Integration of Suricata with Wazuh!

SAMEER HASSAN

Wazuh lab

Github-link: GitHub - sameerhassancode/Wazuh-labs

Linkedin: https://www.linkedin.com/in/sameer-hassan-15a428255/

Wazuh is an open-source security platform that provides unified XDR and SIEM capabilities, including threat detection, compliance monitoring, and incident response. It collects and analyzes data from endpoints and networks to help organizations detect and respond to security threats in real time.

Suricata is an open-source network threat detection engine that offers intrusion detection (IDS), intrusion prevention (IPS), and network security monitoring (NSM). It inspects network traffic using deep packet inspection and signature-based detection techniques to identify suspicious activity.

When integrated, **Wazuh and Suricata** form a powerful security solution. Suricata generates detailed alerts from network traffic, while Wazuh collects and correlates these alerts to provide centralized monitoring, analysis, and automated response—enhancing the overall security visibility and incident response capabilities of any organization.

Suricata:

Sudo apt-get install Suricata -y

```
┌──(root㉿kali)-[~]
└─# sudo apt-get install suricata -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  isa-support libfdt1 libhtp2 libhyperscan5 libnetfilter-log1 librte-bus-pci25 librte-bus-vdev25 librte-eal25 librte-ethdev25 librte-hash25 librte-ip-frag25 librte-kvargs25 librte-log25 librte-mbuf25 librte
  librte-net-bond25 librte-net25 librte-pci25 librte-rcu25 librte-ring25 librte-sched25 librte-telemetry25 libxdp1 sse3-support sse4.2-support suricata-update
Suggested packages:
  libtcmalloc-minimal4
The following NEW packages will be installed:
  isa-support libfdt1 libhtp2 libhyperscan5 libnetfilter-log1 librte-bus-pci25 librte-bus-vdev25 librte-eal25 librte-ethdev25 librte-hash25 librte-ip-frag25 librte-kvargs25 librte-log25 librte-mbuf25 librte
  librte-net-bond25 librte-net25 librte-pci25 librte-rcu25 librte-ring25 librte-sched25 librte-telemetry25 libxdp1 sse3-support sse4.2-support suricata suricata-update
0 upgraded, 28 newly installed, 0 to remove and 1310 not upgraded.
Need to get 6,691 kB of archives.
After this operation, 30.2 MB of additional disk space will be used.
Get:1 http://mirror.ourhost.az/kali kali-rolling/main amd64 isa-support amd64 27 [14.9 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 sse3-support amd64 27 [3,736 B]
Get:3 http://kali.download/kali kali-rolling/main amd64 sse4.2-support amd64 27 [3,692 B]
Get:4 http://kali.download/kali kali-rolling/main amd64 libhtp2 amd64 1:0.5.50-1 [72.9 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 libhyperscan5 amd64 5.4.2-3 [2,695 kB]
27% [5 libhyperscan5 1,904 kB/2,695 kB 71%] [Waiting for headers] [Waiting for headers]
```

Verify installation:

Suricata -v

```
┌──(root㉿kali)-[~]
└─# suricata -V
Suricata 7.0.10
USAGE: suricata [OPTIONS] [BPF FILTER]

        -c <path>                           : path to configuration file
        -T                                  : test configuration file (use with -c)
        -i <dev or ip>                      : run in pcap live mode
        -F <bpf filter file>                : bpf filter file
        -r <path>                           : run in pcap file/offline mode
        -q <qid[:qid]>                      : run in inline nfqueue mode (use colon to specify a range of queues)
        -s <path>                           : path to signature file loaded in addition to suricata.yaml settings (optional)
        -S <path>                           : path to signature file loaded exclusively (optional)
        -l <dir>                            : default log directory
        -D                                  : run as daemon
        -k [all|none]                       : force checksum check (all) or disabled it (none)
        -V                                  : display Suricata version
        -v                                  : be more verbose (use multiple times to increase verbosity)
        --list-app-layer-protos             : list supported app layer protocols
        --list-keywords[=all|csv|<kword>]   : list keywords implemented by the engine
        --list-runmodes                     : list supported runmodes
        --runmode <runmode_id>              : specific runmode modification the engine should run.  The argument
                                              supplied should be the id for the runmode obtained by running
                                              --list-runmodes
        --engine-analysis                   : print reports on analysis of different sections in the engine and exit.
                                              Please have a look at the conf parameter engine-analysis on what reports
                                              can be printed
        --pidfile <file>                    : write pid to this file
        --init-errors-fatal                 : enable fatal failure on signature init error
```

