PROJECT- REPORT

ON

WAZUH SOC DEPLOYMENT & ENDPOINT DETECTION

TITTLE

Wazuh SOC Deployment & Endpoint Detection. A Real-time monitoring, file integrity, and malware detection using Wazuh SIEM/EDR

PROJECT OVERVIEW

This project demonstrates the deployment of a Security Operations Center (SOC) using Wazuh, focusing on:

- Endpoint monitoring for Windows systems
- File Integrity Monitoring (FIM) for critical directories like Desktop
- Malware detection and automated removal using VirusTotal API
- Real-time log collection, alerting, and dashboard visualization

The goal is to showcase hands-on experience in SIEM, EDR, and SOC operations.

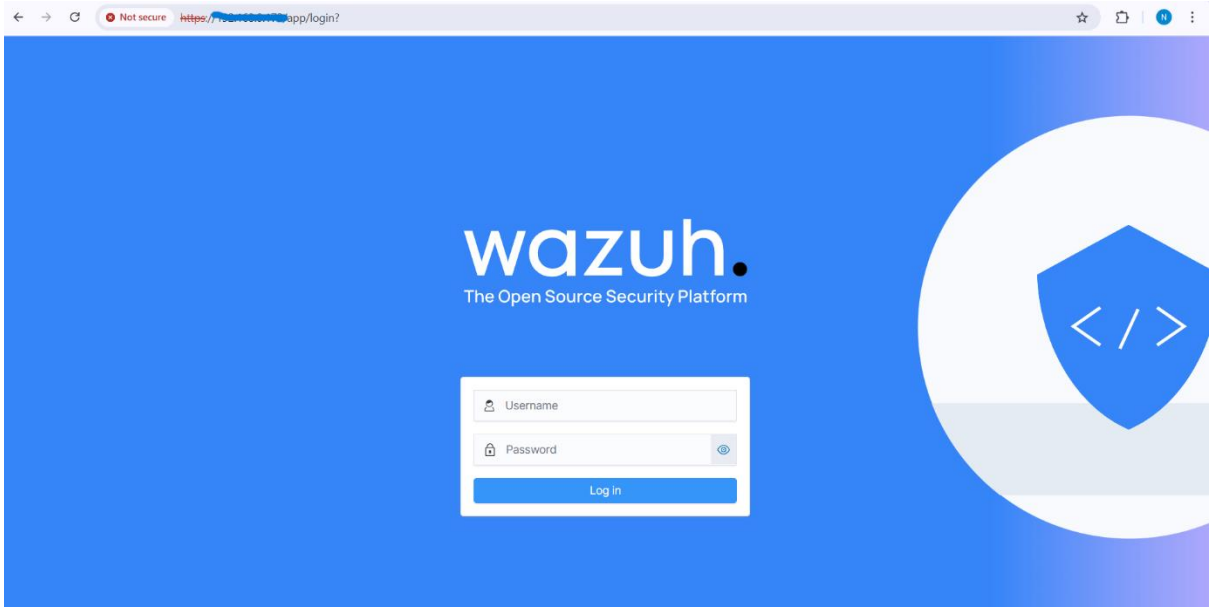
Deployment of Wazuh Manager & Agent

OBJECTIVE: Set up a centralized SOC environment to monitor endpoints, collect logs, and detect security events in real time.

Step 1: Installing Wazuh Dashboard on VirtualBox

- Downloaded Wazuh virtual appliance (OVA) from official documentation: https://documentation.wazuh.com
- Imported the OVA into VirtualBox:
    - Allocated sufficient RAM and CPU
    - Configured network adapter as Bridged for external access
- Started the virtual machine and logged in using:

    - Username: wazuh-user
    - Password: wazuh
- Accessed the dashboard via browser using Wazuh IP:
    - Example: https://192.168.1.111



Step 2: Installing Wazuh Agent on Windows

- Installed Wazuh Agent on Windows 10 virtual machines
- Dashboard IP and authentication key required for registration
- Authentication key extracted via SSH

SSH Connection Steps:

- Hostname: Wazuh dashboard IP
- Port: 22
- Username: wazuh-user
- Password: wazuh
- Commands:

```
sudo su
cd /var/ossec/bin
./manage_agents
```

