SOC Analyst Homelab — Wazuh SIEM with Windows Agent

📌 Project Overview

This homelab demonstrates how a **Security Operations Center (SOC) workflow** can be built and practiced using:

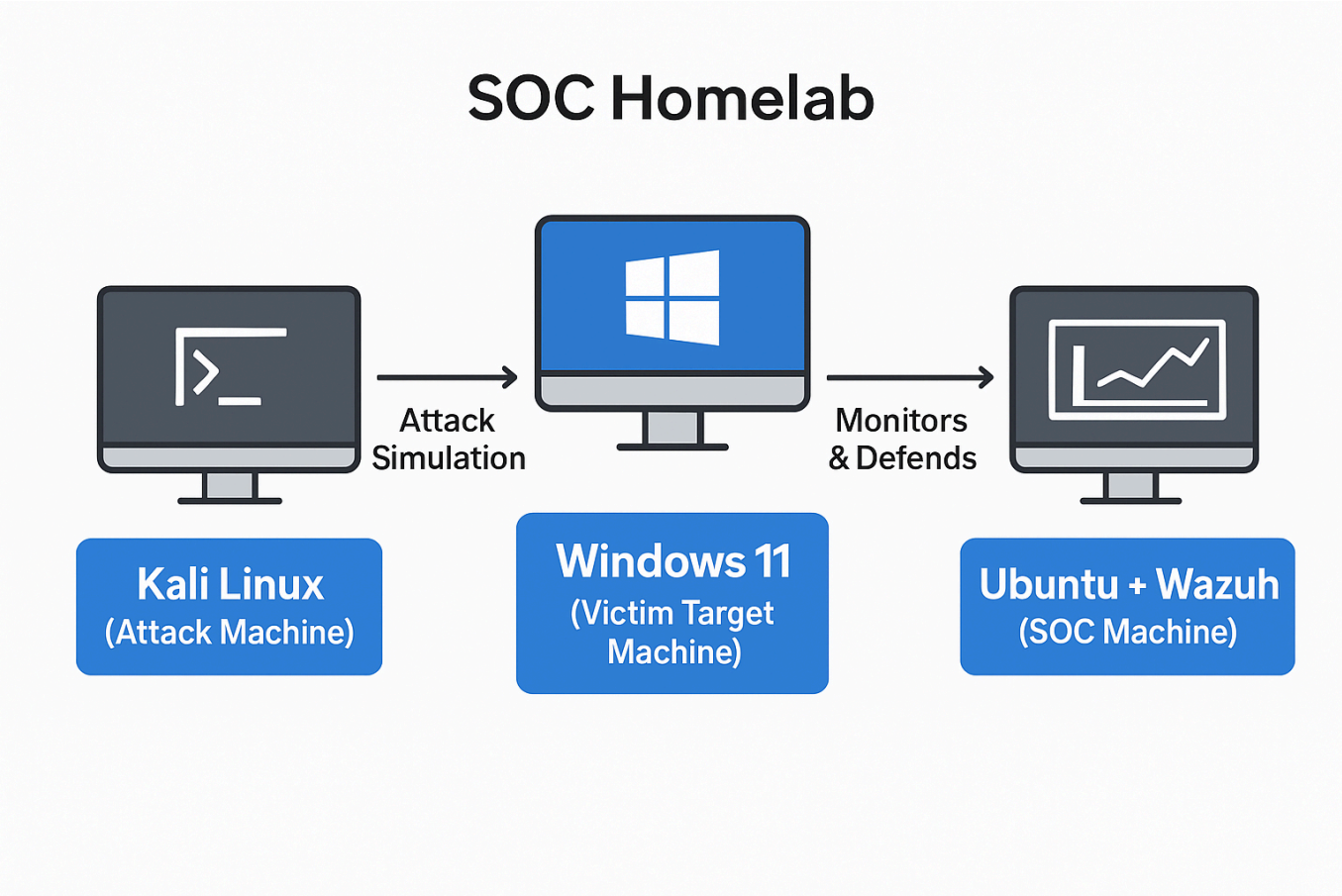
- **Ubuntu Server** → Hosting Wazuh SIEM
- **Kali Linux** → Attack simulation machine
- **Windows 11** → Victim endpoint with Wazuh agent

