

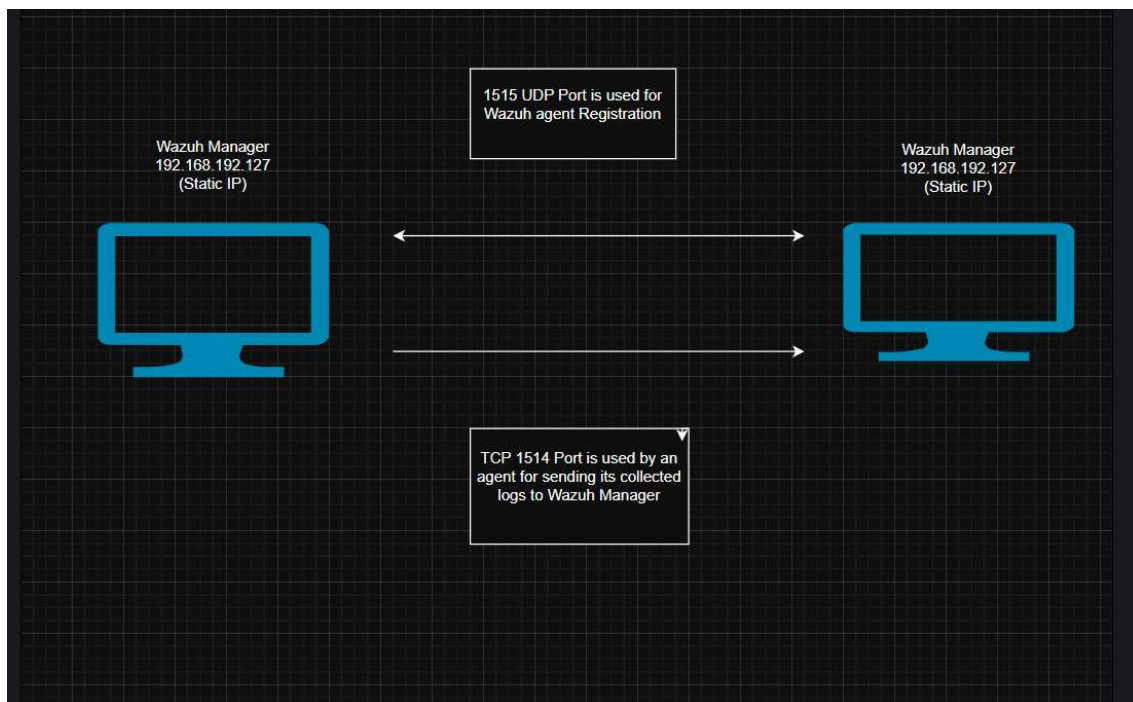
- **Description :** -

- This project demonstrates the setup of a Security Monitoring environment using **Wazuh** (Manager, Agent, and Dashboard) integrated with **Suricata IDS**.
- The goal is to detect and visualize network intrusion attempts and custom Suricata alerts within the Wazuh Dashboard.

- **Environment Setup :** -

The environment consists of :

- Wazuh manager installed on Ubuntu Server (e.g., IP: 192.168.192.127)
- Wazuh dashboard installed on the same system as the manager
- Wazuh agent installed on a monitored endpoint system
- Suricata IDS installed on the agent machine to monitor traffic and generate alerts



– Installing and configuring Wazuh manager, Agent and Dashboard : -

- The Wazuh Manager is responsible for collecting, analyzing, and storing security events from all agents.
- After installation, the Wazuh Dashboard was accessed via the web interface at: <https://192.168.192.127/>
- The dashboard provides an overview of agent status and alerts.