Now start & enable Suricata

Sudo systemctl start Suricata

Sudo systemctl enable Suricata

```
┌──(root㉿kali)-[~]
└─# sudo systemctl restart suricata

┌──(root㉿kali)-[~]
└─# sudo systemctl enable suricata
Synchronizing state of suricata.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable suricata
Created symlink '/etc/systemd/system/multi-user.target.wants/suricata.service' → '/usr/lib/systemd/system/suricata.service'.

┌──(root㉿kali)-[~]
└─#
```

Update the Suricata to download default EMT rules.

Sudo Suricata-update

```
┌──(root㉿kali)-[~]
└─# sudo suricata-update

19/6/2025 -- 15:57:56 - <Info> -- Using data-directory /var/lib/suricata.
19/6/2025 -- 15:57:56 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
19/6/2025 -- 15:57:56 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
19/6/2025 -- 15:57:56 - <Info> -- Found Suricata version 7.0.10 at /usr/bin/suricata.
19/6/2025 -- 15:57:56 - <Info> -- Loading /etc/suricata/suricata.yaml
19/6/2025 -- 15:57:56 - <Info> -- Disabling rules for protocol pgsql
19/6/2025 -- 15:57:56 - <Info> -- Disabling rules for protocol modbus
19/6/2025 -- 15:57:56 - <Info> -- Disabling rules for protocol dnp3
19/6/2025 -- 15:57:56 - <Info> -- Disabling rules for protocol enip
19/6/2025 -- 15:57:56 - <Info> -- No sources configured, will use Emerging Threats Open
19/6/2025 -- 15:57:56 - <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-7.0.10/emerging.rules.tar.gz.
  10% - 507904/4972796
```

Now if you want to check the rules goto rules file

Cd /var/lib/Suricata/rules/

```
┌──(root㉿kali)-[~]
└─# cd /var/lib/suricata/rules

┌──(root㉿kali)-[/var/lib/suricata/rules]
└─# ls
lassification.config  suricata.rules

┌──(root㉿kali)-[/var/lib/suricata/rules]
└─#
```

Open the Suricata.rule

Cat Suricata.rules

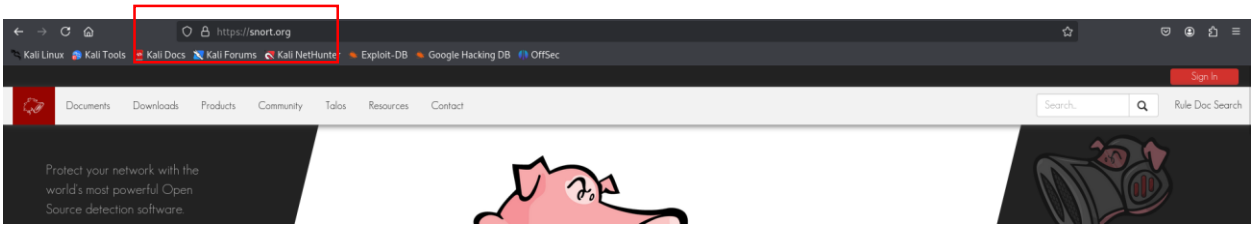
```
┌──(root㉿kali)-[/var/lib/suricata/rules]
└─# cat suricata.rules
alert ip any any -> any any (msg:"SURICATA Applayer Mismatch protocol both directions"; flow:established; app-layer-event:applayer_mismatch_protocol_both_directions; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode; s
id:2260000; rev:1;)
alert ip any any -> any any (msg:"SURICATA Applayer Wrong direction first Data"; flow:established; app-layer-event:applayer_wrong_direction_first_data; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260001; re
v:1;)
alert ip any any -> any any (msg:"SURICATA Applayer Detect protocol only one direction"; flow:established; app-layer-event:applayer_detect_protocol_only_one_direction; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode;
 sid:2260002; rev:1;)
alert ip any any -> any any (msg:"SURICATA Applayer Protocol detection skipped"; flow:established; app-layer-event:applayer_proto_detection_skipped; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260003; rev:1
;)
alert tcp any any -> any any (msg:"SURICATA Applayer No TLS after STARTTLS"; flow:established; app-layer-event:applayer_no_tls_after_starttls; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260004; rev:2;)
alert tcp any any -> any any (msg:"SURICATA Applayer Unexpected protocol"; flow:established; app-layer-event:applayer_unexpected_protocol; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260005; rev:1;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 packet too small"; decode-event:ipv4.pkt_too_small; classtype:protocol-command-decode; sid:2200000; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 header size too small"; decode-event:ipv4.hlen_too_small; classtype:protocol-command-decode; sid:2200001; rev:2;)
```