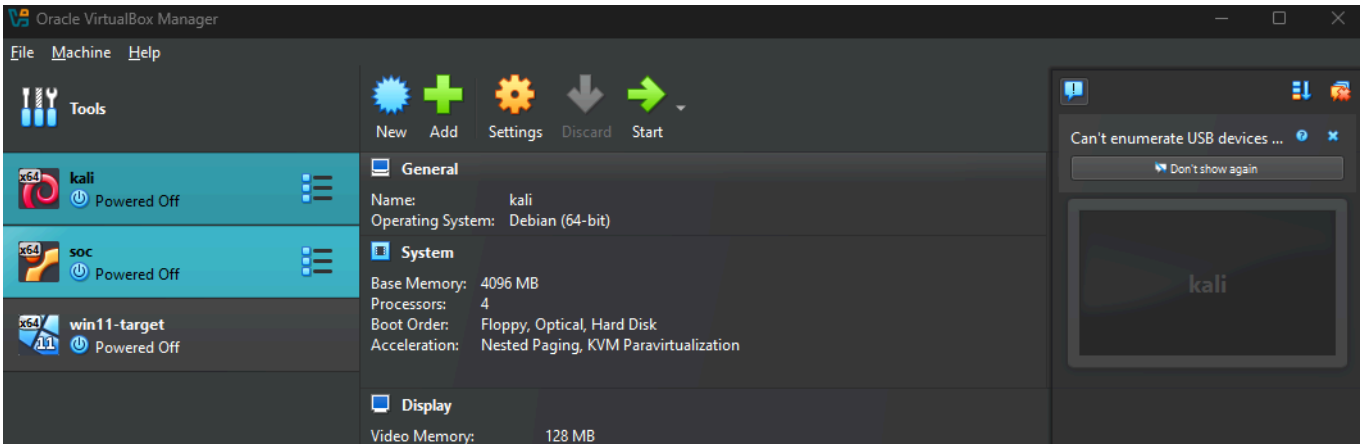
The goal of this project is to **monitor endpoint activity, detect threats, and analyze logs** using Wazuh in a realistic setup.

⚙️ Lab Architecture

Components:

- **Ubuntu Server (Wazuh Manager + Dashboard)**
- **Windows 11 Endpoint (Wazuh Agent installed & connected)**
- **Kali Linux (Attacker machine for generating events)**

