Real-Time Nmap Scan Detection with Suricata + Wazuh Integration | SOC Analyst Hands-On Project

Faizanullah Syed
SOC Analyst (In Training) | Cybersecurity Enthusiast □
Learning by doing. Passionate about threat detection, SIEM, and Blue Team operations.
Chhatrapati Sambhaji Nagar (Aurangabad), Maharashtra
faizanullah.syed19@gmail.com

□ Project Overview:

In this project, I demonstrate how to set up **Suricata**, an open-source network threat detection engine, and integrate it with **Wazuh**, a powerful SIEM platform. The goal is to detect **Nmap scans** and generate alerts in real-time for proactive network security monitoring.

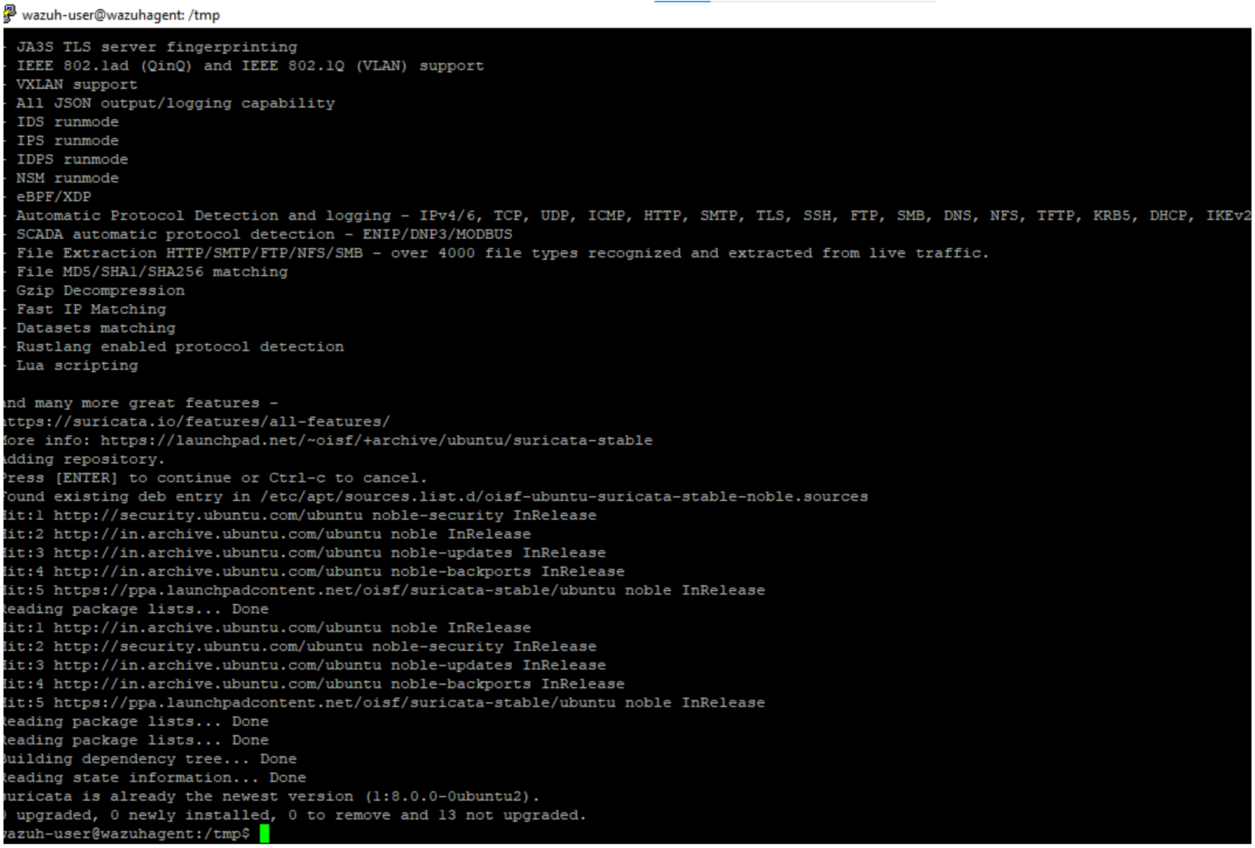
□ Key Objectives:

- Install and configure Suricata using the official PPA
- Download and enable Emerging Threats rules
- Tune Suricata to your network interface
- Integrate Suricata with Wazuh
- Detect active network scans (Nmap)
- Visualize alerts on the Wazuh Dashboard

□ Step-by-Step Setup:

□ *Step 1: Install Suricata via PPA*

```
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt-get update
sudo apt-get install suricata –y
```



wazuh-user@wazuhagent: /tmp

```
- JA3S TLS server fingerprinting
- IEEE 802.1ad (QinQ) and IEEE 802.1Q (VLAN) support
- VXLAN support
- All JSON output/logging capability
- IDS runmode
- IPS runmode
- IDPS runmode
- NSM runmode
- eBPF/XDP
- Automatic Protocol Detection and logging - IPv4/6, TCP, UDP, ICMP, HTTP, SMTP, TLS, SSH, FTP, SMB, DNS, NFS, TFTP, KRB5, DHCP, IKEv2
- SCADA automatic protocol detection - ENIP/DNP3/MODBUS
- File Extraction HTTP/SMTP/FTP/NFS/SMB - over 4000 file types recognized and extracted from live traffic.
- File MD5/SHA1/SHA256 matching
- Gzip Decompression
- Fast IP Matching
- Datasets matching
- Rustlang enabled protocol detection
- Lua scripting

And many more great features -
https://suricata.io/features/all-features/
More info: https://launchpad.net/~oisf/+archive/ubuntu/suricata-stable
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Found existing deb entry in /etc/apt/sources.list.d/oisf-ubuntu-suricata-stable-noble.sources
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu noble InRelease
Reading package lists... Done
Hit:1 http://in.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu noble InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
suricata is already the newest version (1:8.0.0-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 13 not upgraded.
wazuh-user@wazuhagent:/tmp$
```

□ Step 2: Download and Extract Emerging Threats Rules

cd /tmp

curl -LO https://rules.emergingthreats.net/open/suricata-6.0.8/emerging.rules.tar.gz