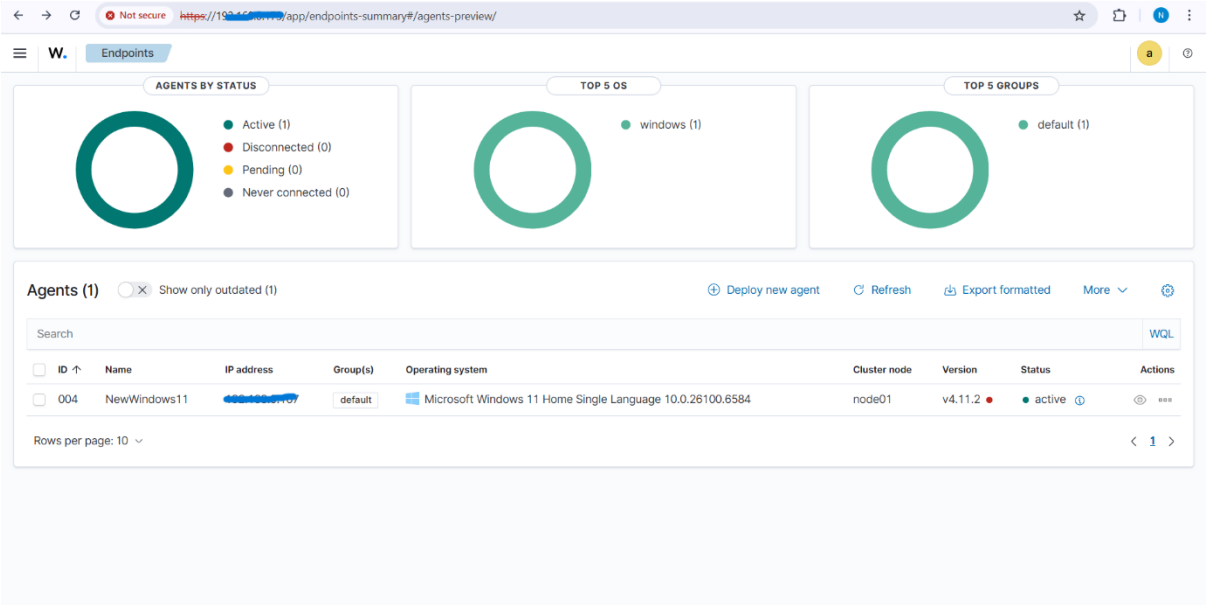
- Add agent → assign name (e.g., Windows-Wazuh) → enter agent IP → confirm → extract key
- Authentication key copied into Windows agent during setup

Step 3: Verifying Agent Connection

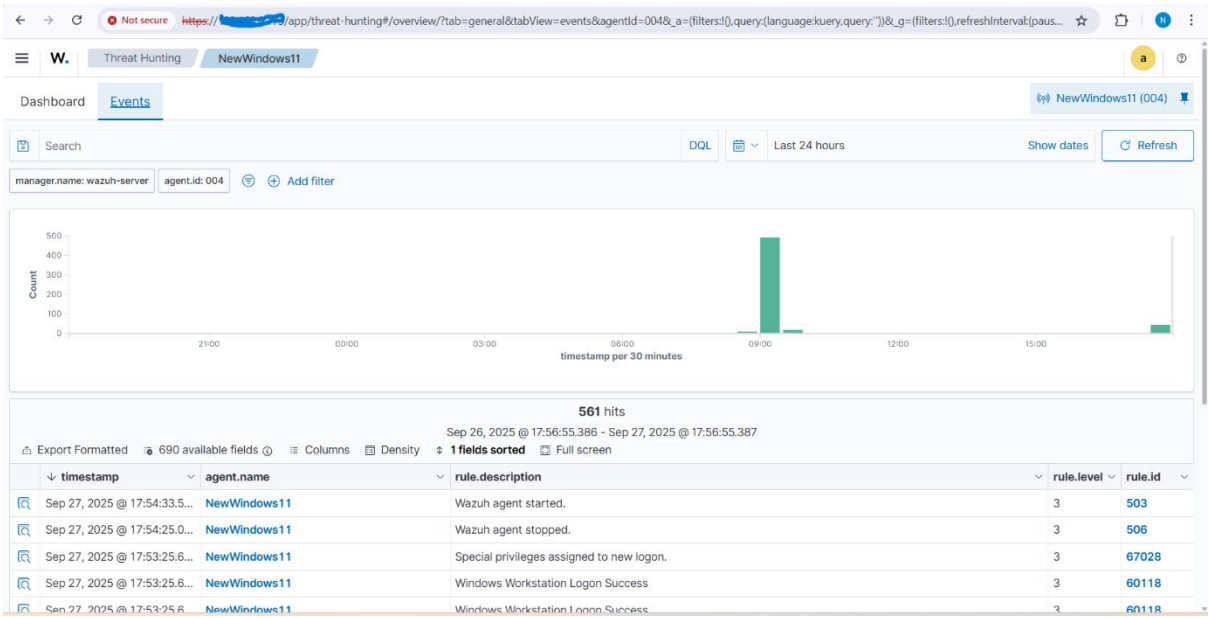
- Restarted Wazuh Agent service:

Restart-Service -Name WazuhSvc -Force

- Dashboard displayed the agent under Agents → Active, showing real-time log reporting



- Verified Windows events, including authentication attempts and file operations
- Generated test logs via simulated Hydra brute-force attack



2. File Integrity Monitoring (FIM)

Objective: Monitor critical directories on Windows endpoints for file creation, deletion, and modifications.

