Wazuh-Based Threat Detection and Log Monitoring System

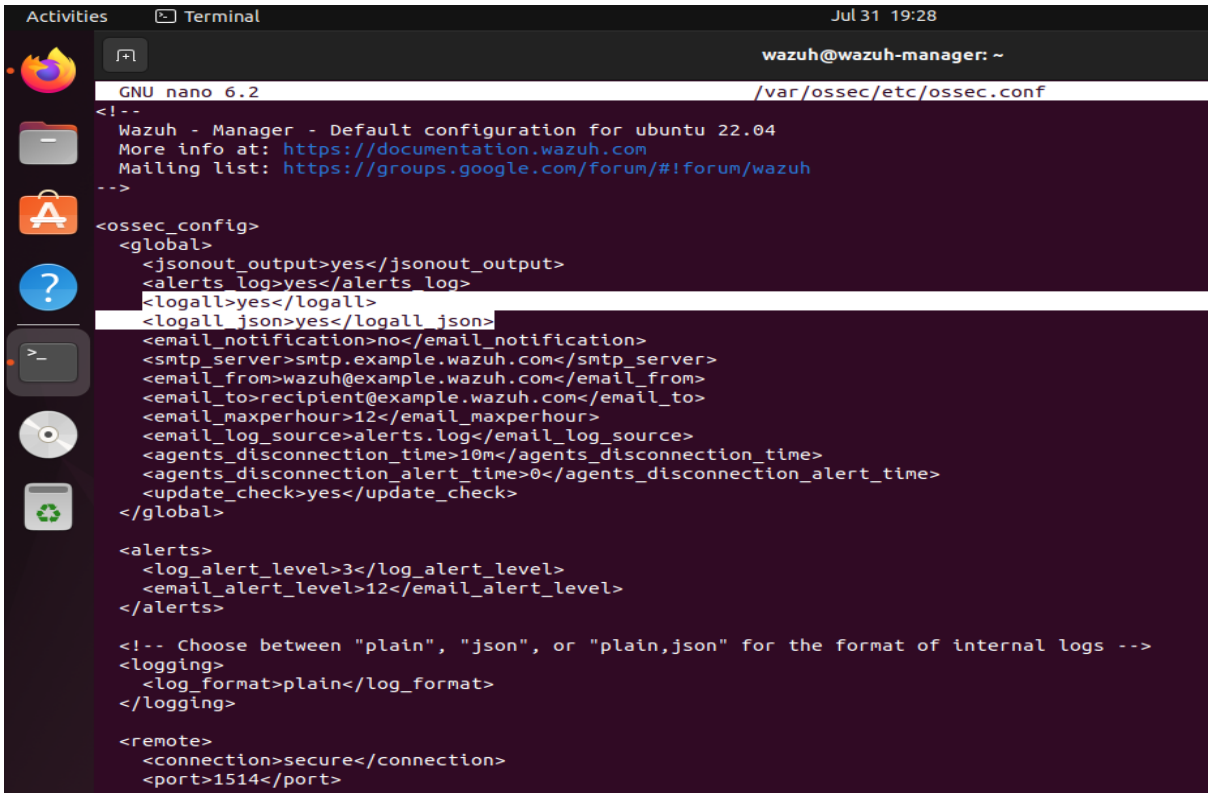
Objective

To set up a centralized security monitoring system using Wazuh Manager and Agent on Ubuntu 22.04 VMs. The project aims to collect system logs, detect security threats such as brute-force SSH attacks, monitor file integrity, and integrate Suricata for advanced intrusion detection.

Configure File Integrity Monitoring (FIM)