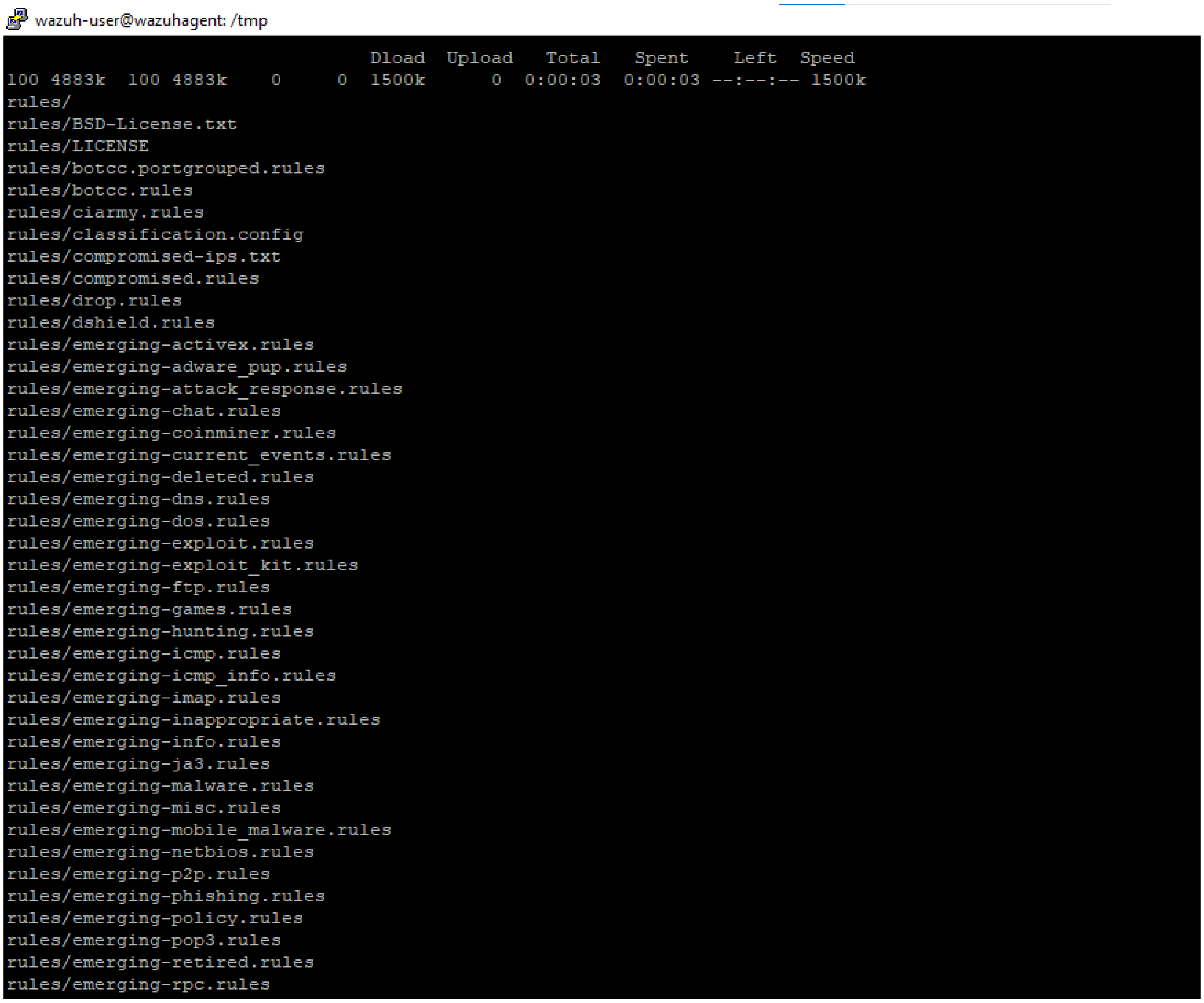
tar -xvzf emerging.rules.tar.gz

□ Step 3: Move Rules to Suricata Directory

sudo mkdir -p /etc/suricata/rules

sudo mv rules/*.rules /etc/suricata/rules/

sudo chmod 640 /etc/suricata/rules/*.rules



```
                                 Dload  Upload   Total   Spent    Left  Speed
100 4883k  100 4883k    0     0  1500k      0  0:00:03  0:00:03 --:--:-- 1500k
rules/
rules/BSD-License.txt
rules/LICENSE
rules/botcc.portgrouped.rules
rules/botcc.rules
rules/ciarmy.rules
rules/classification.config
rules/compromised-ips.txt
rules/compromised.rules
rules/drop.rules
rules/dshield.rules
rules/emerging-activex.rules
rules/emerging-adware_pup.rules
rules/emerging-attack_response.rules
rules/emerging-chat.rules
rules/emerging-coinminer.rules
rules/emerging-current_events.rules
rules/emerging-deleted.rules
rules/emerging-dns.rules
rules/emerging-dos.rules
rules/emerging-exploit.rules
rules/emerging-exploit_kit.rules
rules/emerging-ftp.rules
rules/emerging-games.rules
rules/emerging-hunting.rules
rules/emerging-icmp.rules
rules/emerging-icmp_info.rules
rules/emerging-imap.rules
rules/emerging-inappropriate.rules
rules/emerging-info.rules
rules/emerging-ja3.rules
rules/emerging-malware.rules
rules/emerging-misc.rules
rules/emerging-mobile_malware.rules
rules/emerging-netbios.rules
rules/emerging-p2p.rules
rules/emerging-phishing.rules
rules/emerging-policy.rules
rules/emerging-pop3.rules
rules/emerging-retired.rules
rules/emerging-rpc.rules
```