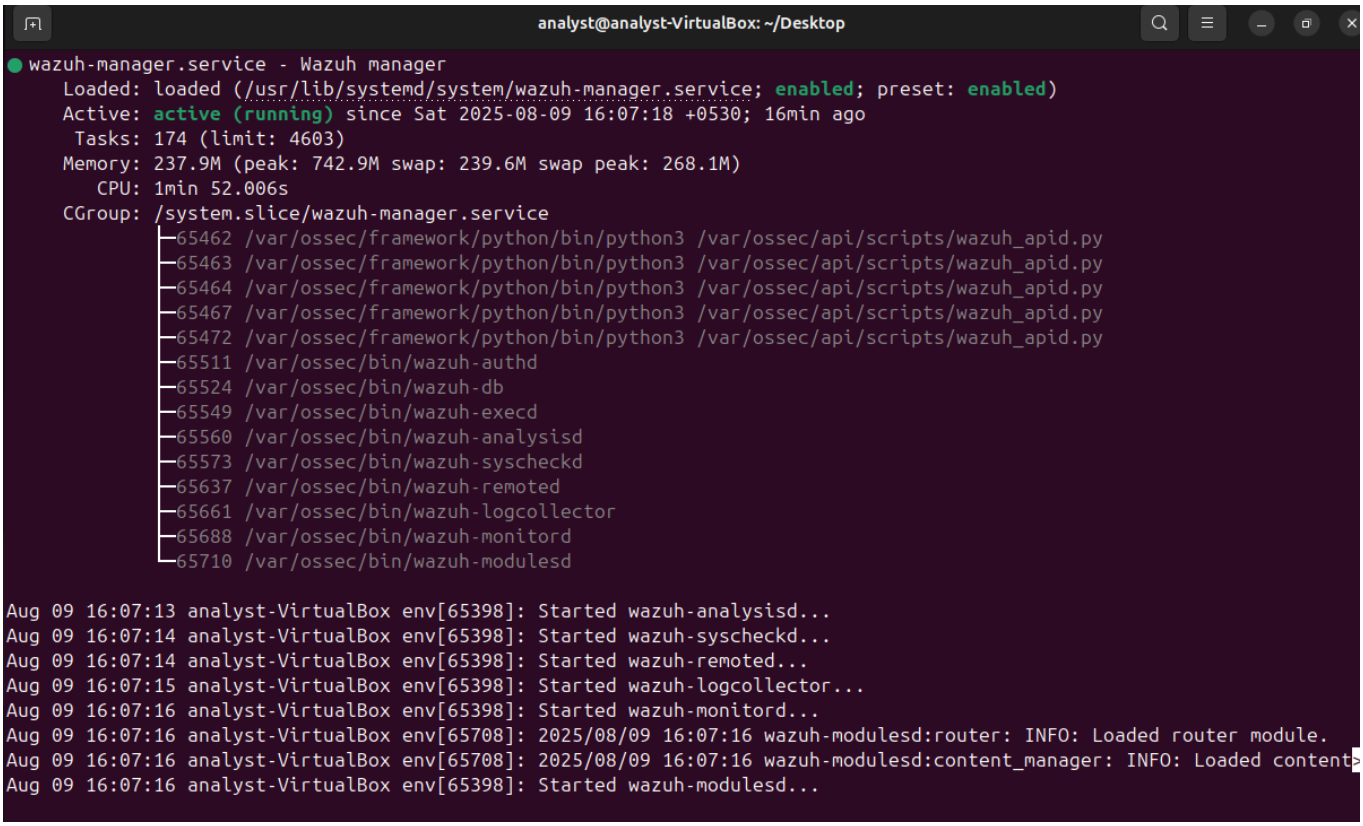


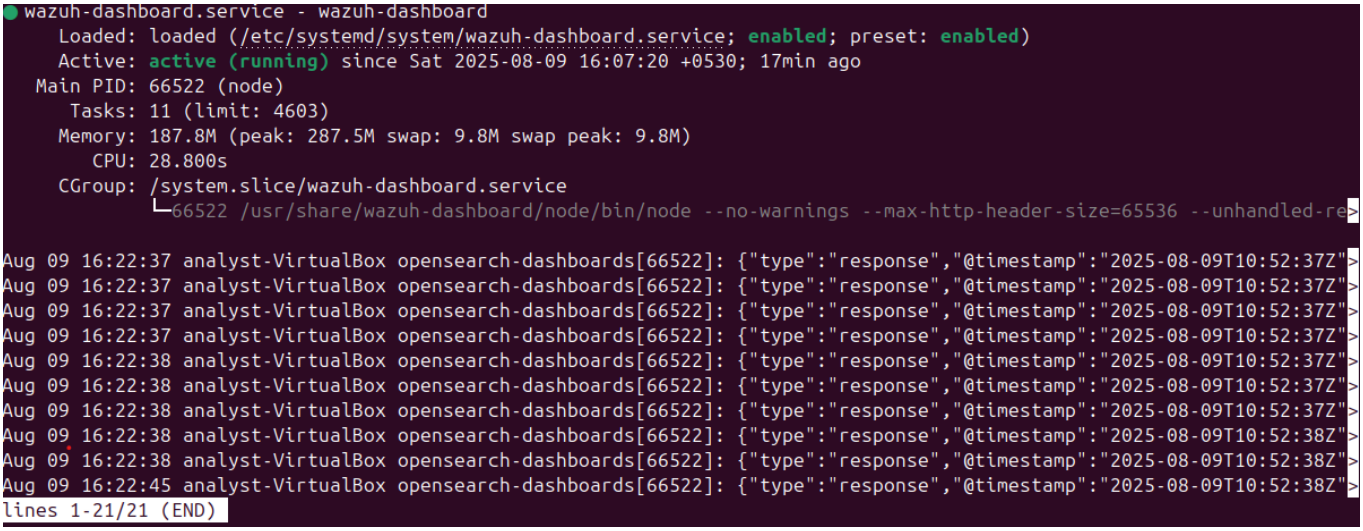


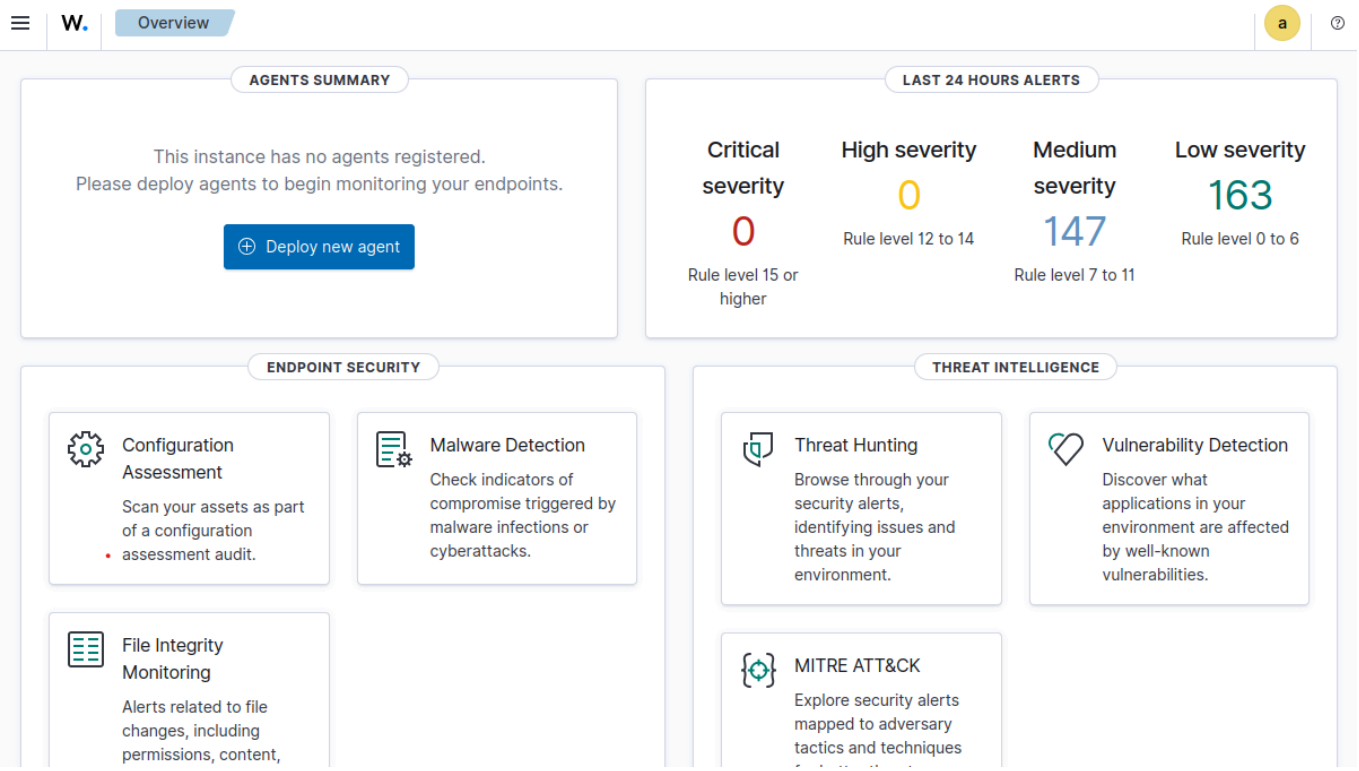
🔑 Steps Implemented

1. Installed Wazuh on Ubuntu

- Deployed **Wazuh Manager + Dashboard** on Ubuntu Server VM.
- Verified that the **Wazuh dashboard** was accessible via web browser.