1. Open Wazuh Manager Configuration on the Wazuh Server:

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



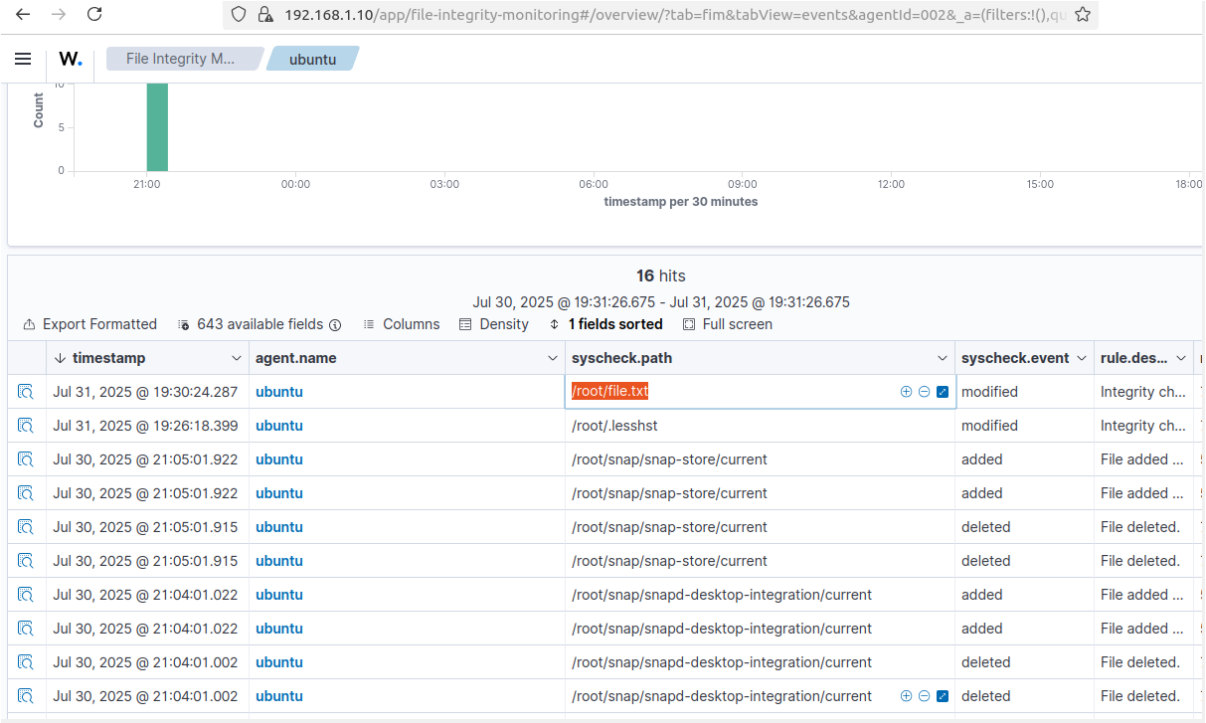
2. Then Open Wazuh Agent Configuration file.

```
wazuh@wazuh-agent: ~
GNU nano 6.2                                        /var/ossec/etc/ossec.conf
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>

<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>

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

  <scan_on_start>yes</scan_on_start>

  <!-- Directories to check  (perform all possible verifications) -->
  <directories>/etc,/usr/bin,/usr/sbin</directories>
  <directories>/bin,/sbin,/boot</directories>
  <directories check_all="yes" report_changes="yes" realtime="yes">/root</directories>
  <!-- Files/directories to ignore -->
  <ignore>/etc/mtab</ignore>
  <ignore>/etc/hosts.deny</ignore>
  <ignore>/etc/mail/statistics</ignore>
  <ignore>/etc/random-seed</ignore>
  <ignore>/etc/random.seed</ignore>
  <ignore>/etc/adjtime</ignore>
  <ignore>/etc/httpd/logs</ignore>
  <ignore>/etc/utmpx</ignore>
  <ignore>/etc/wtmpx</ignore>
  <ignore>/etc/cups/certs</ignore>
  <ignore>/etc/dumpdates</ignore>
  <ignore>/etc/svc/volatile</ignore>

  <!-- File types to ignore -->
  <ignore type="sregex">.log$|.swp$</ignore>

  <!-- Check the file, but never compute the diff -->
```