⍰ Step 4: Configure Suricata for Local Interface

sudo nano /etc/suricata/suricata.yaml

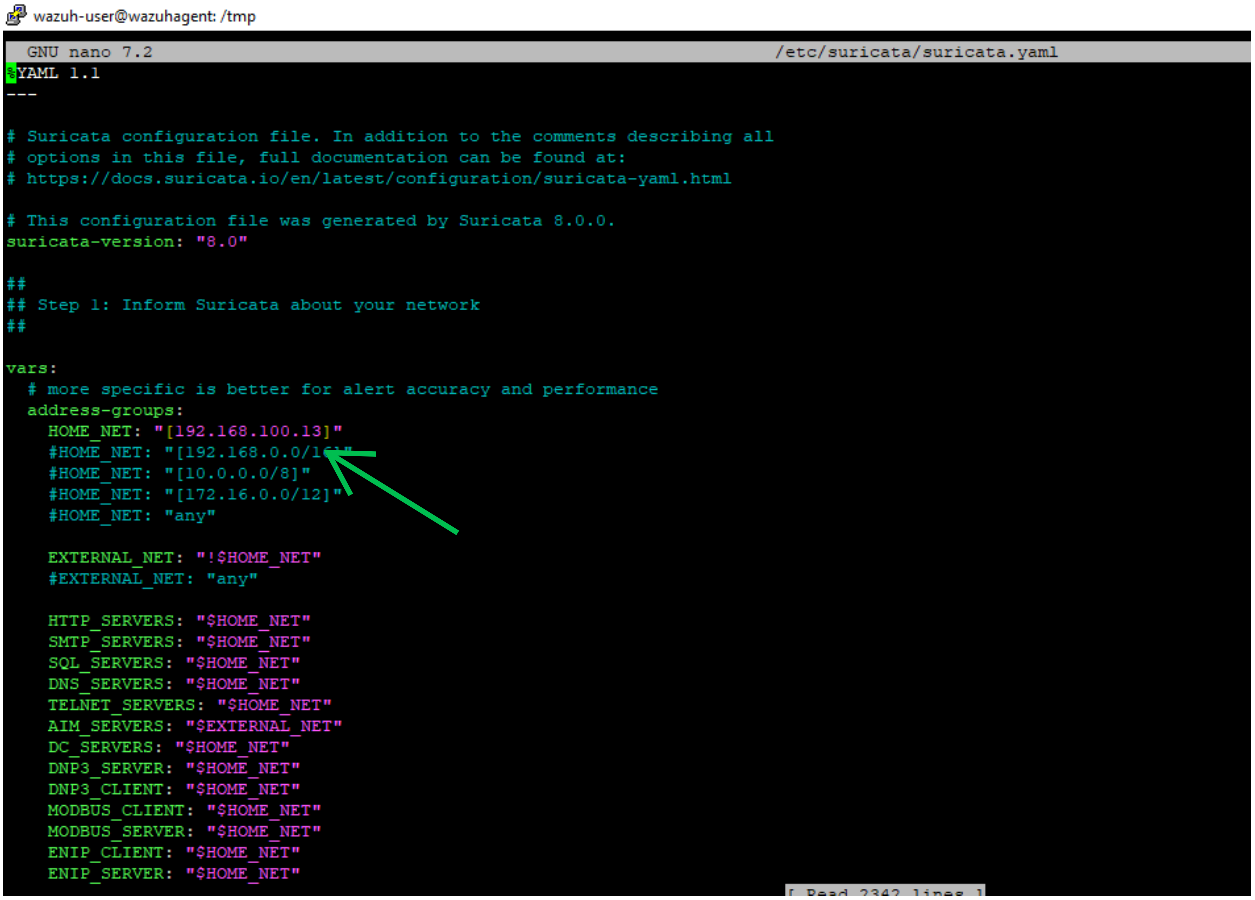
Inside `suricata.yaml`, update:

HOME_NET: "[192.168.100.13]"

EXTERNAL_NET: "!$HOME_NET"

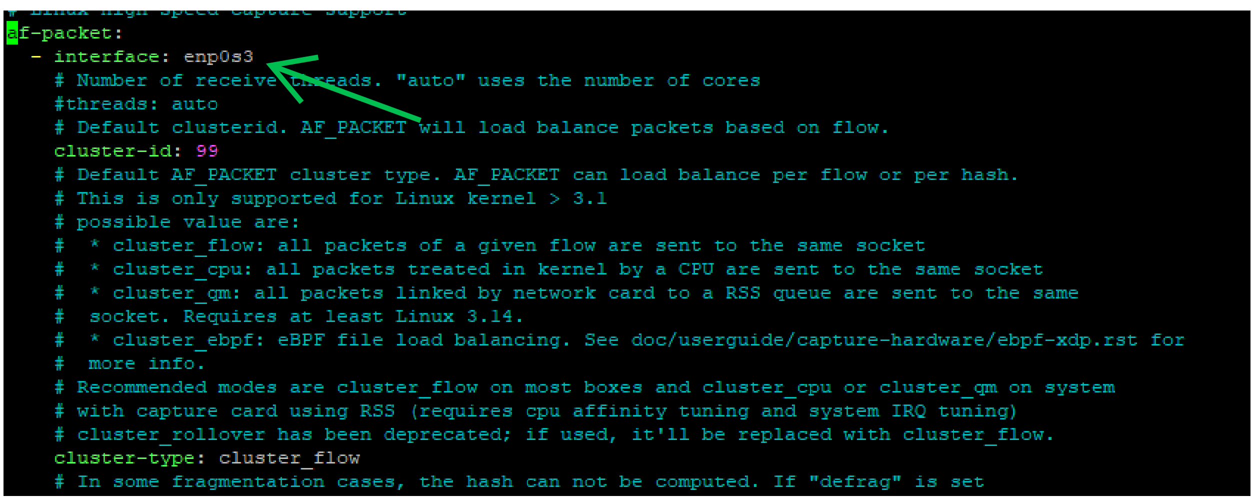
af-packet:

- interface: enp0s3



wazuh-user@wazuhagent: /tmp

```
  GNU nano 7.2                                                          /etc/suricata/suricata.yaml
%YAML 1.1
---

# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://docs.suricata.io/en/latest/configuration/suricata-yaml.html

# This configuration file was generated by Suricata 8.0.0.
suricata-version: "8.0"

##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.100.13]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"
    SQL_SERVERS: "$HOME_NET"
    DNS_SERVERS: "$HOME_NET"
    TELNET_SERVERS: "$HOME_NET"
    AIM_SERVERS: "$EXTERNAL_NET"
    DC_SERVERS: "$HOME_NET"
    DNP3_SERVER: "$HOME_NET"
    DNP3_CLIENT: "$HOME_NET"
    MODBUS_CLIENT: "$HOME_NET"
    MODBUS_SERVER: "$HOME_NET"
    ENIP_CLIENT: "$HOME_NET"
    ENIP_SERVER: "$HOME_NET"
                                                                                  [ Read 2342 lines ]
```



```
# Linux high speed capture support
af-packet:
  - interface: enp0s3
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    #  * cluster_flow: all packets of a given flow are sent to the same socket
    #  * cluster_cpu: all packets treated in kernel by a CPU are sent to the same socket
    #  * cluster_qm: all packets linked by network card to a RSS queue are sent to the same
    #  socket. Requires at least Linux 3.14.
    #  * cluster_ebpf: eBPF file load balancing. See doc/userguide/capture-hardware/ebpf-xdp.rst for
    #  more info.
    # Recommended modes are cluster_flow on most boxes and cluster_cpu or cluster_qm on system
    # with capture card using RSS (requires cpu affinity tuning and system IRQ tuning)
    # cluster_rollover has been deprecated; if used, it'll be replaced with cluster_flow.
    cluster-type: cluster_flow
    # In some fragmentation cases, the hash can not be computed. If "defrag" is set
```

Replace `192.168.100.13` and `enp0s3` with your actual IP and NIC name.

sudo systemctl restart suricata

sudo systemctl enable suricata

Step 5: Integrate Suricata with Wazuh

Edit Wazuh Agent config:

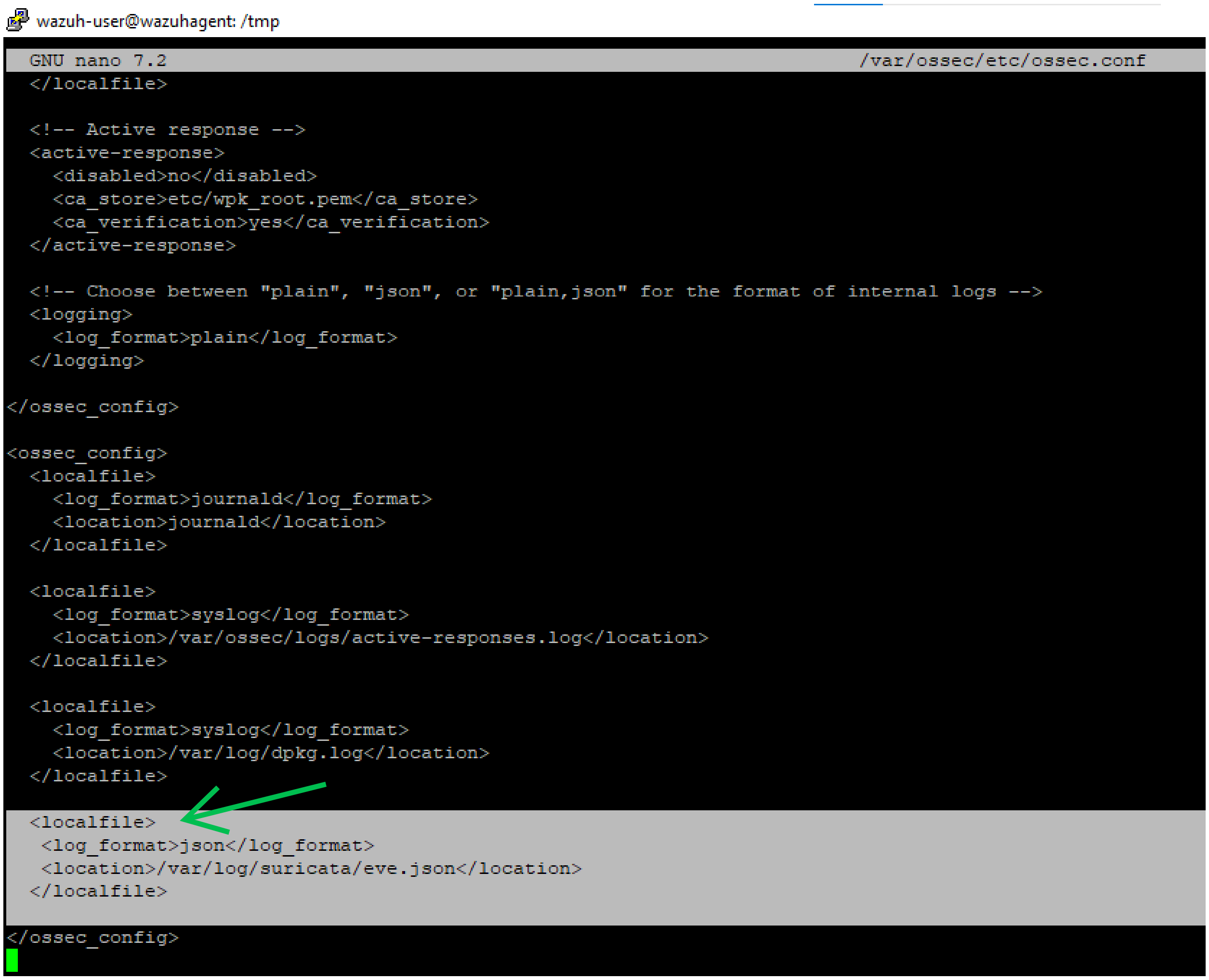
sudo nano /var/ossec/etc/ossec.conf

<localfile>

<log_format>json</log_format>

<location>/var/log/suricata/eve.json</location>

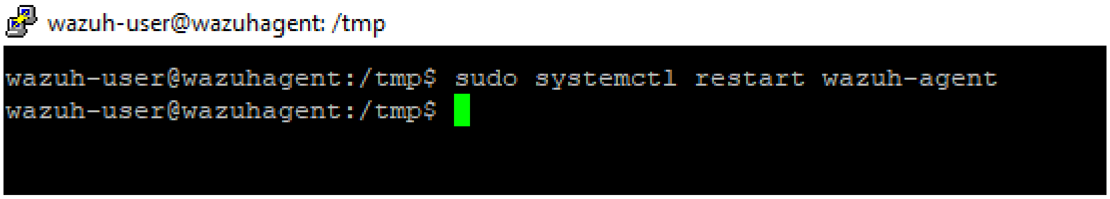
</localfile>



wazuh-user@wazuhagent: /tmp

```
  GNU nano 7.2                                                                    /var/ossec/etc/ossec.conf
  </localfile>

  <!-- Active response -->
  <active-response>
    <disabled>no</disabled>
    <ca_store>etc/wpk_root.pem</ca_store>
    <ca_verification>yes</ca_verification>
  </active-response>

  <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->
  <logging>
    <log_format>plain</log_format>
  </logging>

</ossec_config>

<ossec_config>
  <localfile>
    <log_format>journald</log_format>
    <location>journald</location>
  </localfile>

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

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/dpkg.log</location>
  </localfile>

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

</ossec_config>
```

Restart the agent:



wazuh-user@wazuhagent: /tmp

```
wazuh-user@wazuhagent:/tmp$ sudo systemctl restart wazuh-agent
wazuh-user@wazuhagent:/tmp$
```