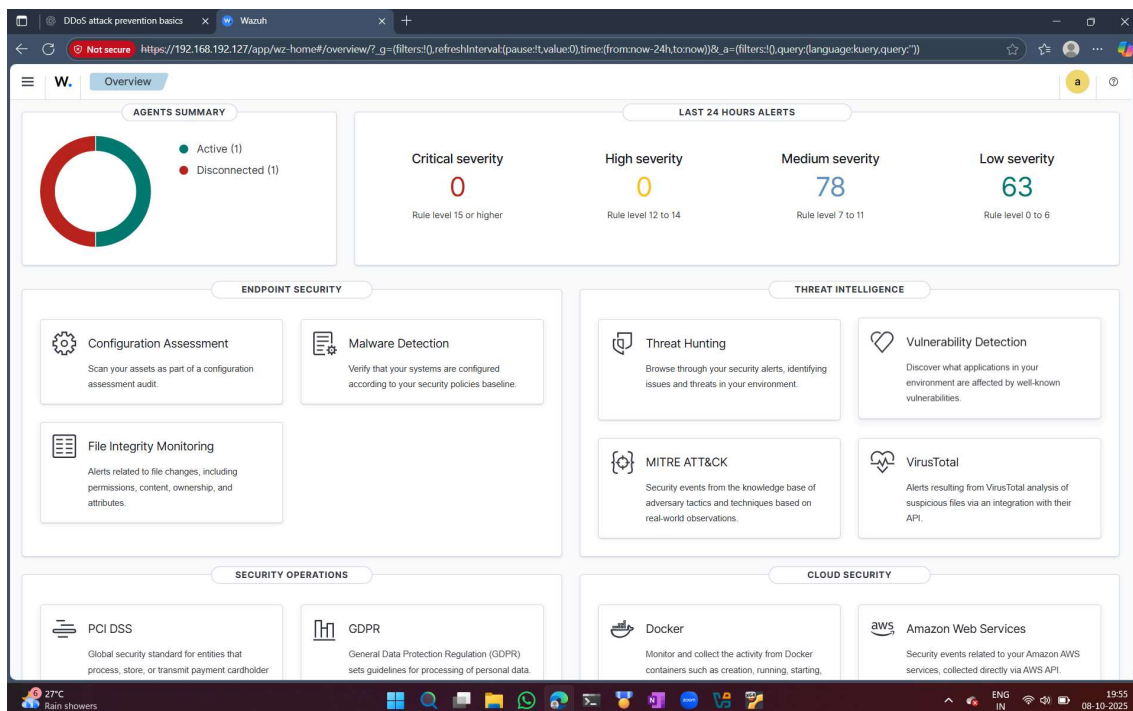


Figure 1 — Wazuh Dashboard Overview showing connected agents and alert summary.

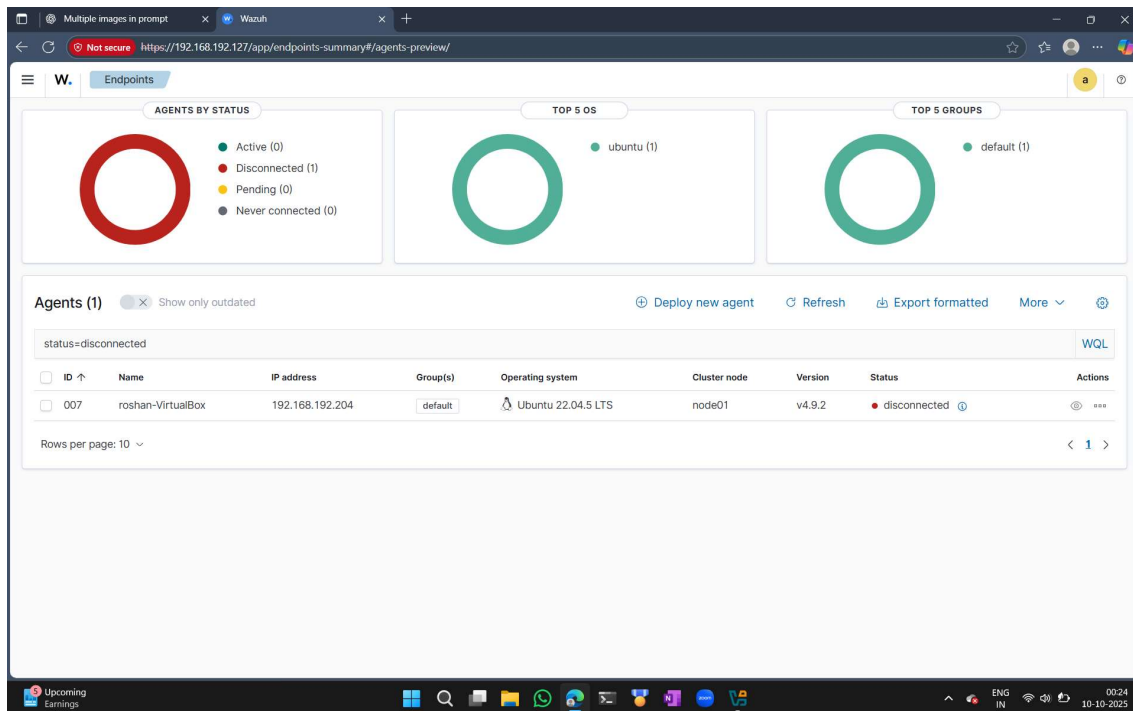
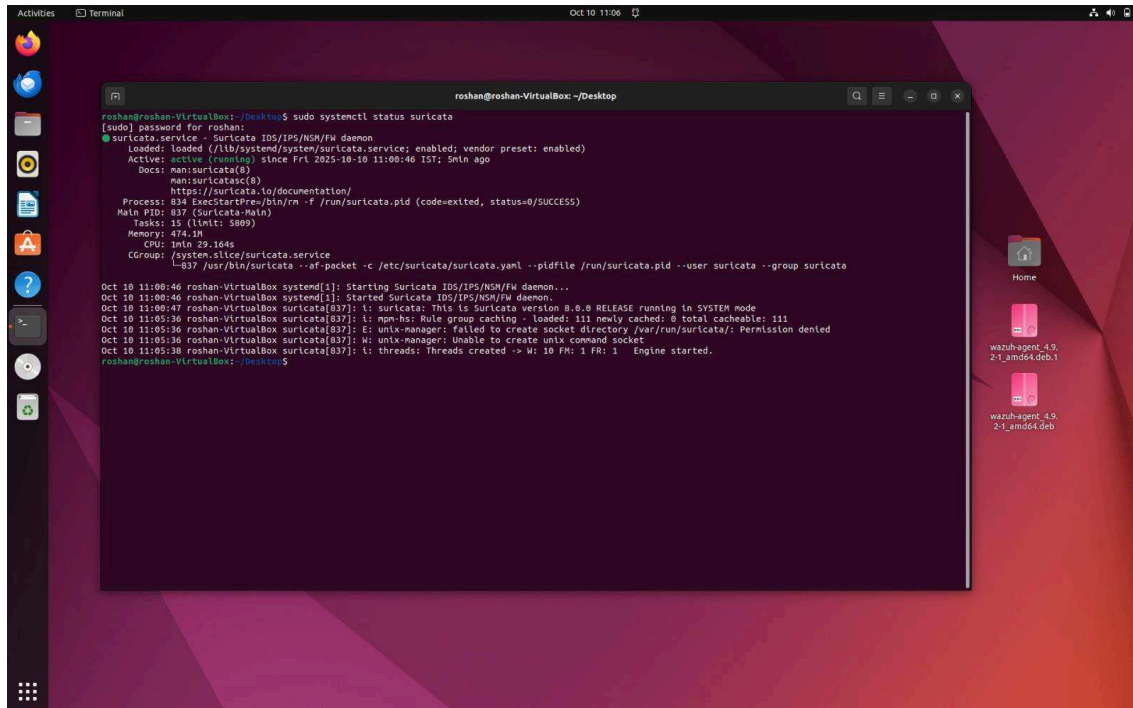