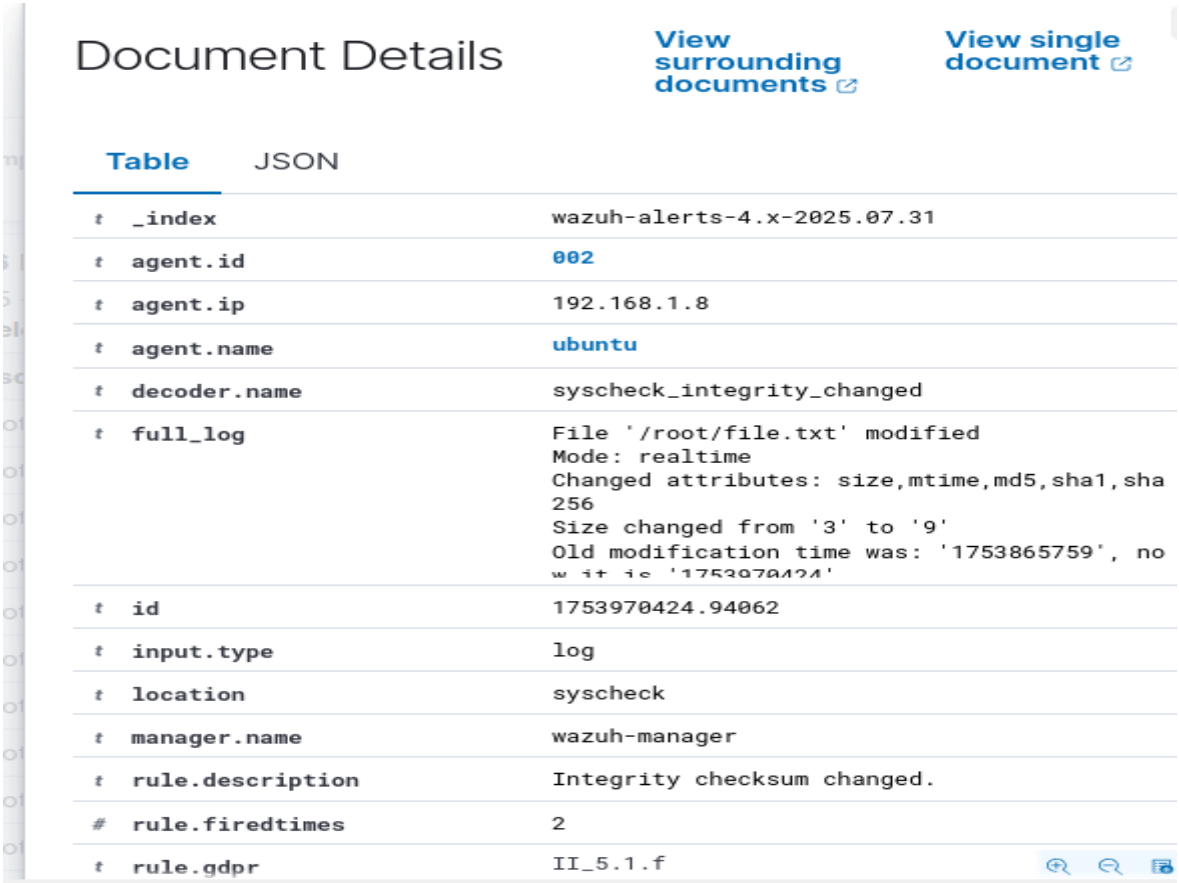
3. Restart Wazuh Agent to apply changes and Change or Modify Files in root Directory

```
wazuh@wazuh-agent:/$ sudo su
root@wazuh-agent:/# cd root
root@wazuh-agent:~# ls
file1.txt  file.txt  snap
root@wazuh-agent:~# nano file.txt
```

4. Restart the Wazuh Manager and Access to File Integrity Monitoring Tab



timestamp	agent.name	syscheck.path	syscheck.event	rule.des...
Jul 31, 2025 @ 19:30:24.287	ubuntu	/root/file.txt	modified	Integrity ch...
Jul 31, 2025 @ 19:26:18.399	ubuntu	/root/.lesshst	modified	Integrity ch...
Jul 30, 2025 @ 21:05:01.922	ubuntu	/root/snap/snap-store/current	added	File added ...
Jul 30, 2025 @ 21:05:01.922	ubuntu	/root/snap/snap-store/current	added	File added ...
Jul 30, 2025 @ 21:05:01.915	ubuntu	/root/snap/snap-store/current	deleted	File deleted.
Jul 30, 2025 @ 21:05:01.915	ubuntu	/root/snap/snap-store/current	deleted	File deleted.
Jul 30, 2025 @ 21:04:01.022	ubuntu	/root/snap/snapd-desktop-integration/current	added	File added ...
Jul 30, 2025 @ 21:04:01.022	ubuntu	/root/snap/snapd-desktop-integration/current	added	File added ...
Jul 30, 2025 @ 21:04:01.002	ubuntu	/root/snap/snapd-desktop-integration/current	deleted	File deleted.
Jul 30, 2025 @ 21:04:01.002	ubuntu	/root/snap/snapd-desktop-integration/current	deleted	File deleted.