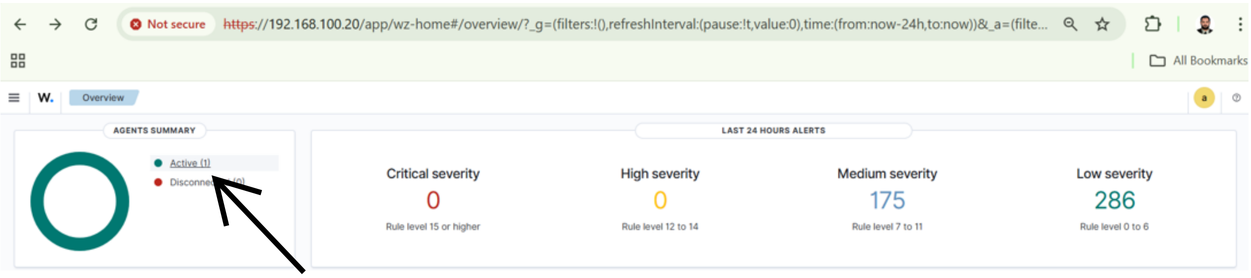
sudo systemctl restart wazuh-agent

□ Step 6: Generate a Simulated Attack with Nmap

nmap -sS -T4 192.168.100.13

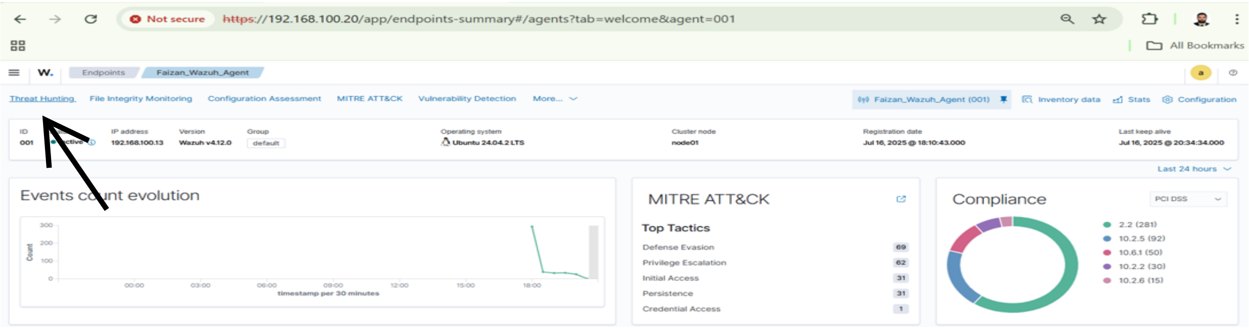
□ Step 7: Visualize Alerts in Wazuh

Go to:
Wazuh Dashboard > Modules > Security Events or NIDS
Look for alerts like:



Not secure https://192.168.100.20/app/wz-home#/overview/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-24h,to:now))&_a=(filte...

Critical severity	High severity	Medium severity	Low severity
0	0	175	286
Rule level 15 or higher	Rule level 12 to 14	Rule level 7 to 11	Rule level 0 to 6



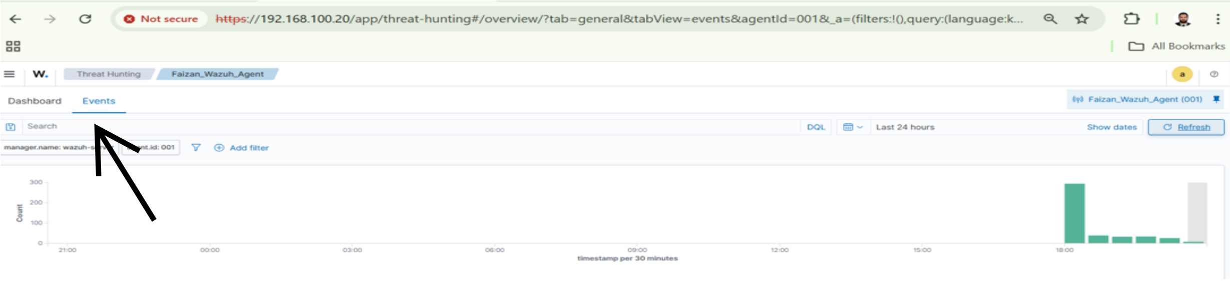
Not secure https://192.168.100.20/app/endpoints-summary#/agents?tab=welcome&agent=001

ID	Status	IP address	Version	Group	Operating system	Cluster node	Registration date	Last keep alive
001	active	192.168.100.13	Wazuh v4.12.0	default	Ubuntu 24.04.2 LTS	node01	Jul 16, 2025 @ 18:10:43.000	Jul 16, 2025 @ 20:34:34.000

Top Tactics

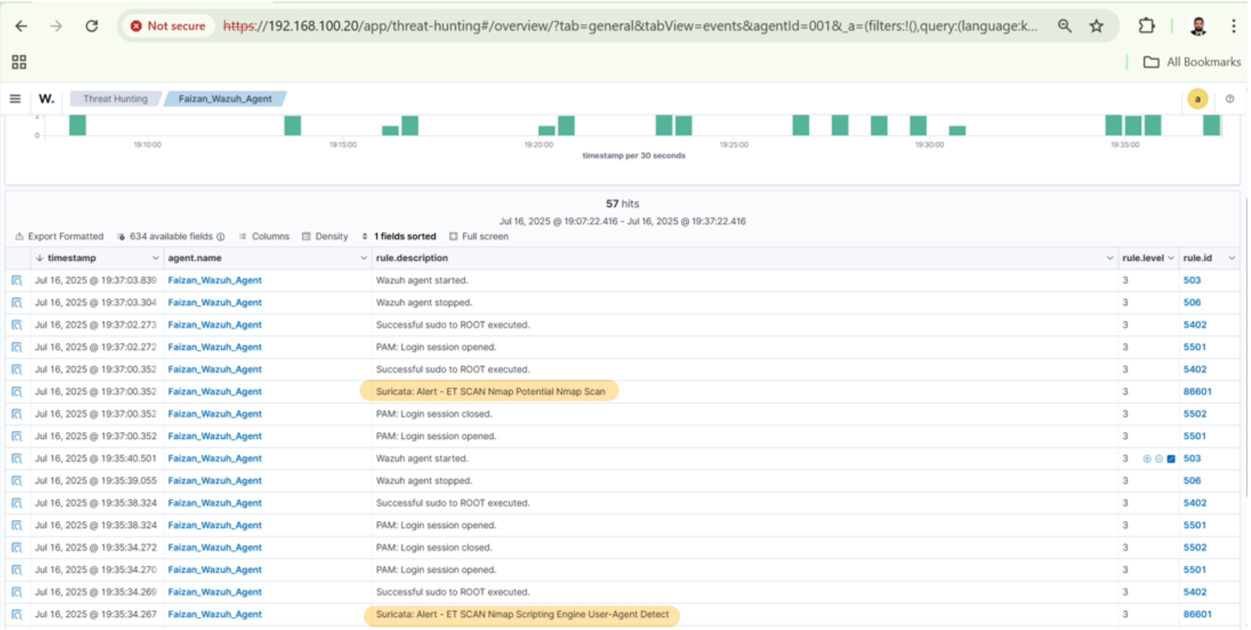
Defense Evasion	69
Privilege Escalation	62
Initial Access	31
Persistence	31
Credential Access	1