Figure 2 — Agent summary showing connected Wazuh agent.

- The Wazuh Agent was installed on a monitored endpoint and successfully connected to the Wazuh Manager.
- This screenshot shows one active agent that was disconnected agent in the system.

– Installing and Configuring Suricata : -

- Suricata was installed on the endpoint where the Wazuh agent is running.
- It was configured to monitor network traffic and generate alerts stored in eve.json.
- The configuration file used: /etc/suricata/suricata.yaml



```
roshan@roshan-VirtualBox: ~/Desktop
[sudo] password for roshan:
● suricata.service - Suricata IDS/IPS/NSM/FW daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2025-10-10 11:00:46 IST; 5min ago
     Docs: man:suricata(8)
           man:suricata(8)
           https://suricata.io/documentation/
   Process: 834 ExecStartPre=/bin/rm -f /run/suricata.pid (code=exited, status=0/SUCCESS)
   Main PID: 837 (Suricata-Main)
     Tasks: 15 (limit: 2880)
   Memory: 474.1M
     CPU: 1min 29.164s
   CGroup: /system.slice/suricata.service
           └─837 /usr/bin/suricata --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid --user suricata --group suricata

Oct 10 11:00:46 roshan-VirtualBox systemd[1]: Starting Suricata IDS/IPS/NSM/FW daemon...
Oct 10 11:00:46 roshan-VirtualBox systemd[1]: Started Suricata IDS/IPS/NSM/FW daemon.
Oct 10 11:00:47 roshan-VirtualBox suricata[837]: I: suricata: This is Suricata version 3.0.0 RELEASE running in SYSTEM mode
Oct 10 11:00:46 roshan-VirtualBox suricata[837]: I: ngn-hs: Rule group caching - loaded: 111 newly cached: 0 total cacheable: 111
Oct 10 11:05:36 roshan-VirtualBox suricata[837]: E: unix-manager: failed to create socket directory /var/run/suricata/: Permission denied
Oct 10 11:05:36 roshan-VirtualBox suricata[837]: W: unix-manager: Unable to create unix command socket
Oct 10 11:05:36 roshan-VirtualBox suricata[837]: I: threads: Threads created -> W: 10 FH: 1 FH: 1 Engine started.
roshan@roshan-VirtualBox:~/Desktop$
```

Figure 3 — Suricata installation and service status.

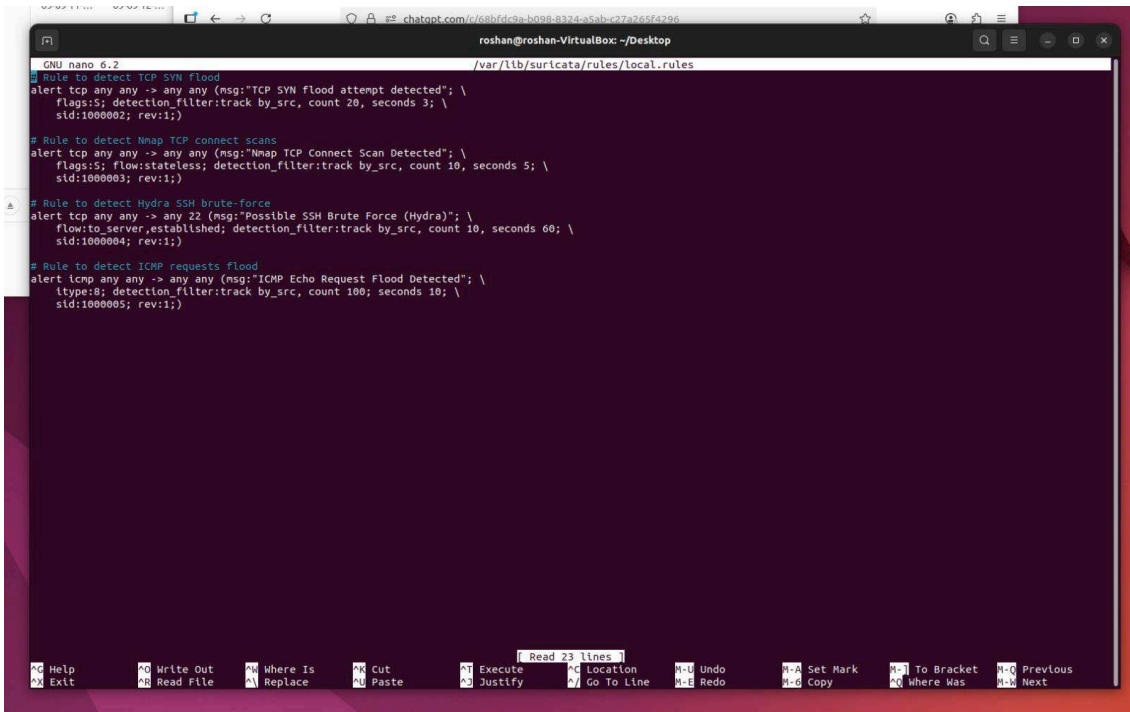
- Writing and Loading Custom Suricata rules : -

- Custom Suricata rules were written inside /etc/suricata/rules/local.rules.
- These rules are designed to detect specific test attacks (e.g., ICMP, SSH brute force, HTTP SQL injection, etc.).