Step 1: Configuring FIM Rules

- Configured FIM in ossec.conf by enabling <syscheck> block
- Monitored directories: Desktop, Downloads
- Example configuration for Desktop:

<directories check-all="yes" report_changes="yes" realtime="yes">

C:\Users\<USER_NAME>\Desktop

</directories>

- Replace <USER_NAME> with Windows username
-

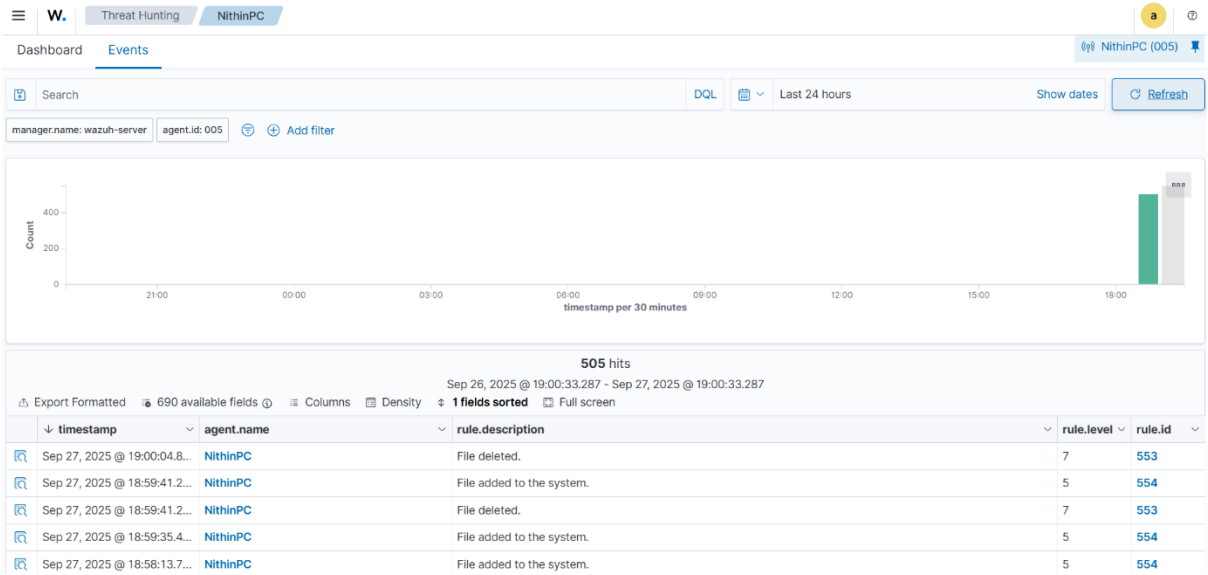
```
File    Edit    View
    <log_format>eventchannel</log_format>
  </localfile>

  <localfile>
    <location>active-response\active-responses.log</location>
    <log_format>syslog</log_format>
  </localfile>

  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>no</disabled>
    <windows_apps>./shared/win_applications_rcl.txt</windows_apps>
    <windows_malware>./shared/win_malware_rcl.txt</windows_malware>
  </rootcheck>

  <!-- Security Configuration Assessment -->
  <sca>
    <enabled>yes</enabled>
    <scan_on_start>yes</scan_on_start>
    <interval>12h</interval>
    <skip_nfs>yes</skip_nfs>
  </sca>

  <!-- File integrity monitoring -->
  <syscheck>

    <directories check_all="yes" report_changes="yes" realtime="yes">C:\Users\[redacted]\[redacted]\Documents\desktop</directories>

    <!-- Frequency that syscheck is executed default every 12 hours -->
    <frequency>43200</frequency>

    <!-- Default files to be monitored. -->
    <directories recursion_level="0" restrict="regedit.exe$|system.ini$|win.ini$">%WINDIR%</directories>

    <directories recursion_level="0" restrict="at.exe$|attrib.exe$|cacls.exe$|cmd.exe$|eventcreate.exe$|ftp.exe$|lsass.exe$|net.exe$|net1.exe$|netsh.exe$|reg.exe
$|regedt32.exe|regsvr32.exe|runas.exe|sc.exe|schtasks.exe|sethc.exe|subst.exe$">%WINDIR%\SysNative</directories>
    <directories recursion_level="0">%WINDIR%\SysNative\drivers\etc</directories>
    <directories recursion_level="0" restrict="WMIC.exe$">%WINDIR%\SysNative\wbem</directories>
    <directories recursion_level="0" restrict="powershell.exe$">%WINDIR%\SysNative\WindowsPowerShell\v1.0</directories>
    <directories recursion_level="0" restrict="winrm.vbs$">%WINDIR%\SysNative</directories>
```

Step 2: Testing FIM Alerts

- Created, edited, and deleted files on Desktop
- Refreshed Wazuh dashboard → logs showed events for file creation, modification, and deletion
- If logs did not appear, restarted agent service:

net stop Wazuh

net start Wazuh



3. Malware Removal Using VirusTotal API

Objective: Detect and automatically remove malware on Windows endpoints using near real-time monitoring of the Downloads directory.

