

PREPARED BY: ZAINAB ARIF

## Exploring Suricata



### What is Suricata?

#### **INTRODUCTION TO SURICATA:**

It is open source/free threat detection software/tool include IDS, IPS, and network monitoring.

Widely used in both private and public organizations.

Developed by community run /non-profit organization OSIF (Open Information Security Foundation).

Suricata has two operational modes:

**1)Active(IPS) :** Used to alert, log and block network traffic that matches specific rules.

**2)Passive(IDS) :** Used to identify, alert and log suspicious network traffic within a network.

Note: We cannot choose IPS directly instead of IDS because in IPS we might come across a lot of false positive cases or multiple issues, so initially we start with IDS.

Network Monitoring: Suricata analyze and log network traffic for troubleshooting and security insights.

#### **HOW SURICATA WORKS :**

It can be deployed as IDS on network to monitor network traffic.

It can be deployed as IPS in in-line mode to detect and stop malicious network traffic.

It identify malicious network traffic by predefined or set rules, when malicious network matched against a rule or set of rules, alert will generated and traffic will be logged.

#### **SURICATA KEY FEATURES :**

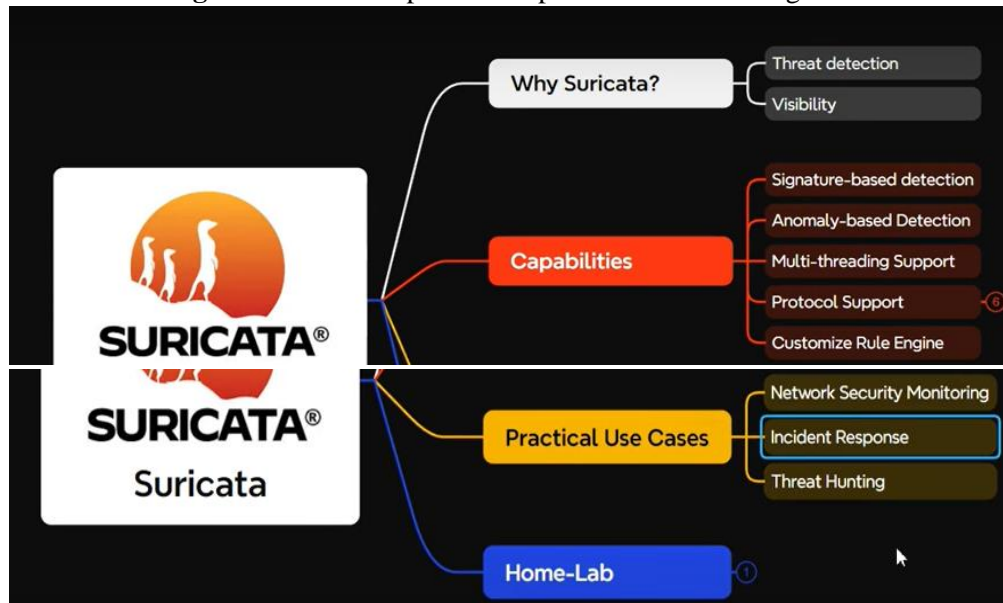
**Protocol Detection:** Detects protocols (like HTTP) on any port to spot malware and threats.

**Fast Performance:** Uses multiple cores for quick and efficient traffic monitoring.

**Detailed Logging:** Logs data for HTTP, DNS, and other protocols for easier analysis.

**Compliance Auditing:** Monitor networks to ensure adherence to security policies.

**Threat Hunting:** Detect unusual patterns or potential attacks using custom rules.



## How install and configure it?

Installing it on Ubuntu.