Now restart the Suricata for further configuration

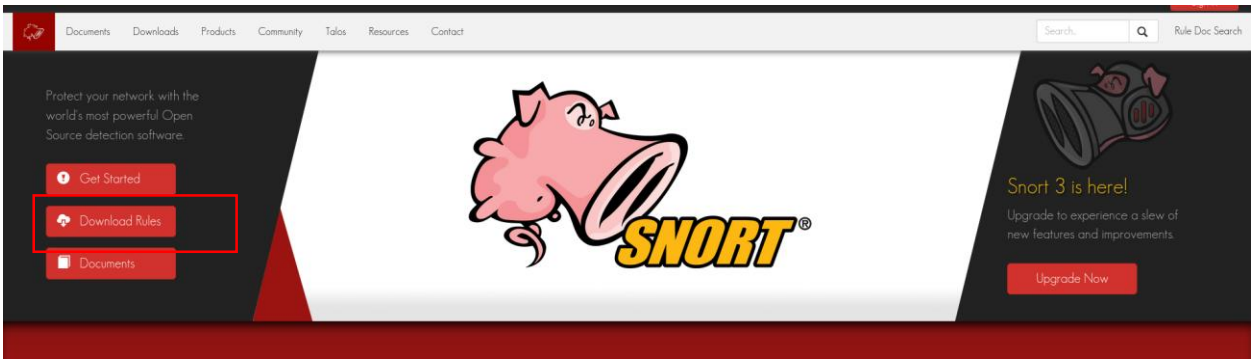
```
┌──(root㉿kali)-[/var/lib/suricata/rules]
└─# sudo systemctl restart suricata
```

Now let’s use Snort IDS rules with Suricata

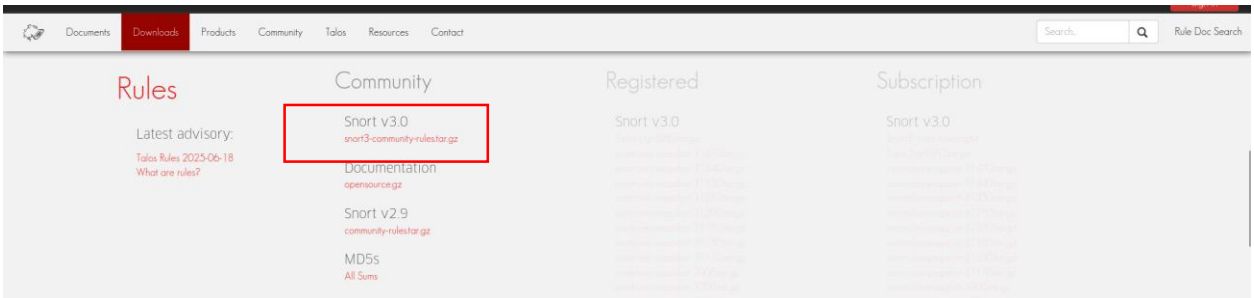
Visit snort.org and download the community rules