- Example custom rule :

```
alert icmp any any -> any any (msg:"ICMP Echo Request Detected"; sid:1000001; rev:1;)
```

- The rule file was configured under the rule-files section of /etc/suricata/suricata.yaml for Suricata to load it.



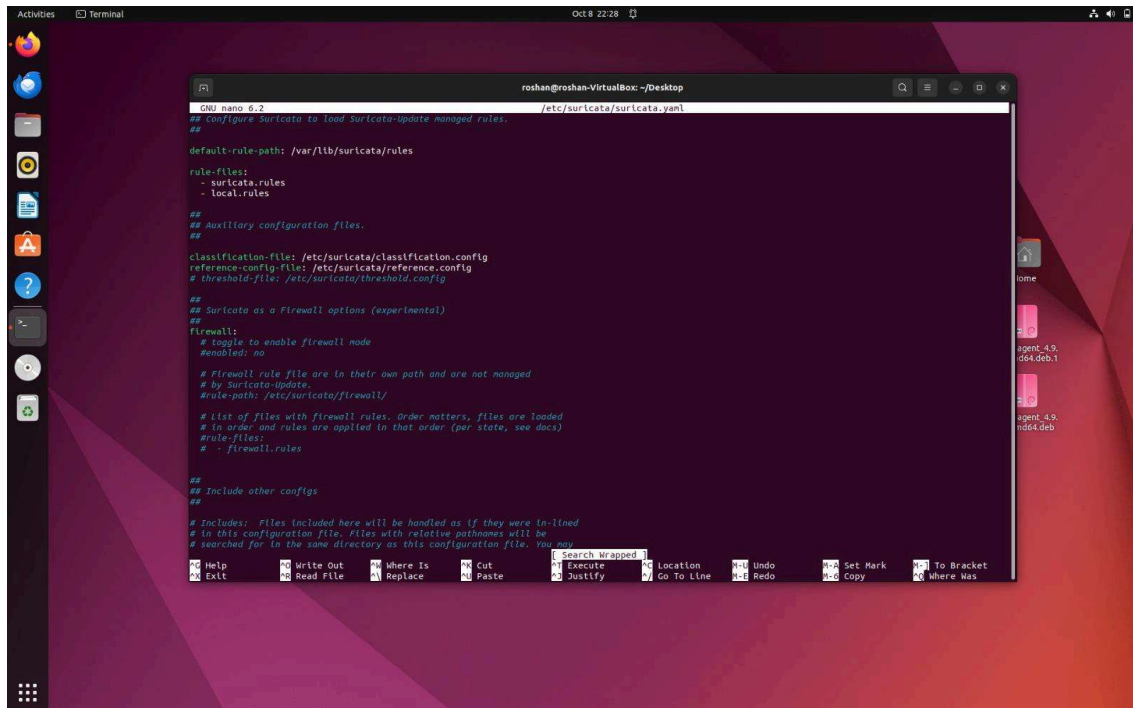
```
GNU nano 2.9 /var/lib/suricata/rules/local.rules
# Rule to detect TCP SYN flood
alert tcp any any -> any any (msg:"TCP SYN flood attempt detected"; \
  flags:S; detection_filter:track by_src, count 20, seconds 3; \
  sid:1000002; rev:1;)

# Rule to detect Nmap TCP connect scans
alert tcp any any -> any any (msg:"Nmap TCP Connect Scan Detected"; \
  flags:S; flow:stateless; detection_filter:track by_src, count 10, seconds 5; \
  sid:1000003; rev:1;)

# Rule to detect Hydra SSH brute-force
alert tcp any any -> any 22 (msg:"Possible SSH Brute Force (Hydra)"; \
  flow:to_server,established; detection_filter:track by_src, count 10, seconds 60; \
  sid:1000004; rev:1;)

# Rule to detect ICMP requests flood
alert icmp any any -> any any (msg:"ICMP Echo Request Flood Detected"; \
  ltype:S; detection_filter:track by_src, count 100, seconds 10; \
  sid:1000005; rev:1;)
```

- These are the Suricata Custom rules that has to be written only in the file /etc/suricata/rules/local.rules
- The Suricata default in-built rules are stored in the directory /var/lib/suricata/rules. This directory usually contains the default/system-installed Suricata rules, often from the community rulesets like Emerging threats.



```
GNU nano 2.2 /etc/suricata/suricata.yaml
# Configure Suricata to load Suricata Update managed rules.
##

default-rule-path: /var/lib/suricata/rules

rule-files:
- suricata.rules
- local.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)
##
firewall:
# Toggle to enable firewall mode
#enabled: no

# Firewall rule files are in their own path and are not managed
# by Suricata-Update.
#rule-path: /etc/suricata/firewall/

# List of files with firewall rules. Order matters, files are loaded
# in order and rules are applied in that order (per state, see docs)
#rule-files:
# - firewall.rules

## Include other configs
##

# Includes: Files included here will be handled as if they were included
# in this configuration file. Files with relative pathnames will be
# searched for in the same directory as this configuration file. You may
```