t _index	wazuh-alerts-4.x-2025.07.31
t agent.id	002
t agent.ip	192.168.1.8
t agent.name	ubuntu
t decoder.name	syscheck_integrity_changed
t full_log	File '/root/file.txt' modified Mode: realtime Changed attributes: size,mtime,md5,sha1,sha256 Size changed from '3' to '9' Old modification time was: '1753865759', now it is '1753970424'
t id	1753970424.94062
t input.type	log
t location	syscheck
t manager.name	wazuh-manager
t rule.description	Integrity checksum changed.
# rule.firedtimes	2
t rule.gdpr	II_5.1.f

Detecting SSH Brute Force attack

1. From the attacker machine, install Hydra:

   sudo apt update && sudo apt install hydra -y

2. Run the following command to simulate an SSH brute-force attack:

   Hydra -l admin -P pass.txt 192.168.1.8 ssh

```
wazuh@wazuh-manager:~$ hydra -l admin -P pass.txt 192.168.1.8 ssh
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illeg
al purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-31 19:38:56
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per task
[DATA] attacking ssh://192.168.1.8:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-31 19:38:59
wazuh@wazuh-manager:~$
```

3. Open the Wazuh Dashboard and Navigate to Threat Hunting with rule.id: "5710"

268 hits

Jul 30, 2025 @ 19:39:41.769 - Jul 31, 2025 @ 19:39:41.769

Export Formatted | 643 available fields | Columns | Density | **1 fields sorted** | Full screen

timestamp	agent.name	rule.description	rule.level	rule.id
Jul 31, 2025 @ 19:39:00.918	ubuntu	sshd: Attempt to login using a non-existent user	5	5710
Jul 31, 2025 @ 19:39:00.912	ubuntu	sshd: Attempt to login using a non-existent user	5	5710
Jul 31, 2025 @ 19:39:00.909	ubuntu	sshd: Attempt to login using a non-existent user	5	5710
Jul 31, 2025 @ 19:39:00.907	ubuntu	sshd: Attempt to login using a non-existent user	5	5710
Jul 31, 2025 @ 19:39:00.904	ubuntu	sshd: Attempt to login using a non-existent user	5	5710
Jul 31, 2025 @ 19:39:00.903	ubuntu	sshd: Attempt to login using a non-existent user	5	5710
Jul 31, 2025 @ 19:39:00.902	ubuntu	sshd: Attempt to login using a non-existent user	5	5710
Jul 31, 2025 @ 19:38:58.955	ubuntu	PAM: User login failed.	5	5503
Jul 31, 2025 @ 19:38:58.950	ubuntu	PAM: User login failed.	5	5503
Jul 31, 2025 @ 19:38:58.941	ubuntu	PAM: User login failed.	5	5503