Step 1: Configure Wazuh Agent on Windows

1. Open ossec.conf in:

C:\Program Files (x86)\ossec-agent\ossec.conf

2. Ensure <disabled>no</disabled> under <syscheck> to enable FIM
3. Add Downloads directory for near real-time monitoring:

<directories realtime="yes">C:\Users\<USER_NAME>\Downloads</directories>

4. Install Python 3.X and add it to PATH
5. Open PowerShell as Administrator and install PyInstaller:

pip install pyinstaller

pyinstaller –version

```
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> pip install pyinstaller
Collecting pyinstaller
  Downloading pyinstaller-6.16.0-py3-none-win_amd64.whl.metadata (8.5 kB)
Collecting altgraph (from pyinstaller)
  Downloading altgraph-0.17.4-py2.py3-none-any.whl.metadata (7.3 kB)
Collecting packaging>=22.0 (from pyinstaller)
  Downloading packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Collecting pefile!=2024.8.26,>=2022.5.30 (from pyinstaller)
  Downloading pefile-2023.2.7-py3-none-any.whl.metadata (1.4 kB)
Collecting pyinstaller-hooks-contrib>=2025.8 (from pyinstaller)
  Downloading pyinstaller_hooks_contrib-2025.9-py3-none-any.whl.metadata (16 kB)
Collecting pywin32-ctypes>=0.2.1 (from pyinstaller)
  Downloading pywin32_ctypes-0.2.3-py3-none-any.whl.metadata (3.9 kB)
Collecting setuptools>=42.0.0 (from pyinstaller)
  Downloading setuptools-80.9.0-py3-none-any.whl.metadata (6.6 kB)
Downloading pyinstaller-6.16.0-py3-none-win_amd64.whl (1.4 MB)
   ---------------------------------------- 1.4/1.4 MB 3.4 MB/s  0:00:00
Downloading packaging-25.0-py3-none-any.whl (66 kB)
Downloading pefile-2023.2.7-py3-none-any.whl (71 kB)
Downloading pyinstaller_hooks_contrib-2025.9-py3-none-any.whl (444 kB)
Downloading pywin32_ctypes-0.2.3-py3-none-any.whl (30 kB)
Downloading setuptools-80.9.0-py3-none-any.whl (1.2 MB)
   ---------------------------------------- 1.2/1.2 MB 950.7 kB/s  0:00:01
Downloading altgraph-0.17.4-py2.py3-none-any.whl (21 kB)
Installing collected packages: altgraph, setuptools, pywin32-ctypes, pefile, packaging, pyinstaller-hooks-contrib, pyinstaller
Successfully installed altgraph-0.17.4 packaging-25.0 pefile-2023.2.7 pyinstaller-6.16.0 pyinstaller-hooks-contrib-2025.9 pywin32-ctypes-0.2.3 setuptools-80.9.0
```

Step 2: Create Active Response Script

- Create remove-threat.py in agent active-response folder
- Script deletes malicious files securely
- Convert Python script to executable:

pyinstaller -F \path_to_remove-threat.py

- Move remove-threat.exe to:

C:\Program Files (x86)\ossec-agent\active-response\bin

- Restart agent:

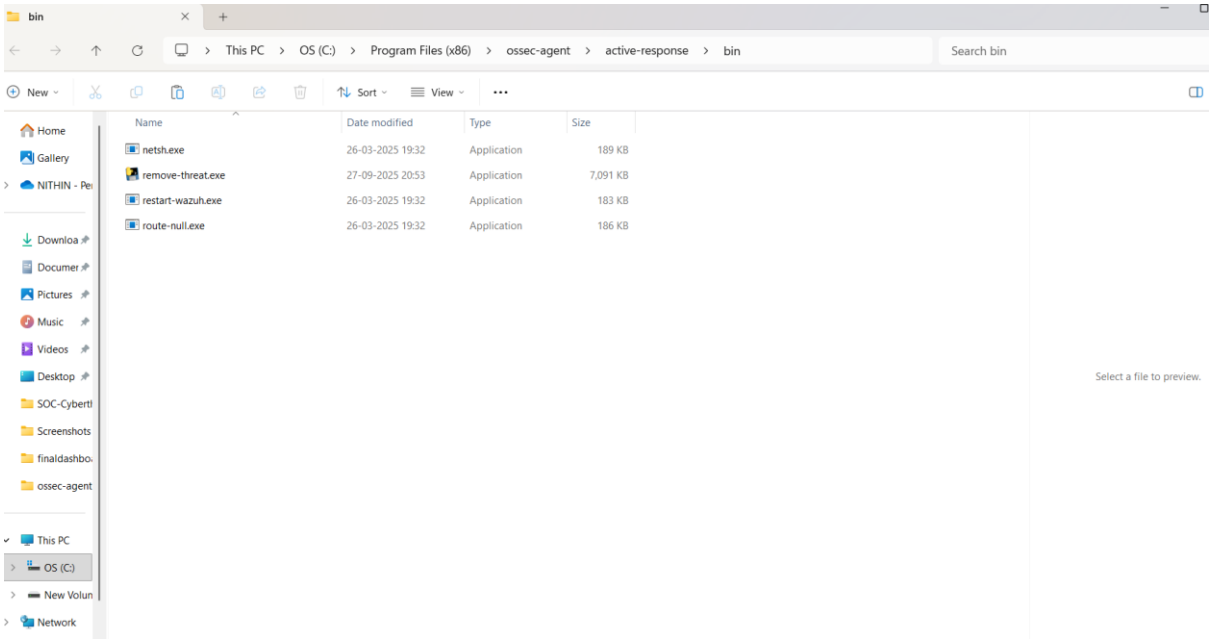
Restart-Service -Name wazuh

remove-threat.py

File Edit View

```
import os
import sys
import json
import datetime
import stat
import tempfile
import pathlib

if os.name == 'nt':
    LOG_FILE = "C:\\Program Files (x86)\\ossec-agent\\active-response\\active-responses.log"
else:
    LOG_FILE = "/var/ossec/logs/active-responses.log"

ADD_COMMAND = 0
DELETE_COMMAND = 1
CONTINUE_COMMAND = 2
ABORT_COMMAND = 3

OS_SUCCESS = 0
OS_INVALID = -1

class message:
    def __init__(self):
        self.alert = ""
        self.command = 0

def write_debug_file(ar_name, msg):
    with open(LOG_FILE, mode="a") as log_file:
        log_file.write(str(datetime.datetime.now().strftime('%Y/%m/%d %H:%M:%S')) + " " + ar_name + ": " + msg +"\n")

def setup_and_check_message(argv):
    input_str = ""
    for line in sys.stdin:
        input_str = line
        break

    msg_obj = message()
    try:
        data = json.loads(input_str)
    except ValueError:
        write_debug_file(argv[0], 'Decoding JSON has failed, invalid input format')
```

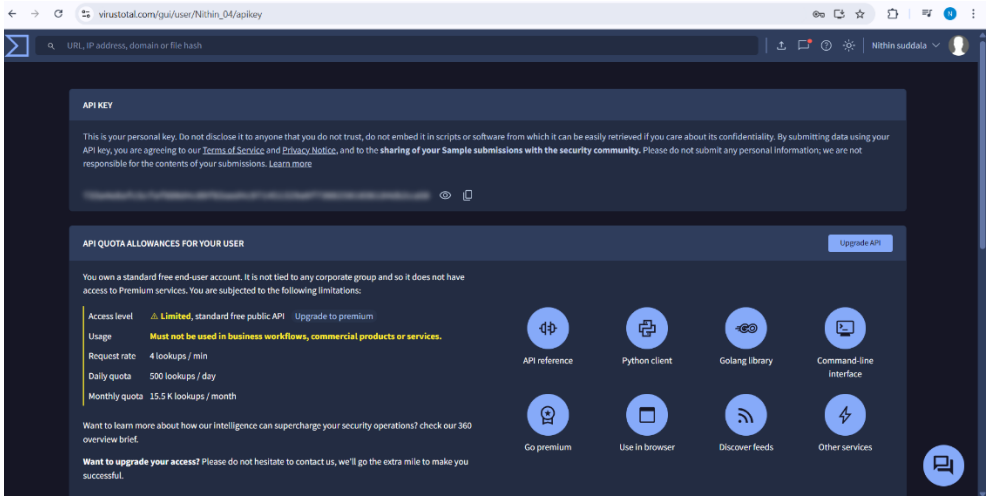
```
PS C:\Users\Nithin\Downloads>  pyinstaller -F .\remove-threat.py
240 INFO: PyInstaller: 6.16.0, contrib hooks: 2025.9
241 INFO: Python: 3.13.7
274 INFO: Platform: Windows-11-10.0.26100-SP0
274 INFO: Python environment: C:\Users\Nithin\AppData\Local\Programs\Python\Python313
275 INFO: wrote C:\Users\Nithin\Downloads\remove-threat.spec
281 INFO: Module search paths (PYTHONPATH):
['C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\pyinstaller.exe',
 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\python313.zip',
 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\DLLs',
 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\Lib',