- Here the default rule-path given is /var/lib/suricata/rules in the suricata configuration file /etc/suricata/suricata.yaml
- Here Suricata does not load the suricata custom rules because custom rules file path /etc/suricata/rules/local.rules is not configured under the rule-files section in the suricata configuration file.
- If we write the suricata custom rules in the file /var/lib/suricata/rules/local.rules it may cause overwritten problem with the file /var/lib/suricata/rules/local.rules when we run suricata-update tool.
- When you run sudo suricata-update, Suricata fetches new rules and may recreate or replace /var/lib/suricata/rules/local.rules with a blank or default version.
- **Issues caused during overwritten problem :**
 - Custom rules lost - your manually created rules (e.g., ICMP, SSH brute force, SQLi) get replaced or deleted.
 - Alerts stop generating - since suricata no longer loads your custom rules, you stop getting alerts for your test attacks.
- So, to overcome this we may “Symbolic link” which is also known as symlink between the files /var/lib/suricata/rules/local.rules and /etc/suricata/rules/local.rules
- A symlink is like a *shortcut* or *pointer* in Linux.
- It links one file or directory to another — so if a program looks for a file in one location, it can actually be read from another location transparently.
- Otherwise, configure the direct suricata custom rules file as /etc/suricata/rules/local.rules under the rule-files section in the suricata configuration file suricata.yaml

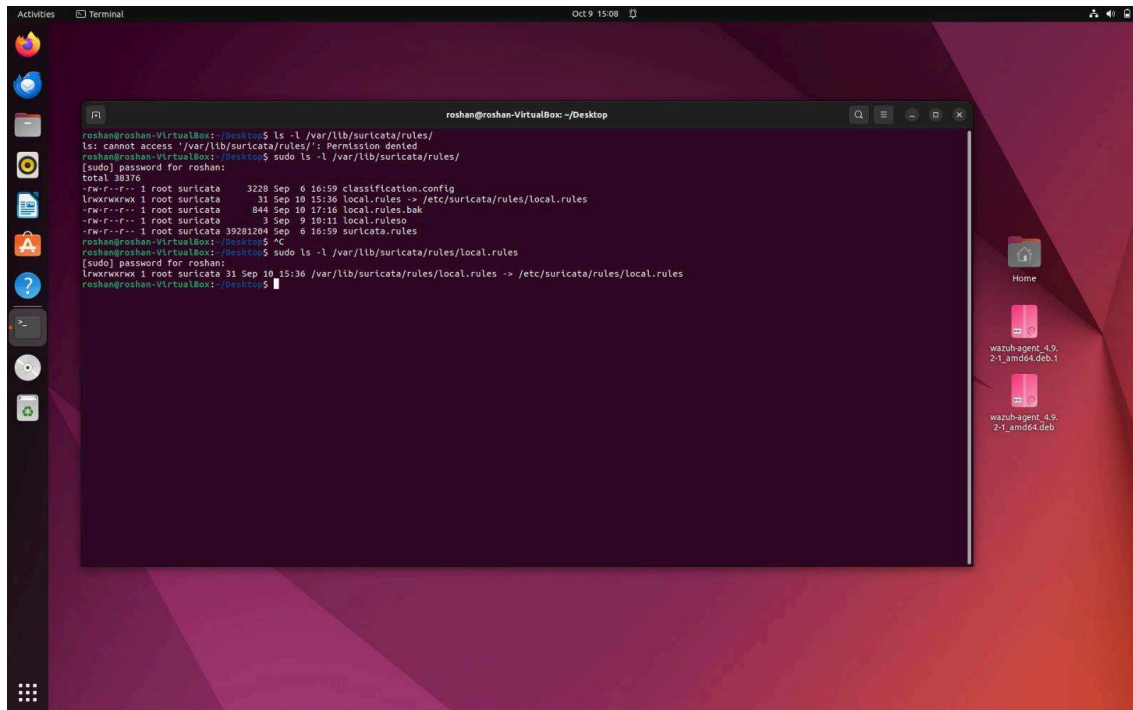


Figure 5 — Symlink showing link between default and custom rule directories.

– Triggering and Capturing Alerts : -

- To test the Suricata setup, sample attacks were simulated (such as ICMP ping or TCP port scans) against the monitored host.
- These actions triggered alerts that were logged in /var/log/suricata/eve.json.
- To read the logs in human readable format we install the tool 'jq'.