Now click download rules



Now click on community rules



Extract the rules

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# 7z e snort3-community-rules.tar.gz
```

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ls
AUTHORS  community-rules  community.rules  LICENSE  sid-msg.map  snort3-community-rules.tar.gz  VRT-License.txt

┌──(root㉿kali)-[/home/kali/Downloads]
└─#
```

Now Copy the Community.rules and paste it the suricata rule folder.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ls
AUTHORS  community-rules  community.rules  LICENSE  sid-msg.map  snort3-community-rules.tar.gz  VRT-License.txt

┌──(root㉿kali)-[/home/kali/Downloads]
└─#
```

sudo cp community.rules /var/lib/suricata/rules/

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# sudo cp community.rules /var/lib/suricata/rules/
```

Now change the directory to suricata and check your ip with ip a command

```
┌──(root㉿kali)-[/etc/suricata]
└─# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b4:a6:09 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.121/24 brd 192.168.0.255 scope global dynamic noprefixroute eth0
       valid_lft 2376sec preferred_lft 2376sec
    inet6 fe80::a00:27ff:feb4:a609/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(root㉿kali)-[/etc/suricata]
└─#
```

My ip is 192.168.0.121 but in suricata configuration I will use it as 192.168.0.0/24 to work against all IP’s

Now open the suricata.yaml file for further configuration

```
┌──(root㉿kali)-[/etc/suricata]
└─# nano suricata.yaml
```

```
  GNU nano 8.3                                   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 generated by Suricata 7.0.10.
suricata-version: "7.0"
```

Now find this section

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

And make these changes

```
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    # HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
    HOME_NET: "[192.168.0.0/24]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

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

Now scroll down and find the rule file path section

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

rule-files:
  - suricata.rules

##
## Auxiliary configuration files.
##

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

##
```

And add the community.rules file in the file path

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

rule-files:
  - suricata.rules
  - community.rules

##
## Auxiliary configuration files.
##
```

Now save the file and type ifconfig to check the network interface

It's showing eht0 interface in my system check your system interface and note the name of interface

```
┌──(root㉿kali)-[/var/lib/suricata/rules]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.121  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:feb4:a609  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:b4:a6:09  txqueuelen 1000  (Ethernet)
        RX packets 31160  bytes 25371631 (24.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10723  bytes 1565704 (1.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 146  bytes 11952 (11.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 146  bytes 11952 (11.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Now again open th suricata.yaml file and look for af=packet section and doubale check the interface i.e eth0

Note: in suricata 8.0 or above it auto detect the interface

```
# Linux high speed capture support
af-packet:
  - interface: eth0
    # 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
```

Now save the file and restart the suricata and check the status

```
┌──(root㉿kali)-[/etc/suricata]
└─# sudo systemctl restart suricata


┌──(root㉿kali)-[/etc/suricata]
└─# sudo systemctl status suricata

● suricata.service - Suricata IDS/IDP daemon
     Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-06-19 16:45:46 EDT; 9s ago
 Invocation: 1802eb1d29fa4f3e98bed2390e634f7f
       Docs: man:suricata(8)
             man:suricatasc(8)
             https://suricata.io/documentation/
    Process: 9641 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid (code=exited, status=0/SUCCESS)
   Main PID: 9642 (Suricata-Main)
      Tasks: 1 (limit: 12632)
     Memory: 360.5M (peak: 360.5M)
        CPU: 9.270s
     CGroup: /system.slice/suricata.service
             └─9642 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

Jun 19 16:45:46 kali systemd[1]: Starting suricata.service - Suricata IDS/IDP daemon...
Jun 19 16:45:46 kali suricata[9641]: i: suricata: This is Suricata version 7.0.10 RELEASE running in SYSTEM mode
Jun 19 16:45:46 kali systemd[1]: Started suricata.service - Suricata IDS/IDP daemon.

┌──(root㉿kali)-[/etc/suricata]
└─#
```