 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313',
 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages',
 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\setuptools\\_vendor',
 'C:\\Users\\Nithin\\Downloads']
687 INFO: checking Analysis
687 INFO: Building Analysis because Analysis-00.toc is non existent
687 INFO: Looking for Python shared library...
687 INFO: Using Python shared library: C:\Users\Nithin\AppData\Local\Programs\Python\Python313\python313.dll
688 INFO: Running Analysis Analysis-00.toc
688 INFO: Target bytecode optimization level: 0
688 INFO: Initializing module dependency graph...
690 INFO: Initializing module graph hook caches...
709 INFO: Analyzing modules for base_library.zip ...
1947 INFO: Processing standard module hook 'hook-encodings.py' from 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages
nstaller\\hooks'
3160 INFO: Processing standard module hook 'hook-heapq.py' from 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\Py
ller\\hooks'
6330 INFO: Processing standard module hook 'hook-pickle.py' from 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\P
aller\\hooks'
9211 INFO: Caching module dependency graph...
9290 INFO: Analyzing C:\Users\Nithin\Downloads\remove-threat.py
9304 INFO: Processing module hooks (post-graph stage)...
9313 INFO: Performing binary vs. data reclassification (1 entries)
9317 INFO: Looking for ctypes DLLs
9330 INFO: Analyzing run-time hooks ...
9332 INFO: Including run-time hook 'pyi_rth_inspect.py' from 'C:\\Users\\Nithin\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\PyIns
r\\hooks\\rthooks'
9342 INFO: Creating base_library.zip...
9382 INFO: Looking for dynamic libraries
9519 INFO: Extra DLL search directories (AddDllDirectory): []
```