Document Details

View surrounding documents | View single document

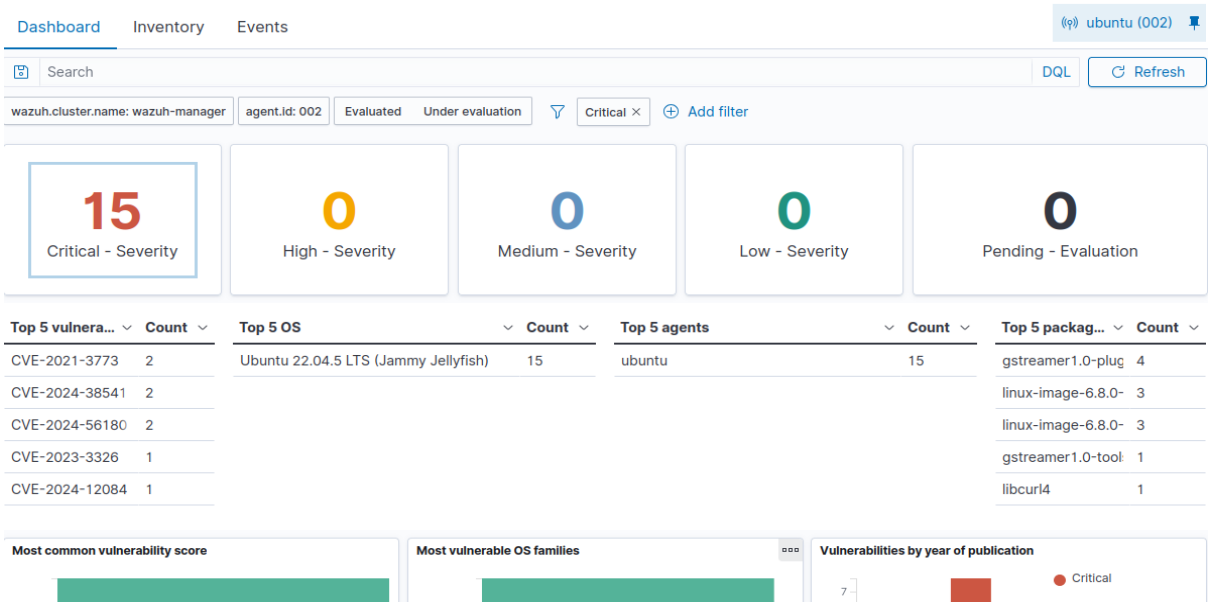
Table | JSON

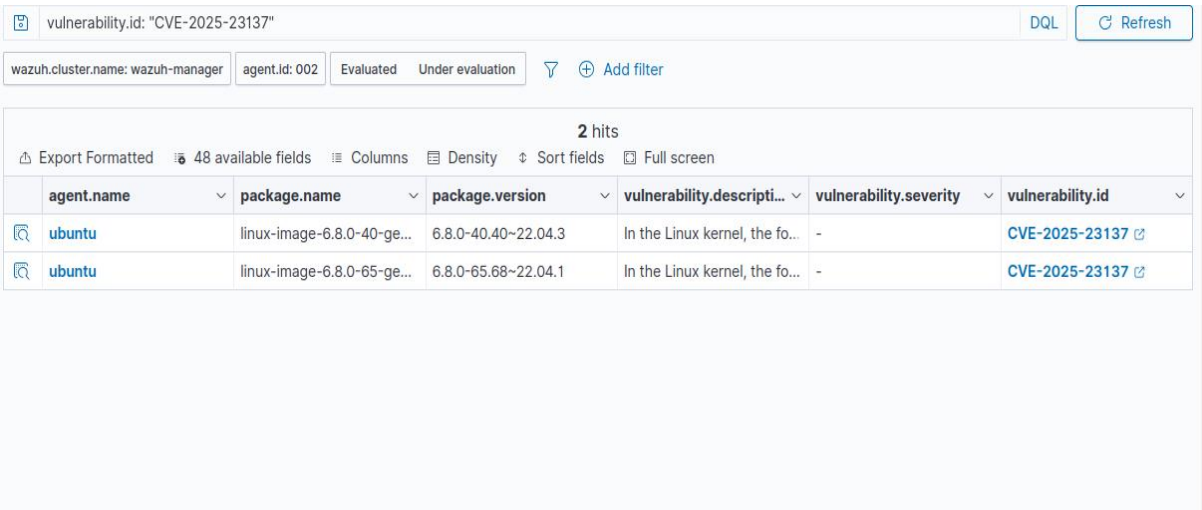
_index	wazuh-alerts-4.x-2025.07.31
agent.id	002
agent.ip	192.168.1.8
agent.name	ubuntu
data.srcip	192.168.1.10
data.srcuser	admin
decoder.name	sshd
decoder.parent	sshd
full_log	Jul 31 14:08:59 wazuh-agent sshd[3277]: Failed password for invalid user admin from 192.168.1.10 port 57356 ssh2
id	1753970940.108316
input.type	log
location	journald
manager.name	wazuh-manager
predecoder.hostname	wazuh-agent
predecoder.program_name	sshd

4. The attack came From 192.168.1.10 to get a password of admin(not existing)

Vulnerability Detection using Wazuh

1. Navigate to Vulnerability detection





2. CVE-2025-23137 : In the Linux kernel, the following vulnerability has been resolved: cpufreq/amd-pstate: Add missing NULL ptr check in amd_pstate_update Check if policy is NULL before dereferencing it in amd_pstate_update.