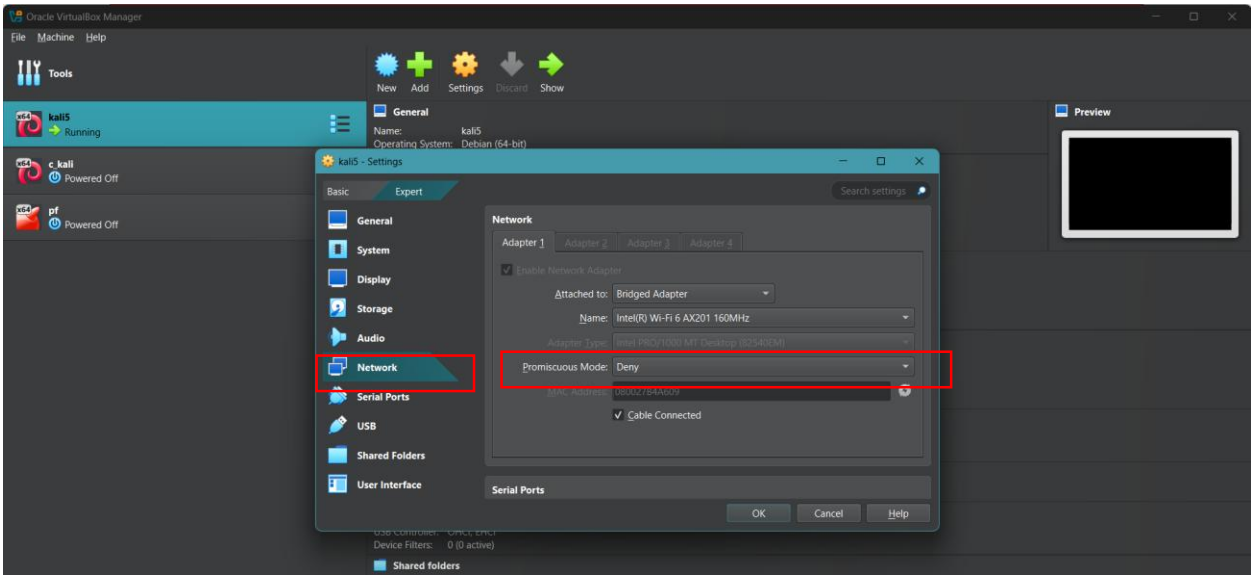
Now run the suricata in test mode to check working