analyst@analyst-VirtualBox: ~/Desktop
wazuh-manager.service - Wazuh manager
Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; preset: enabled)
Active: active (running) since Sat 2025-08-09 16:07:18 +0530; 16min ago
Tasks: 174 (limit: 4603)
Memory: 237.9M (peak: 742.9M swap: 239.6M swap peak: 268.1M)
CPU: 1min 52.006s
CGroup: /system.slice/wazuh-manager.service
65462 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
65463 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
65464 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
65467 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
65472 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
65511 /var/ossec/bin/wazuh-authd
65524 /var/ossec/bin/wazuh-db
65549 /var/ossec/bin/wazuh-execd
65560 /var/ossec/bin/wazuh-analysisd
65573 /var/ossec/bin/wazuh-syscheckd
65637 /var/ossec/bin/wazuh-remoted
65661 /var/ossec/bin/wazuh-logcollector
65688 /var/ossec/bin/wazuh-monitord
65710 /var/ossec/bin/wazuh-modulesd
Aug 09 16:07:13 analyst-VirtualBox env[65398]: Started wazuh-analysisd...
Aug 09 16:07:14 analyst-VirtualBox env[65398]: Started wazuh-syscheckd...
Aug 09 16:07:14 analyst-VirtualBox env[65398]: Started wazuh-remoted...
Aug 09 16:07:15 analyst-VirtualBox env[65398]: Started wazuh-logcollector...
Aug 09 16:07:16 analyst-VirtualBox env[65398]: Started wazuh-monitord...
Aug 09 16:07:16 analyst-VirtualBox env[65708]: 2025/08/09 16:07:16 wazuh-modulesd:router: INFO: Loaded router module.
Aug 09 16:07:16 analyst-VirtualBox env[65708]: 2025/08/09 16:07:16 wazuh-modulesd:content_manager: INFO: Loaded content
Aug 09 16:07:16 analyst-VirtualBox env[65398]: Started wazuh-modulesd...