### Commands for Installation on Ubuntu:

```
sudo apt-get install software-properties-common
```

```
sudo add-apt-repository ppa:oisf/suricata-stable
```

```
sudo apt-get update
```

*Then can install the latest stable with:*

```
sudo apt-get install suricata
```

```
Jan 1 19:31
zainab@zainab-VMware-Virtual-Platform: ~
zainab@zainab-VMware-Virtual-Platform:~$ sudo apt-get install software-properties-common
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt-get update
[sudo] password for zainab:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.48).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu/
Suites: noble
Components: main
'
Description:
Suricata IDS/IPS/NSM stable packages
Homepage: suricata.io/
Terminal: oisf.net/

Suricata IDS/IPS/NSM - Suricata is a high performance Intrusion Detection and Prevention System and Network Security Monitoring engine.

Open Source and owned by a community run non-profit foundation, the Open Information Security Foundation (OISF). Suricata is developed by the OISF, its supporting vendors and the community.

This Engine supports:
- Multi-Threading - provides for extremely fast and flexible operation on multicore systems.
- Multi Tenancy - Per vlan/Per interface
- Uses Rust for most protocol detection/parsing
- TLS/SSL certificate matching/logging
- JA3 TLS client fingerprinting
- JA3S TLS server fingerprinting

Activate Windows
Go to Settings to activate Windows.

Jan 1 19:32
zainab@zainab-VMware-Virtual-Platform: ~
- File MD5/SHA1/SHA256 matching
- Gzip Decompression
- Fast IP Matching
- Datasets matching
- Rustlang enabled protocol detection
- Lua scripting

and many more great features -
i. Help: //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://pk.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://pk.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://pk.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Hit:1 http://pk.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://pk.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://pk.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu noble InRelease
Reading package lists... Done
zainab@zainab-VMware-Virtual-Platform:~$ sudo apt-get install suricata
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
suricata is already the newest version (1:7.0.8-0ubuntu0).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
zainab@zainab-VMware-Virtual-Platform:~$
```

## SURICATA SET UP:

Suricata is installed as see files in this

```
baek@:/etc/suricata: Is a directory
Trash zainab-VMware-Virtual-Platform:~$ cd /etc/suricata
zainab@zainab-VMware-Virtual-Platform:/etc/suricata$ ls
classification.config reference.config rules suricata.yaml suricata.yaml.dpkg-dist thresholds.config
zainab@zainab-VMware-Virtual-Platform:/etc/suricata$
```

Download and extract the Emerging Threats Suricata ruleset:

```
Jan 2 00:19
zainab@zainab-VMware-Virtual-Platform: /
zainab@zainab-VMware-Virtual-Platform:/etc/suricata$ cd ..
zainab@zainab-VMware-Virtual-Platform:/etc$ cd ..
zainab@zainab-VMware-Virtual-Platform:$ 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 mv rules/*.rules /etc/suricata/rules/
sudo chmod 640 /etc/suricata/rules/*.rules
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
  0     0    0     0    0     0      0      0  --:--:--  0:00:01  --:--:--   0
```

```
zainab@zainab-VMware-Virtual-Platform: /tmp
sudo tar -xvzf emerging.rules.tar.gz && sudo mv rules/*.rules /etc/suricata/rules/
sudo chmod 640 /etc/suricata/rules/*.rules
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 4593k 100 4593k    0     0  80562      0  0:00:58  0:00:58  --:--:-- 53658
[sudo] password for zainab:
rules/
rules/3coresec.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
```

```
zainab@zainab-VMware-Virtual-Platform: /tmp
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
rules/emerging-scada.rules
rules/emerging-scan.rules
rules/emerging-shellcode.rules
rules/emerging-smtp.rules
rules/emerging-snmp.rules
rules/emerging-sql.rules
rules/emerging-telnet.rules
rules/emerging-tftp.rules
rules/emerging-user_agents.rules
rules/emerging-voip.rules
rules/emerging-web_client.rules
rules/emerging-web_server.rules
rules/emerging-web_specific_apps.rules
rules/emerging-worm.rules
rules/yyc-2.0.txt
rules/sid-msg.map
rules/suricata-5.0-enhanced-open.txt
rules/threatview_CS_c2.rules
rules/tor.rules
zainab@zainab-VMware-Virtual-Platform:/tmp$
```