sudo suricata -T -c /etc/suricata/suricata.yaml -v

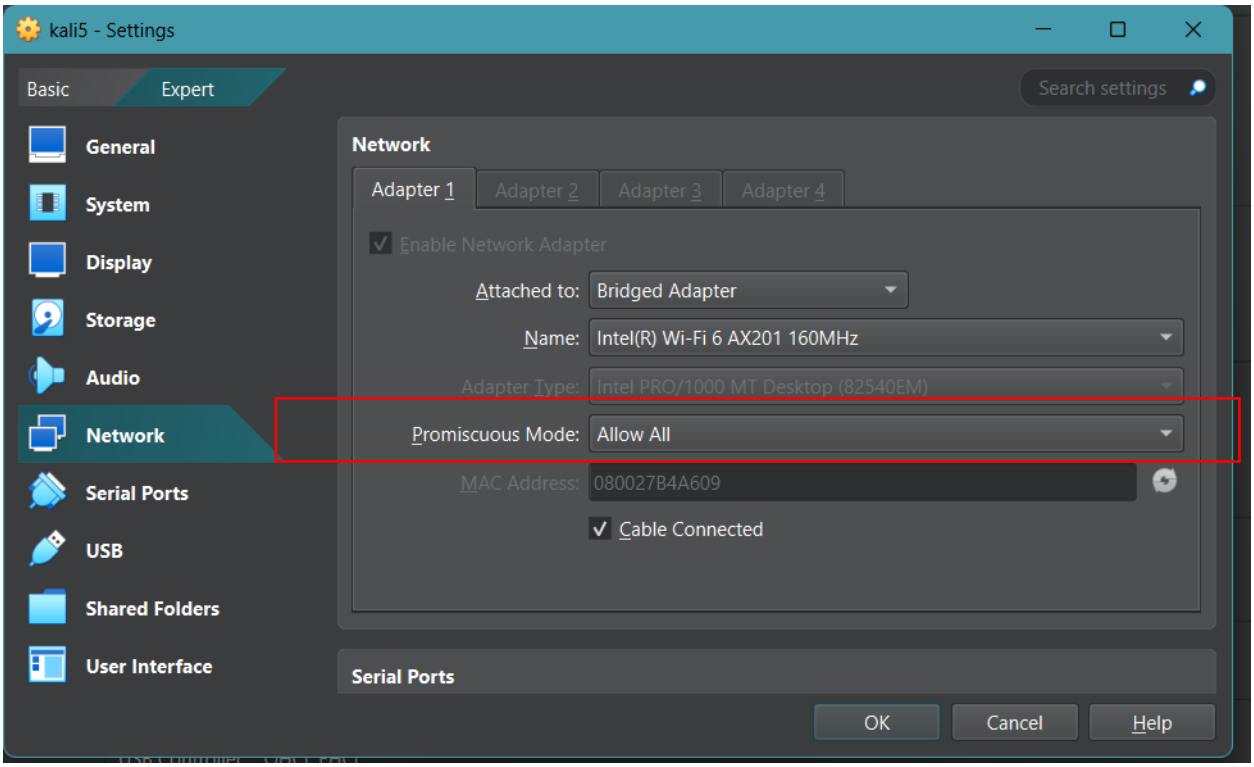
```
┌──(root㉿kali)-[/var/lib/suricata/rules]
└─# sudo suricata -T -c /etc/suricata/suricata.yaml -v

Notice: suricata: This is Suricata version 7.0.10 RELEASE running in SYSTEM mode
Info: cpu: CPUs/cores online: 4
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Info: detect: 2 rule files processed. 44721 rules successfully loaded, 0 rules failed, 0
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 44724 signatures processed. 1220 are IP-only rules, 4539 are inspecting packet payload, 38639 inspect application layer, 109 are decoder event only
Notice: suricata: Configuration provided was successfully loaded. Exiting.
```

It’s working and properly configured

Now open the vm machine setting and go to network menu

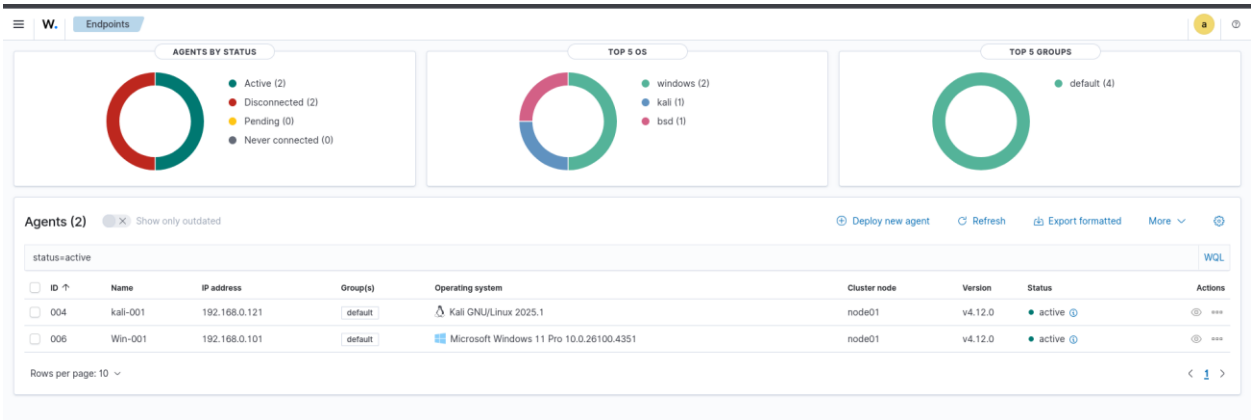


Chang it from deny to **allow all**



Now start the Wazuh:

Wazuh-Dashboard



My IDS/IPS is installed on kali agent machine which is already configure

Now let's configure the Wazuh-agent **ossec.conf** file for suricata.