3. Wazuh will give the information about the vulnerability from vulnerability databases like NVD and Canonical Security Tracker

Detecting Suspicious Network Traffic using Suricata

1. Install Suricata on the Ubuntu endpoint. We tested this process with version 6.0.8 and it can take some time:

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

2. Download and extract the Emerging Threats Suricata ruleset:

   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 mkdir /etc/suricata/rules && sudo mv rules/.rules /etc/suricata/rules/*
   sudo chmod 640 /etc/suricata/rules/.rules*
3. Navigate to Suricata configuration File

   /etc/suricata/suricata.yaml
   Then Give Ip address of agent, Interface, rule-path and rule.file

```
root@wazuh-agent: /etc/suricata
GNU nano 6.2                    suricata.yaml
# 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: "[192.168.1.8]"
    #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"
```

```
root@wazuh-agent: /etc/suricata
GNU nano 6.2                    suricata.yaml
      # format: "[%i - %m] %z %d: %S: %M"
      # type: json
  - syslog:
      enabled: no
      facility: local5
      format: "[%i] <%d> -- "
      # type: json


##
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##

# 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.
```

```
root@wazuh-agent: /etc/suricata
GNU nano 6.2                                    suricata.yaml
    # When auto-config is enabled the hashmode specifies the algorithm for
    # determining to which stream a given packet is to be delivered.
    # This can be any valid Napatech NTPL hashmode command.
    #
    # The most common hashmode commands are:  hash2tuple, hash2tuplesorted,
    # hash5tuple, hash5tuplesorted and roundrobin.
    #
    # See Napatech NTPL documentation other hashmodes and details on their use.
    #
    # This parameter has no effect if auto-config is disabled.
    #
    hashmode: hash5tuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /etc/suricata/rules

rule-files:
  - "*.rules"

##
## Auxiliary configuration files.
##

classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config

##
## Suricata as a Firewall options (experimental)
##
```

4. Then Navigate to agent config file */var/ossec/etc/ossec.conf*

   Add the following configuration

```
root@wazuh-agent: /
GNU nano 6.2                                    /var/ossec/etc/ossec.conf
  <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>
```

5. Restart the Suricata and Initiate nmap scan from another machine

```
             ├─1968 /var/ossec/bin/wazuh-db
             ├─1996 /var/ossec/bin/wazuh-execd
             ├─2017 /var/ossec/bin/wazuh-analysisd
             ├─2030 /var/ossec/bin/wazuh-syscheckd
             ├─2050 /var/ossec/bin/wazuh-remoted
             ├─2091 /var/ossec/bin/wazuh-logcollector
             ├─2112 /var/ossec/bin/wazuh-monitord
             └─2128 /var/ossec/bin/wazuh-modulesd

lines 1-23
wazuh@wazuh-manager:~$ sudo su
root@wazuh-manager:/home/wazuh# nmap -sS 192.168.1.8
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-31 22:27 IST
Nmap scan report for 192.168.1.8
Host is up (0.00020s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:8A:10:AE (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
root@wazuh-manager:/home/wazuh#
```

Conclusion

- Successfully configured File Integrity Monitoring (FIM) in Wazuh.
- Simulated unauthorized file creation and modifications on a Ubuntu system
- Detected file changes in Wazuh logs and dashboard alerts.
- Successfully configured Wazuh Agent on an Ubuntu machine for SSH monitoring.
- Detected brute-force attempts in Wazuh logs and alerts.
- Understand how vulnerabilities are detected by Wazuh EDR.
- View and interpret CVE alerts on the dashboard.
- Install and configure Suricata to monitor traffic.
- Detect and respond to a port scanning attack.
- Visualize network alerts on Wazuh SIEM.