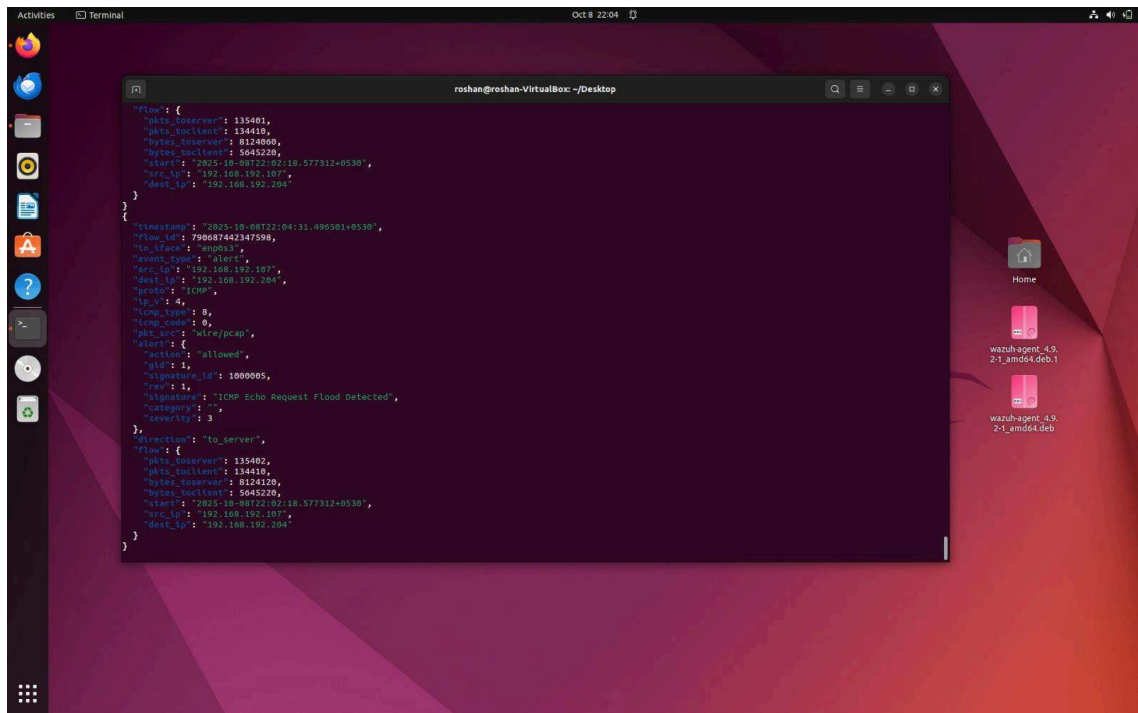


Figure 6 — Test attack traffic generated to trigger Suricata rule.

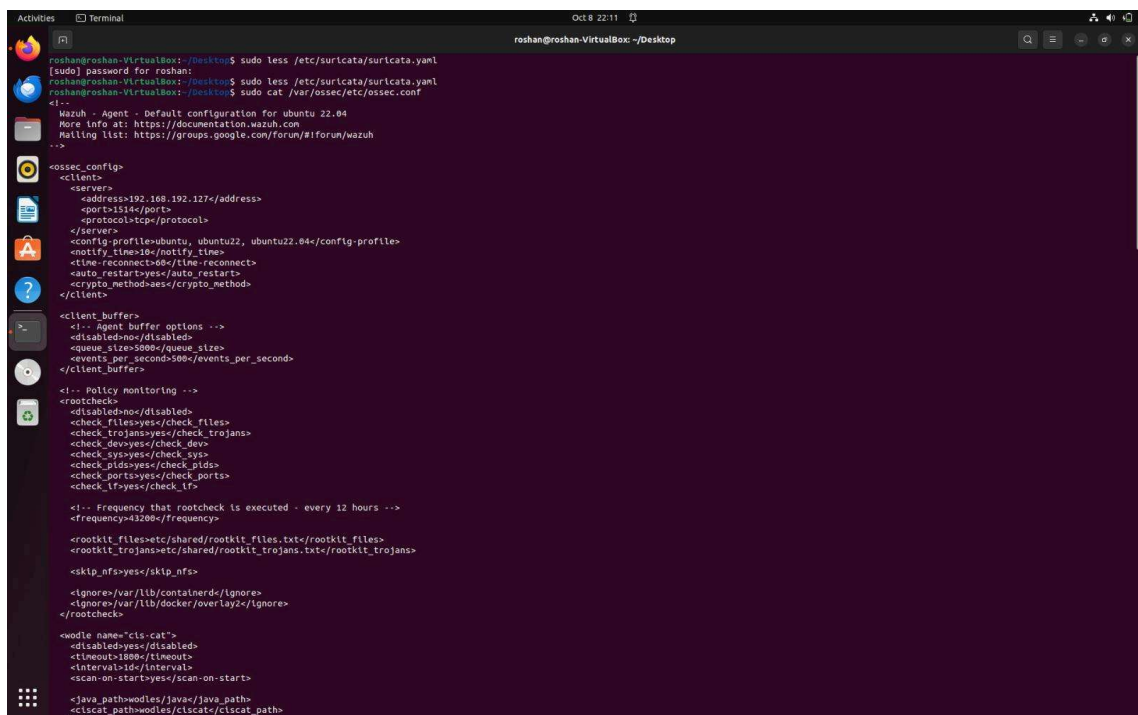


Figure 7 — Wazuh agent Configuration file send lpgs to the Wazuh Manager IP Address 192.168.192.127