Go to /var/ossec/etc/ directory and there you will see the ossec.conf file

Open ossec.conf file

Now look for local file section to add the path of suricata logs path

now again open the terminal and go to path /var/log/suricata to confirm the **eve.json** log file. We will add the path of this file in the ossec.conf file

```
┌──(root㉿kali)-[/var/lib/suricata]
└─# cd /var/log/suricata

┌──(root㉿kali)-[/var/log/suricata]
└─# ls
eve.json  fast.log  stats.log  suricata.log

┌──(root㉿kali)-[/var/log/suricata]
└─#
```

Now again open the **ossec.conf** file and add this code

<localfile>

<log_format></ log_format >

<location></location>

<localfile>

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

</ossec_config>
```

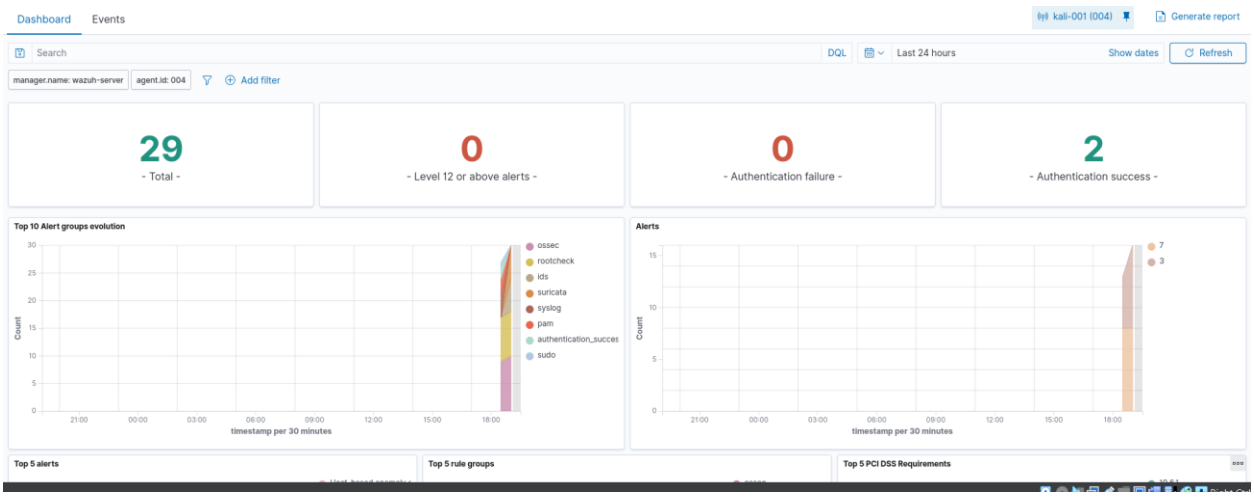
Now restart the Wazuh agent

```
┌──(root㉿kali)-[/var/ossec/etc]
└─# systemctl restart  wazuh-agent.service

┌──(root㉿kali)-[/var/ossec/etc]
└─#
```

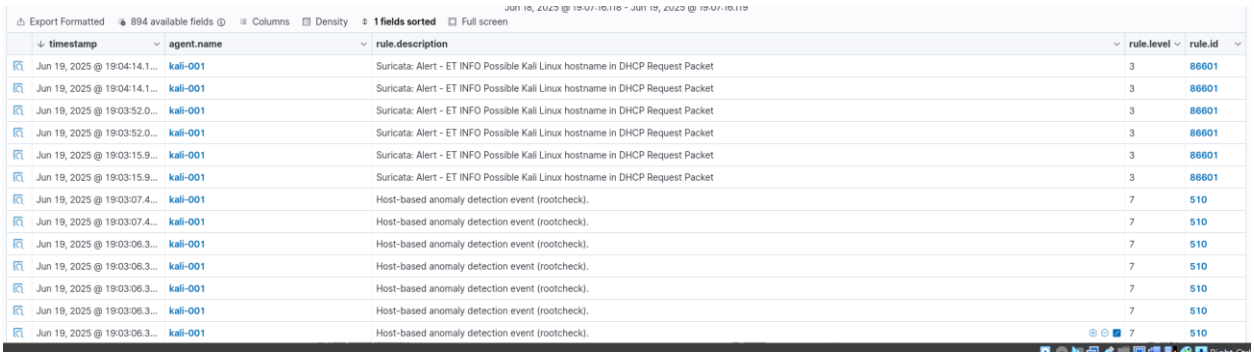
Now scan the system with nmap

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV 192.168.0.121
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-19 19:05 EDT
Nmap scan report for 192.168.0.121
Host is up (0.00030s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.9p1 Debian 3 (protocol 2.0)
MAC Address: 08:00:27:B4:A6:09 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds

┌──(kali㉿kali)-[~]
└─$
```