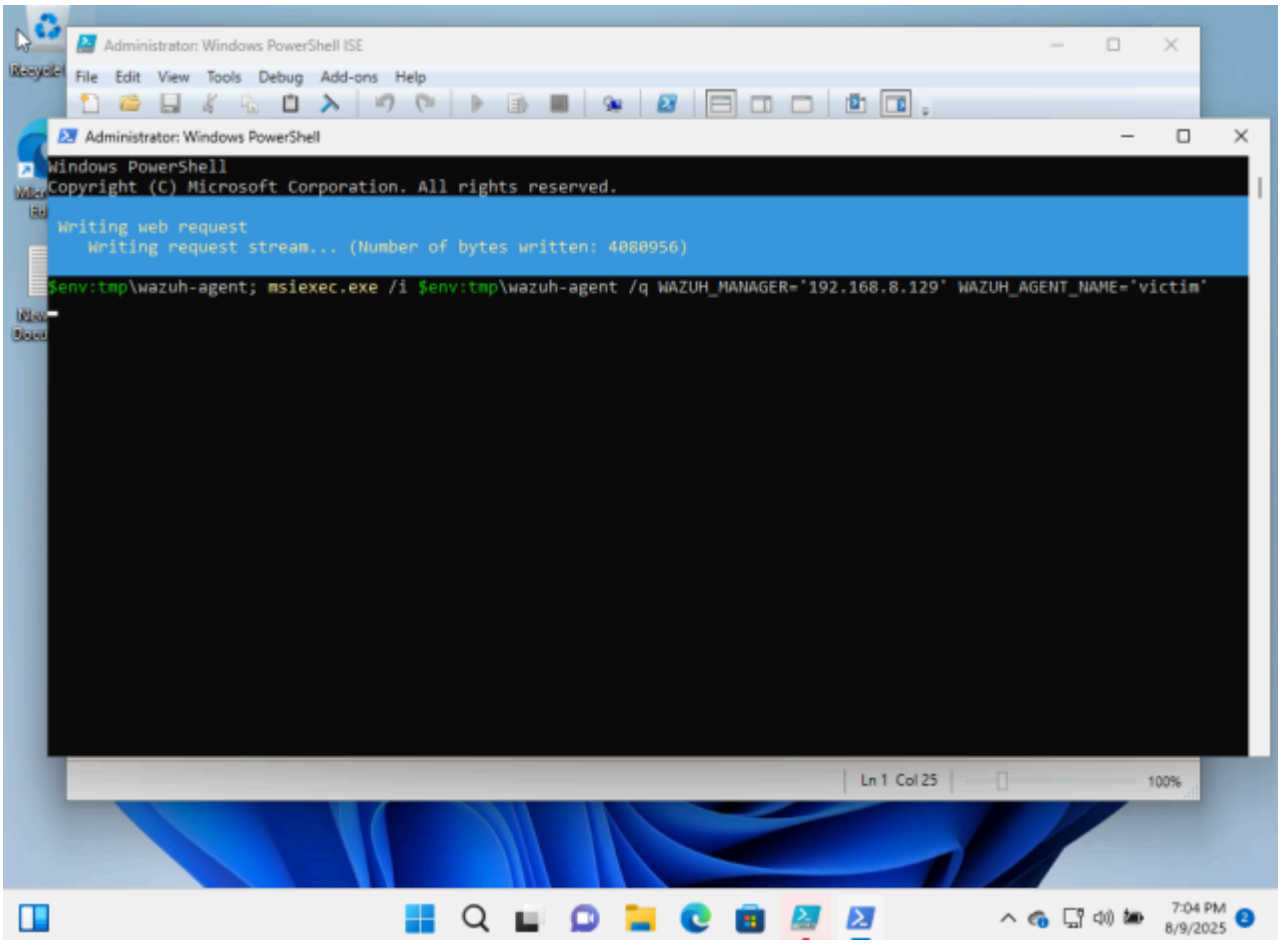

wazuh-dashboard.service - wazuh-dashboard
Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; preset: enabled)
Active: active (running) since Sat 2025-08-09 16:07:20 +0530; 17min ago
Main PID: 66522 (node)
Tasks: 11 (limit: 4603)
Memory: 187.8M (peak: 287.5M swap: 9.8M swap peak: 9.8M)
CPU: 28.800s
CGroup: /system.slice/wazuh-dashboard.service
66522 /usr/share/wazuh-dashboard/node/bin/node --no-warnings --max-http-header-size=65536 --unhandled-re
Aug 09 16:22:37 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:37Z"
Aug 09 16:22:37 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:37Z"
Aug 09 16:22:37 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:37Z"
Aug 09 16:22:37 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:37Z"
Aug 09 16:22:38 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:37Z"
Aug 09 16:22:38 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:37Z"
Aug 09 16:22:38 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:37Z"
Aug 09 16:22:38 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:38Z"
Aug 09 16:22:38 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:38Z"
Aug 09 16:22:45 analyst-VirtualBox opensearch-dashboards[66522]: {"type":"response","@timestamp":"2025-08-09T10:52:38Z"
lines 1-21/21 (END)