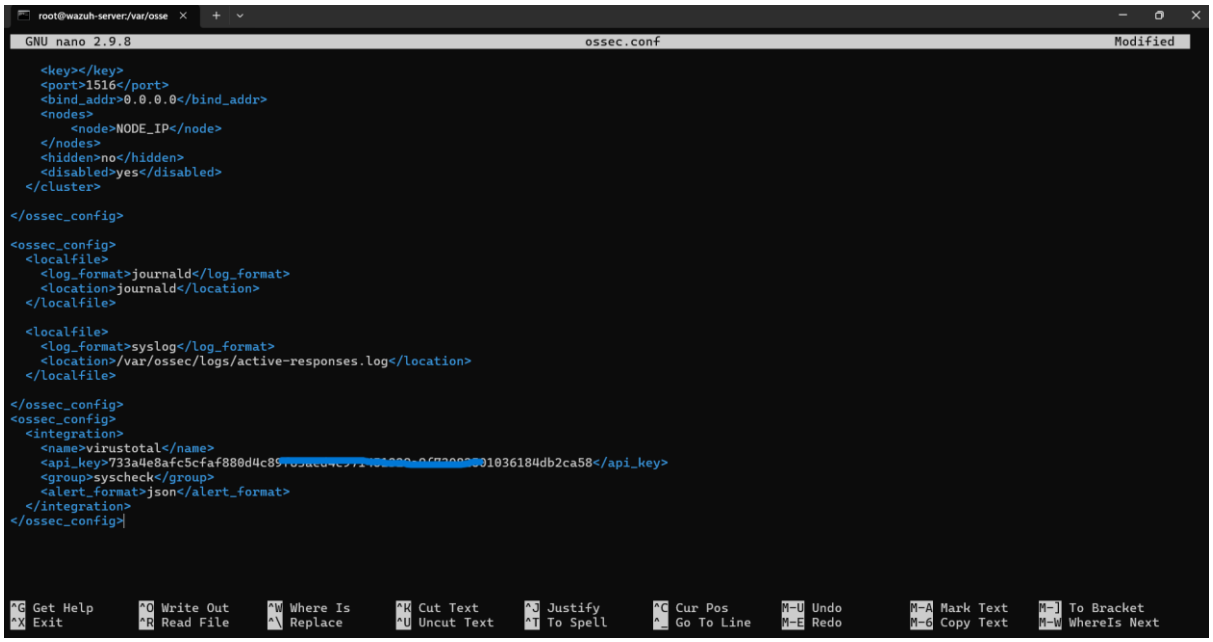


Step 3: Configure VirusTotal Integration on Wazuh Server

1. **Edit server /var/ossec/etc/ossec.conf:**

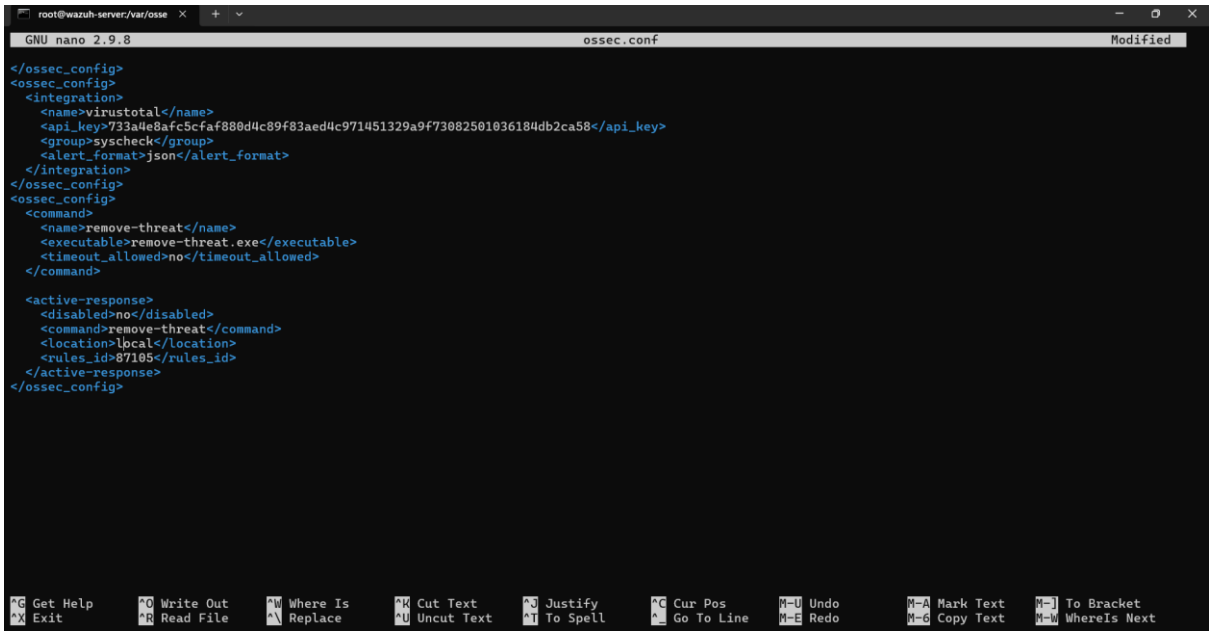
```
<ossec_config>
 <integration>
  <name>virustotal</name>
  <api_key><YOUR_VIRUS_TOTAL_API_KEY></api_key>
  <group>syscheck</group>
  <alert_format>json</alert_format>
 </integration>
</ossec_config>
```





2. **Configure active response to trigger remove-threat.exe:**

```
<ossec_config>
  <command>
    <name>remove-threat</name>
    <executable>remove-threat.exe</executable>
    <timeout_allowed>no</timeout_allowed>
  </command>

  <active-response>
    <disabled>no</disabled>
    <command>remove-threat</command>
    <location>local</location>
    <rules_id>87105</rules_id>
  </active-response>
</ossec_config>
```