Now let’s move to Wazuh-Dashboard



Sucicata Logs



Export Formatted | 894 available fields | Columns | Density | 1 fields sorted | Full screen

timestamp	agent.name	rule.description	rule.level	rule.id
Jun 19, 2025 @ 19:04:14.1...	kali-001	Suricata: Alert - ET INFO Possible Kali Linux hostname in DHCP Request Packet	3	86601
Jun 19, 2025 @ 19:04:14.1...	kali-001	Suricata: Alert - ET INFO Possible Kali Linux hostname in DHCP Request Packet	3	86601
Jun 19, 2025 @ 19:03:52.0...	kali-001	Suricata: Alert - ET INFO Possible Kali Linux hostname in DHCP Request Packet	3	86601
Jun 19, 2025 @ 19:03:52.0...	kali-001	Suricata: Alert - ET INFO Possible Kali Linux hostname in DHCP Request Packet	3	86601
Jun 19, 2025 @ 19:03:15.9...	kali-001	Suricata: Alert - ET INFO Possible Kali Linux hostname in DHCP Request Packet	3	86601
Jun 19, 2025 @ 19:03:15.9...	kali-001	Suricata: Alert - ET INFO Possible Kali Linux hostname in DHCP Request Packet	3	86601
Jun 19, 2025 @ 19:03:07.4...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 19, 2025 @ 19:03:07.4...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 19, 2025 @ 19:03:06.3...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 19, 2025 @ 19:03:06.3...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 19, 2025 @ 19:03:06.3...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 19, 2025 @ 19:03:06.3...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 19, 2025 @ 19:03:06.3...	kali-001	Host-based anomaly detection event (rootcheck).	7	510

Document Details

View surrounding documents

View single document

data.flow.src_port	68
data.flow.start	2025-06-19T19:03:41.621783-0400
data.flow_id	1544641387278584.000000
data.in_iface	eth0
data.pkt_src	wire/pcap
data.proto	UDP
data.src_ip	0.0.0.0
data.src_port	68
data.timestamp	Jun 19, 2025 @ 19:04:03.580
decoder.name	json
id	1750374254.25376195
input.type	log
location	/var/log/suricata/eve.json
manager.name	wazuh-server
rule.description	Suricata: Alert - ET INFO Possible Kali Linux hostname in DHCP Request Packet
rule.firedtimes	6
rule.groups	ids, suricata
rule.id	86601
rule.level	3
rule.mail	false
timestamp	Jun 19, 2025 @ 19:04:14.102

Summary:

Wazuh provides powerful SIEM and endpoint security capabilities, offering log analysis, file integrity monitoring, and real-time threat detection across systems. Suricata enhances network security by delivering high-performance intrusion detection and prevention with deep packet inspection and rule-based alerts. Together, they create a comprehensive security solution that monitors both host and network layers for a stronger, more centralized threat defense.

Need training on Wazuh ?

Contact number: +923355345678

Email: sameeerishassan@gmail.com

LinkedIn: https://pk.linkedin.com/in/sameer-hassan-15a428255

Other SIEM

1. IBM Qradar

2. Splunk

3. Azure Sentinel