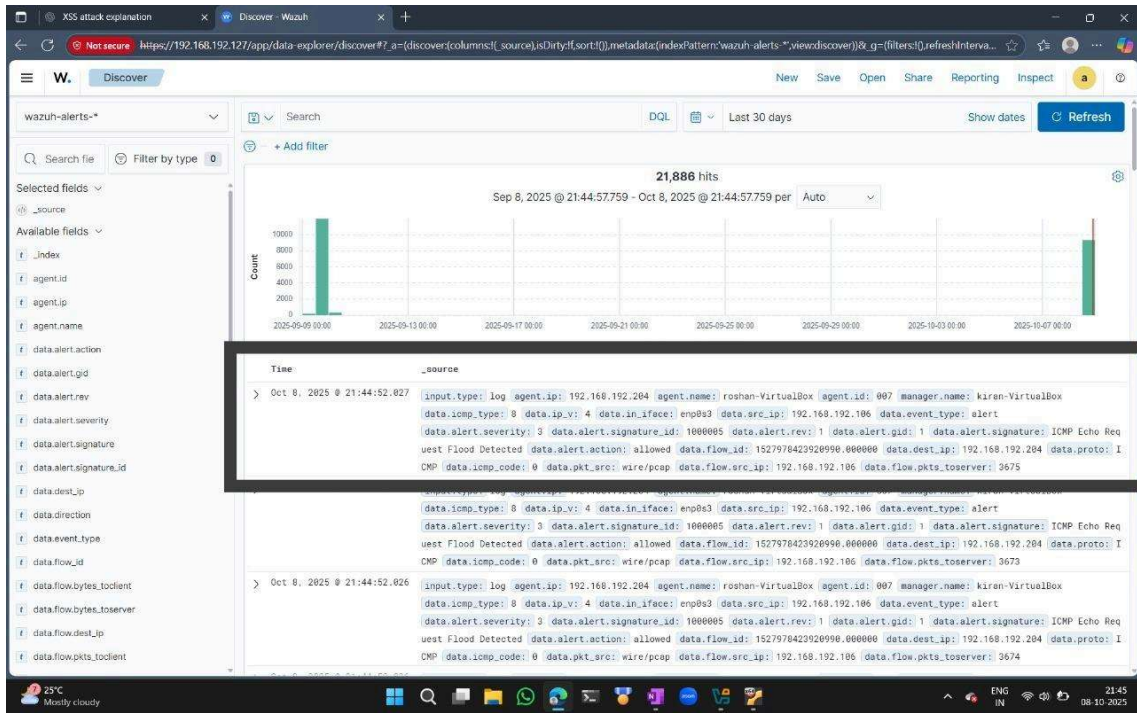


Figure 9 — ICMP Flood Suricata rule alerts successfully displayed in Wazuh Dashboard.

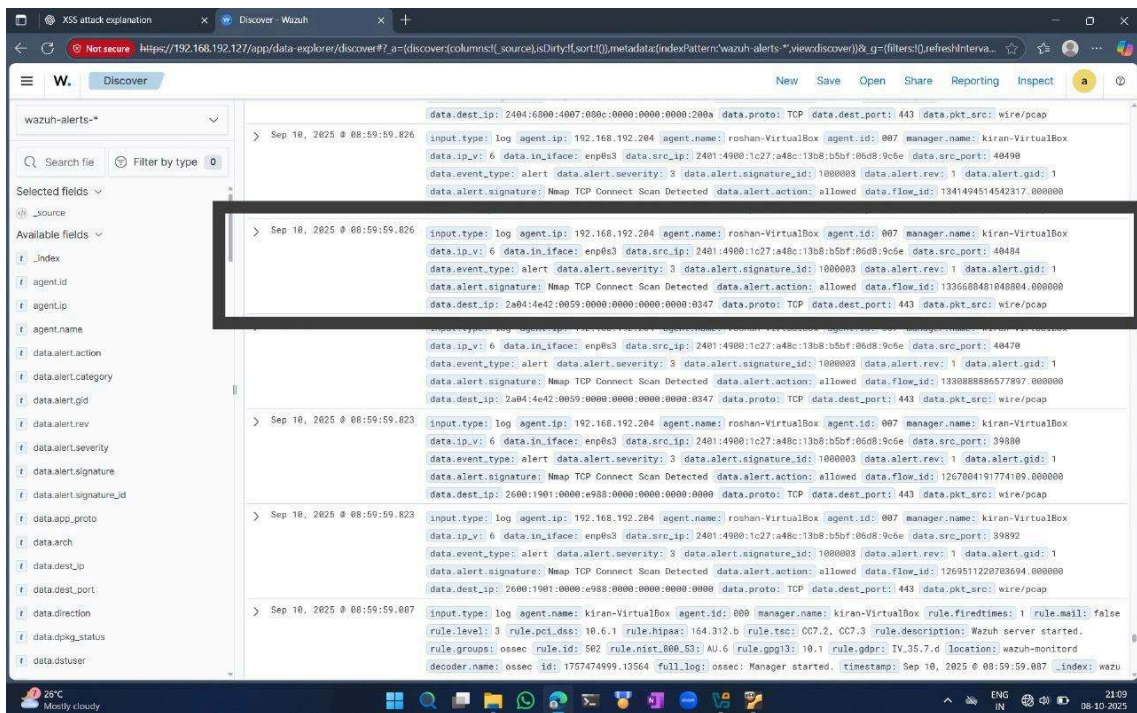


Figure 10 — Nmap Scan detection Suricata rule alerts successfully displayed in Wazuh Dashboard.

– Understanding Alerts Severities : -

Wazuh categorizes alerts based on rule levels (0–15) :

- 0–6 → Low severity
- 7–11 → Medium severity
- 12–14 → High severity
- 15+ → Critical severity

The dashboard displays alert counts by severity level over the last 24 hours.

