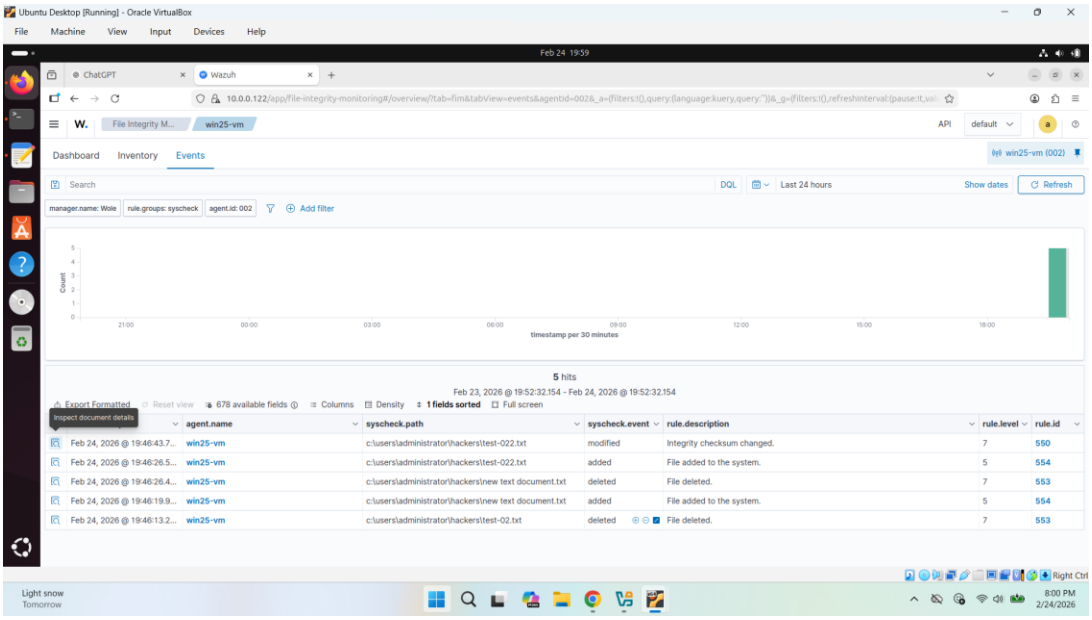


- Scan frequency set to 60 seconds so alerts can be quickly generated when changes are made to a file
- New Directory was created for solely monitoring the changes made to it
- Restarted Wazuh service so Wazuh reloads updated **ossec.conf** file

6. Testing & Validation



Dashboard showing Agent is active



File Creation

When a new file is created in a monitored directory, **Wazuh** detects that the file did not exist in its previous baseline scan. It records the file's metadata and hash, then generates an **"added"** alert as shown in the screenshot above.

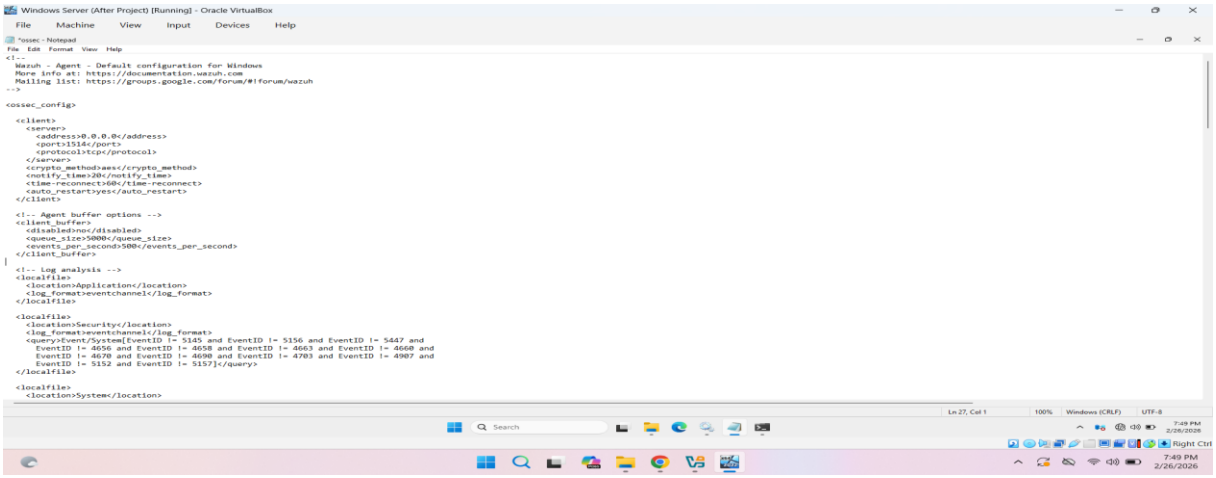
File Modification

When a monitored file is changed, Wazuh recalculates its hash and compares it to the stored baseline value. If the hash or attributes differ, it generates a **"modified"** alert indicating possible tampering as shown in the screenshot above.

File Deletion

When a previously recorded file is removed, Wazuh detects that it is missing during the next scan. It then generates a **"deleted"** alert showing the file path and prior recorded details as shown in the screenshot above.

7. Troubleshooting



At first, the deployed agent wasn't showing up on the Wazuh dashboard; it was because the IP address in the ossec file was set to 0.0.0.0 by default. Edited the IP address to 10.0.0.122, the Ubuntu Manager address to fix this.

Conclusion

What I Learned

I learned how to configure and validate File Integrity Monitoring using **Wazuh**, including editing the <syscheck> configuration and adjusting scan frequency. I also learned how file hashes, timestamps, and baselines are used to detect changes and generate security alerts.

Additionally, I gained hands-on experience analyzing "added," "modified," and "deleted" alerts in the Wazuh dashboard, which improved my understanding of how file-level monitoring works in a SIEM environment.

Why FIM Matters in Real SOC Environments

File Integrity Monitoring is critical because attackers often modify, create, or delete files after gaining access to a system. Monitoring these changes helps SOC analysts detect malware drops, configuration tampering, web shell uploads, and log deletion attempts.

In real SOC environments, FIM provides visibility into unauthorized system changes and supports compliance requirements (such as monitoring sensitive system files). It acts as an early warning system for post-compromise activity.

How This Simulates Real-World Attack Detection

By creating, modifying, and deleting test files, I simulated behaviors commonly seen during real cyberattacks. For example, attackers may create malicious scripts, modify configuration files for persistence, or delete logs to cover their tracks.

This lab demonstrates how a SOC team would receive alerts from Wazuh and investigate suspicious file activity, replicating real-world detection, and response workflows.