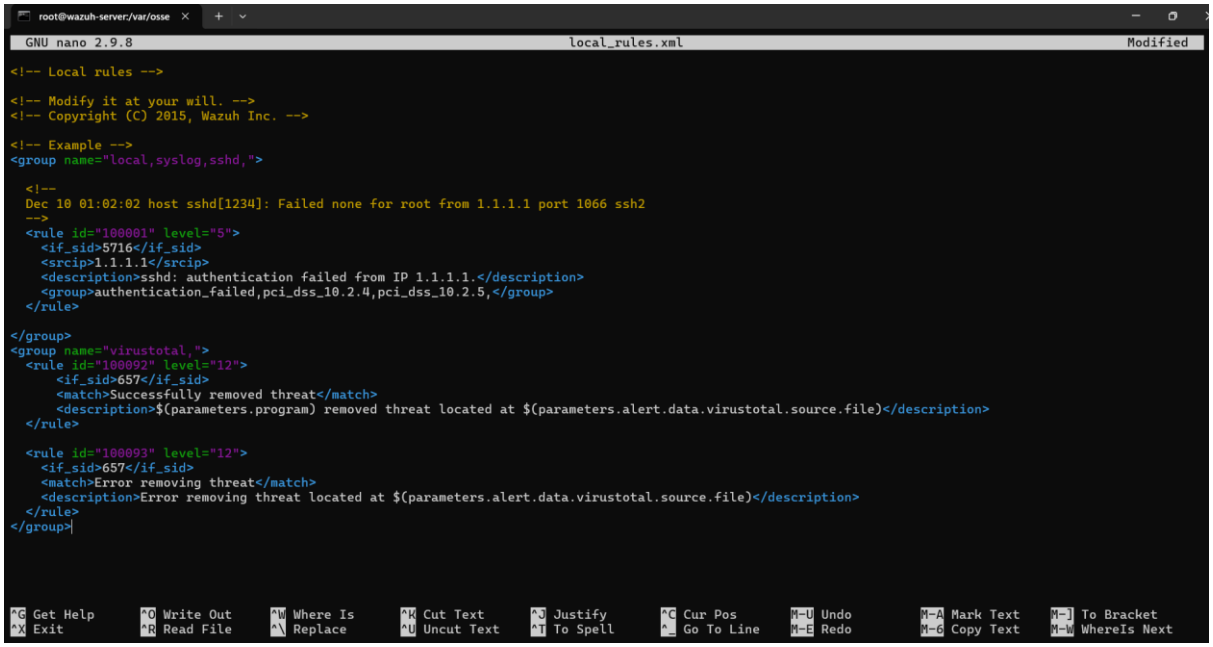


3. **Add rules in local_rules.xml to log results**:

```
<group name="virustotal">
 <rule id="100092" level="12">
   <if_sid>657</if_sid>
   <match>Successfully removed threat</match>
   <description>$(parameters.program) removed threat located at $(parameters.alert.data.virustotal.source.file)</description>
 </rule>

 <rule id="100093" level="12">
  <if_sid>657</if_sid>
  <match>Error removing threat</match>
  <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>
 </rule>
</group>
```

4. Restart Wazuh Manager:

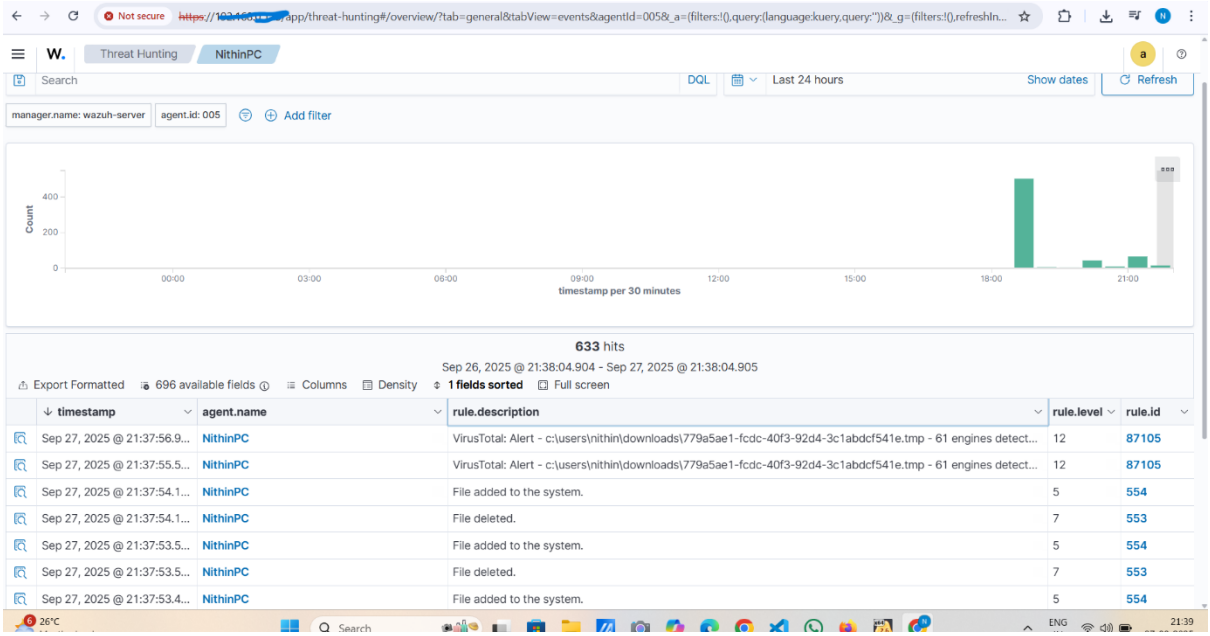
sudo systemctl restart wazuh-manager



Step 4: Attack Emulation and Testing

1. Disable real-time Microsoft Defender
2. Download EICAR test file to monitored folder:

Invoke-WebRequest -Uri https://secure.eicar.org/eicar.com.txt -OutFile C:\Users\<USER_NAME>\Downloads\eicar.txt

3. Wazuh triggers VirusTotal query and deletes file automatically
4. Alerts visualized in Wazuh Dashboard → Threat Hunting
5. 

Conclusion

- Successfully deployed Wazuh Manager and Windows Agents using VirtualBox
- Implemented File Integrity Monitoring for Desktop and Downloads
- Configured malware detection and automated removal using VirusTotal API
- Demonstrated real-time log collection, alerting, and SOC monitoring capabilities
- Project highlights hands-on experience in SIEM, EDR, and endpoint security operations