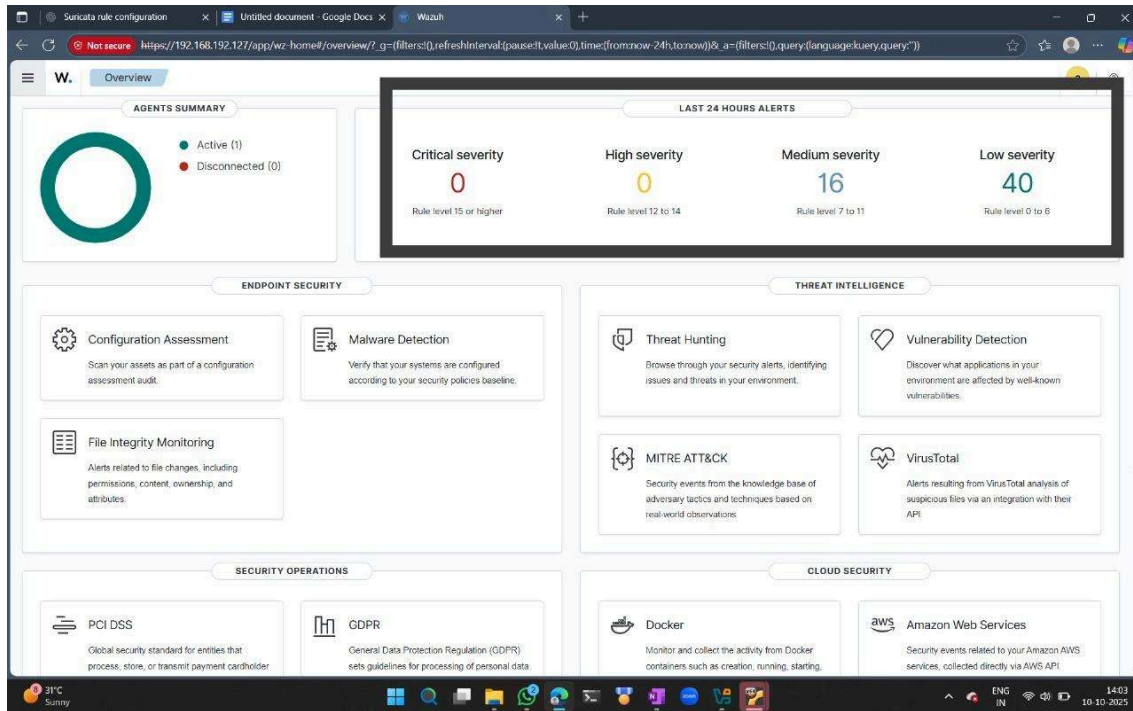


Figure 11 — Severity distribution of alerts in the Wazuh Dashboard.

– Threat Hunting and MITRE ATT&CK Mapping : -

- Wazuh integrates with **MITRE ATT&CK** to classify detected events based on known adversary tactics and techniques.
- This helps security analysts understand the context and purpose of detected behaviors.

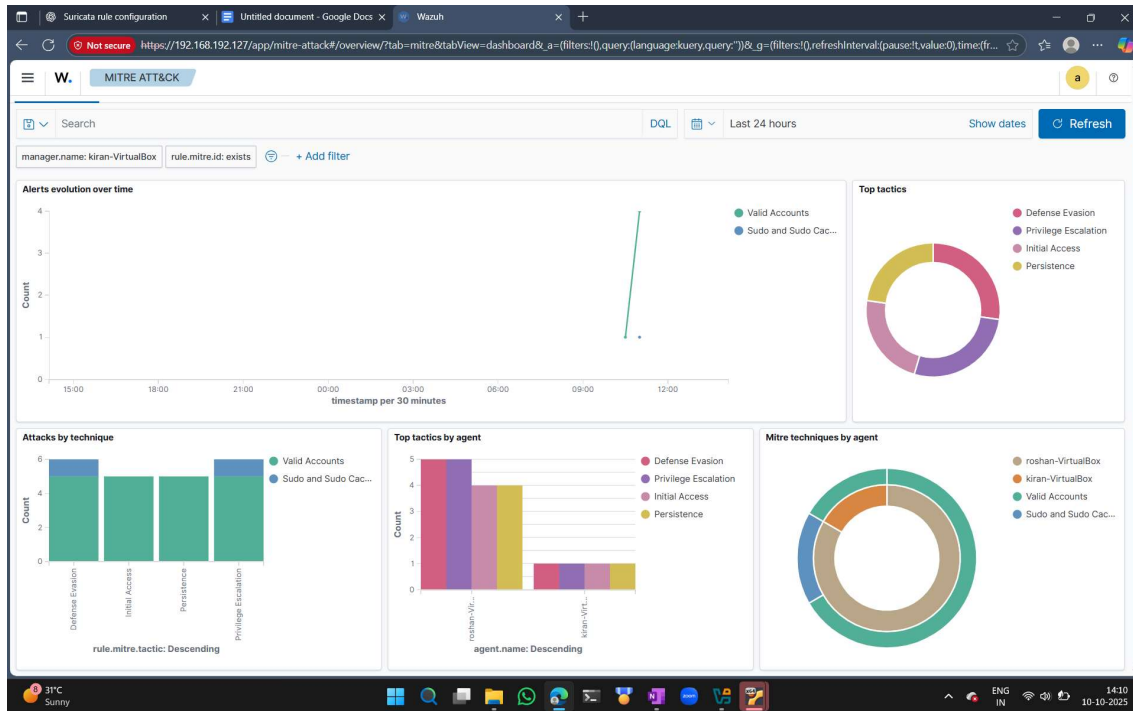


Figure 12 — MITRE ATT&CK mapping of detected threats in Wazuh.

– Conclusion : -

- This project demonstrated how Wazuh and Suricata can be integrated for real-time intrusion detection and alert visualization.
- By creating and loading custom Suricata rules, network attacks were successfully detected and displayed on the Wazuh Dashboard.
- The setup provides a complete open-source SIEM + IDS monitoring environment.