- 2.2 (281)
- 10.2.5 (92)
- 10.6.1 (50)
- 10.2.2 (30)
- 10.2.6 (15)



Not secure https://192.168.100.20/app/threat-hunting#/overview/?tab=general&tabView=events&agentId=001&_a=(filters:!(),query:(language:k...

manager.name: wazuh-server agent.id: 001



57 hits

Jul 16, 2025 @ 19:07:22.416 - Jul 16, 2025 @ 19:37:22.416

timestamp	agent.name	rule.description	rule.level	rule.id
Jul 16, 2025 @ 19:37:03.839	Faizan_Wazuh_Agent	Wazuh agent started.	3	503
Jul 16, 2025 @ 19:37:03.304	Faizan_Wazuh_Agent	Wazuh agent stopped.	3	506
Jul 16, 2025 @ 19:37:02.273	Faizan_Wazuh_Agent	Successful sudo to ROOT executed.	3	5402
Jul 16, 2025 @ 19:37:02.272	Faizan_Wazuh_Agent	PAM: Login session opened.	3	5501
Jul 16, 2025 @ 19:37:00.352	Faizan_Wazuh_Agent	Successful sudo to ROOT executed.	3	5402
Jul 16, 2025 @ 19:37:00.352	Faizan_Wazuh_Agent	Suricata: Alert - ET SCAN Nmap Potential Nmap Scan	3	86601
Jul 16, 2025 @ 19:37:00.352	Faizan_Wazuh_Agent	PAM: Login session closed.	3	5502
Jul 16, 2025 @ 19:37:00.352	Faizan_Wazuh_Agent	PAM: Login session opened.	3	5501
Jul 16, 2025 @ 19:35:40.501	Faizan_Wazuh_Agent	Wazuh agent started.	3	503
Jul 16, 2025 @ 19:35:39.055	Faizan_Wazuh_Agent	Wazuh agent stopped.	3	506
Jul 16, 2025 @ 19:35:38.324	Faizan_Wazuh_Agent	Successful sudo to ROOT executed.	3	5402
Jul 16, 2025 @ 19:35:38.324	Faizan_Wazuh_Agent	PAM: Login session opened.	3	5501
Jul 16, 2025 @ 19:35:34.272	Faizan_Wazuh_Agent	PAM: Login session closed.	3	5502
Jul 16, 2025 @ 19:35:34.270	Faizan_Wazuh_Agent	PAM: Login session opened.	3	5501
Jul 16, 2025 @ 19:35:34.269	Faizan_Wazuh_Agent	Successful sudo to ROOT executed.	3	5402
Jul 16, 2025 @ 19:35:34.267	Faizan_Wazuh_Agent	Suricata: Alert - ET SCAN Nmap Scripting Engine User-Agent Detect	3	86601

- **`ET SCAN Nmap Scripting Engine User-Agent Detected`**
- **`ET SCAN Potential SSH Scan`**

▯ Result:

✅Real-time detection of network scan attempts
✅Suricata rules successfully integrated
✅Wazuh capturing and displaying alerts
✅End-to-end SOC visibility for network intrusion attempts

▯ Key Skills Gained:

- IDS/IPS Tuning with Suricata
- SIEM integration with Wazuh
- Threat detection use case (Nmap)
- Rule management and tuning
- Linux security operations