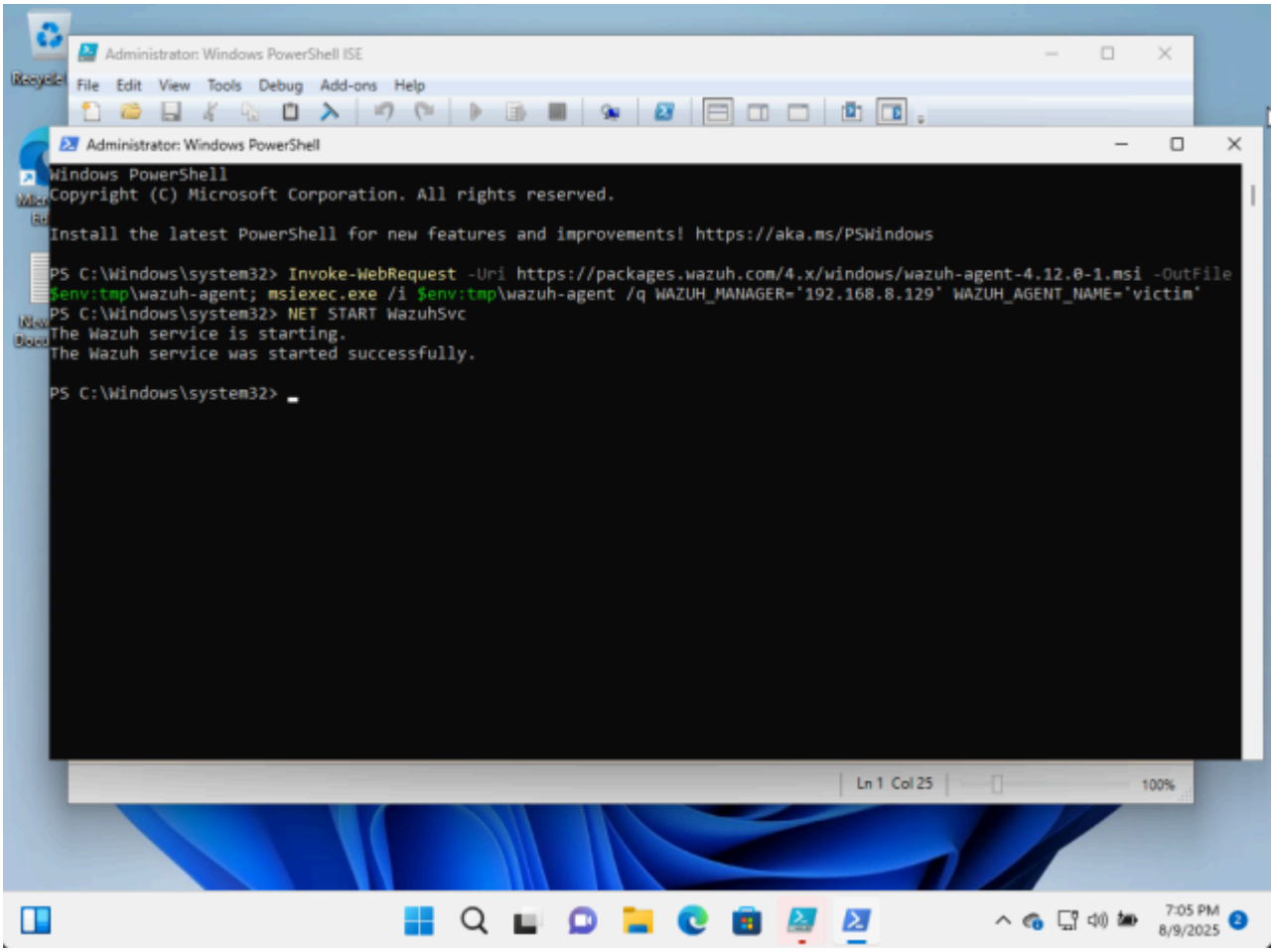


---

2. Configured Windows 11 as Endpoint

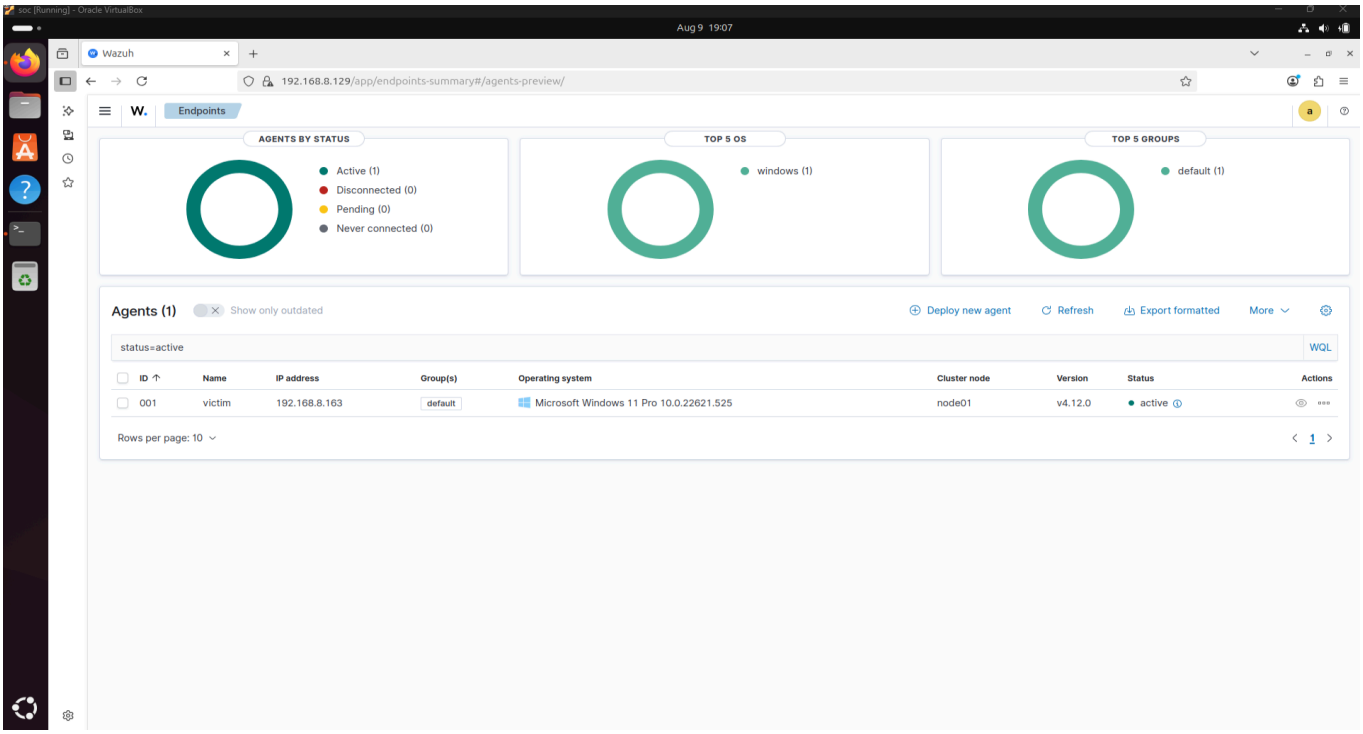
- Installed the **Wazuh agent** on Windows 11 VM.
- Connected the agent to the Ubuntu Wazuh server using the server IP & enrollment key.
- Confirmed agent registration.


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Writing web request
Writing request stream... (Number of bytes written: 4080956)
$env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.8.129' WAZUH_AGENT_NAME='victim'
Ln 1 Col 25
100%
7:04 PM
8/9/2025


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Windows\system32> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.12.0-1.msi -OutFile
$env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.8.129' WAZUH_AGENT_NAME='victim'
PS C:\Windows\system32> NET START WazuhSvc
The Wazuh service is starting.
The Wazuh service was started successfully.
PS C:\Windows\system32>
Ln 1 Col 25
100%
7:05 PM
8/9/2025

3. Validated Wazuh Agent Connectivity

- Logged into Wazuh dashboard.
- Checked the **Agents tab** → Windows 11 endpoint appeared as **Active**.
- Verified logs were being collected from the endpoint.



Results

- Wazuh successfully detected activities on the Windows 11 endpoint.
- Windows agent stayed active and continuously forwarded logs to Wazuh.
- Demonstrated the **end-to-end SOC process**:
  - Attack → Logging → Detection → Monitoring

🚀 Skills & Knowledge Gained

By completing this homelab, I strengthened skills directly applicable to a **SOC Analyst job role**:

- ✅ Hands-on experience with **SIEM (Wazuh)**
- ✅ Configuring **endpoint security agents** (Windows agent)
- ✅ Understanding **attack simulation** and detection workflow

- ✅ Building a realistic **SOC environment from scratch**

This project builds a strong foundation in **threat detection, log analysis, and SOC operations**, helping me bridge theory with practical SOC workflows.

📝 Conclusion

This project shows how a student can build a **mini SOC lab** using free tools and virtualization. It demonstrates **practical cybersecurity skills** that align with real SOC analyst job responsibilities.

Next steps → expand with **Linux endpoint agents, log forwarding, and advanced alert rules**.