## Verifying suricata rules

```
Jan 2 00:28
zainab@zainab-VMware-Virtual-Platform: /etc/suricata/rules
zainab@zainab-VMware-Virtual-Platform:/$ cd /etc/suricata/rules
zainab@zainab-VMware-Virtual-Platform:/etc/suricata/rules$ ls
3coresec.rules          emerging-ftp.rules      emerging-scada.rules
botcc.portgrouped.rules emerging-games.rules    emerging-scan.rules
botcc.rules             emerging-hunting.rules  emerging-shellcode.rules
ciarny.rules            emerging-icmp_info.rules emerging-smtp.rules
compromised.rules       emerging-icmp.rules     emerging-snmp.rules
custom.rules            emerging-imap.rules     emerging-sql.rules
drop.rules              emerging-inappropriate.rules emerging-telnet.rules
dshield.rules           emerging-info.rules     emerging-tftp.rules
emerging-activex.rules  emerging-ja3.rules      emerging-user_agents.rules
emerging-adware_pup.rules emerging-malware.rules  emerging-voip.rules
emerging-attack_response.rules emerging-misc.rules     emerging-web_client.rules
emerging-chat.rules     emerging-mobile_malware.rules emerging-web_server.rules
emerging-coinniner.rules emerging-netbios.rules  emerging-web_specific_apps.rules
emerging-current_events.rules emerging-p2p.rules      emerging-worm.rules
emerging-deleted.rules  emerging-phishing.rules local.rules
emerging-dns.rules       emerging-policy.rules  threatview_CS_c2.rules
emerging-dos.rules       emerging-pop3.rules    tor.rules
emerging-exploit_kit.rules emerging-retired.rules
emerging-exploit.rules   emerging-rpc.rules
zainab@zainab-VMware-Virtual-Platform:/etc/suricata/rules$
```

First copy ip address of Ubuntu through if config

Modify Suricata settings in the /etc/suricata/suricata.yaml file and set the following variables:

Run this command to open file

```
zainab@zainab-VMware-Virtual-Platform:/etc/suricata/rules$ cd /:/:/:/
zainab@zainab-VMware-Virtual-Platform:/$ nano /etc/suricata/suricata.yaml
zainab@zainab-VMware-Virtual-Platform:/$
Floppy Disk
```

Under file do following changes

HOME\_NET: "<UBUNTU\_IP>"

```
Jan 2 00:41
zainab@zainab-VMware-Virtual-Platform: /
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 generated by Suricata 7.0.7.
suricata-version: "7.0"

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

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[redacted]"
    Ubuntu 24.04.1 LTS amd64 [redacted]
    #HOME_NET: "[redacted]"
    #HOME_NET: "any"

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

    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location  ^U Undo Window ^A Set Mark
^X Exit      ^R Read File  ^M Replace   ^U Paste      ^J Justify   ^_ Go To Line  ^E Redo to activate ^G Copyws.
```

because our server will be in public network so we we don't

know from what would be the external network right so will keep it any. So Uncomment it.

EXTERNAL\_NET: "any"

```
EXTERNAL_NET: ["EXTERNAL_NET", "any"]
CDROM "E_NET: "any"

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

HTTP_SERVERS: "$HOME_NET"
SMTP_SERVERS: "$HOME_NET"
```

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

rule-files:

- "\*.rules"

Changes this

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

rule-files:
- /etc/suricata/rules/local.rules
- /etc/suricata/rules/custom.rules

##
## Auxiliary configuration files.
##
classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config
```

To this

```
##

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

rule-files:
- "*.rules"

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

# Global stats configuration

stats:

enabled: Yes

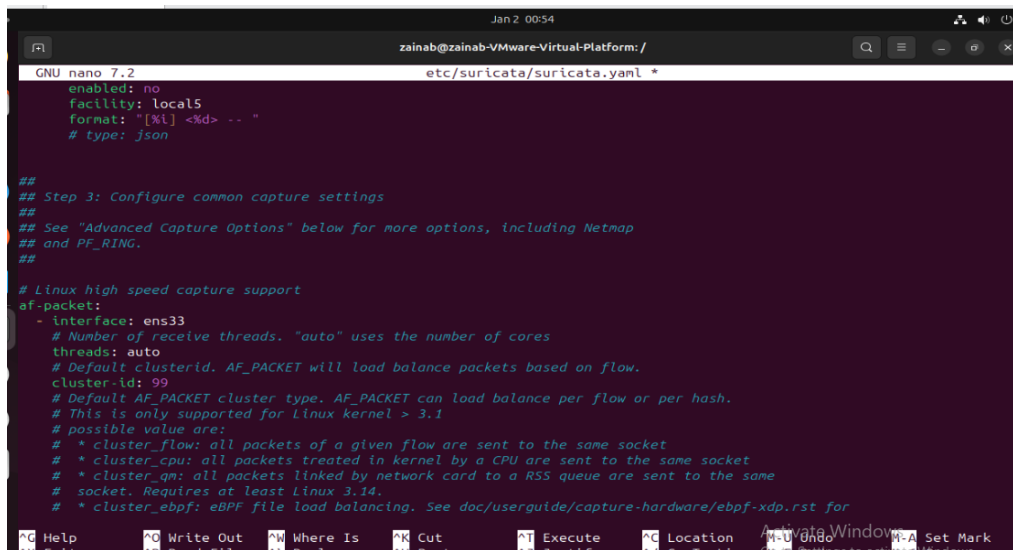
it is default set

```
# Global stats configuration
stats:
  enabled: yes
  # The interval field (in seconds) controls the interval at
  # which stats are updated in the log.
  interval: 8
  # Add decode events to stats.
  decoder-events: true
  # decoder event prefix in stats. Has been 'decoder' before, but that leads
  # to missing events in the eve.stats records. See issue #2225.
  #decoder-events-prefix: "decoder.event"
  # Add stream events as stats.
  #stream-events: false
```

# Linux high speed capture support

af-packet:

- interface: ens33 (change this according to your ip)



```
GNU nano 7.2 etc/suricata/suricata.yaml *
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: ens33
  # 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
```

press ctrl+O then enter to save.

press ctrl+x to exit.

Note: If it deny permission to make changes then use sudo:

sudo nano etc/suricata/suricata.yaml

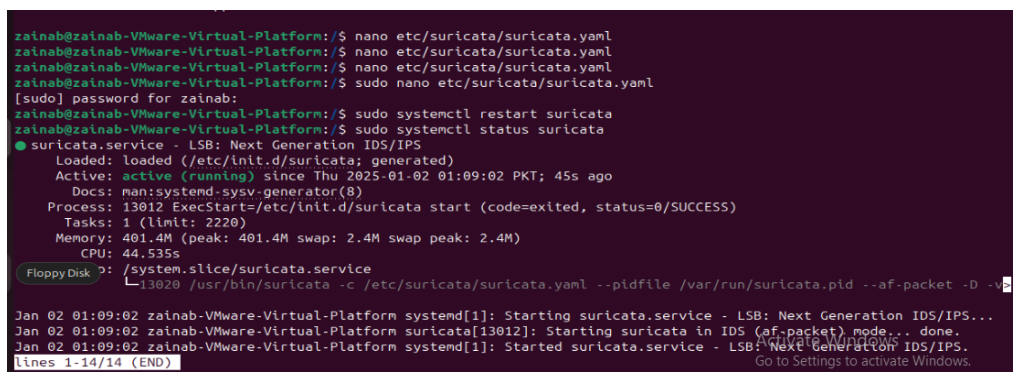
Restart the Suricata service:

sudo systemctl restart suricata



```
zainab@zainab-VMware-Virtual-Platform:~$ nano etc/suricata/suricata.yaml
zainab@zainab-VMware-Virtual-Platform:~$ nano etc/suricata/suricata.yaml
zainab@zainab-VMware-Virtual-Platform:~$ nano etc/suricata/suricata.yaml
zainab@zainab-VMware-Virtual-Platform:~$ sudo nano etc/suricata/suricata.yaml
[sudo] password for zainab:
zainab@zainab-VMware-Virtual-Platform:~$ sudo systemctl restart suricata
zainab@zainab-VMware-Virtual-Platform:~$
```

sudo systemctl status suricata



```
zainab@zainab-VMware-Virtual-Platform:~$ nano etc/suricata/suricata.yaml
zainab@zainab-VMware-Virtual-Platform:~$ nano etc/suricata/suricata.yaml
zainab@zainab-VMware-Virtual-Platform:~$ nano etc/suricata/suricata.yaml
zainab@zainab-VMware-Virtual-Platform:~$ sudo nano etc/suricata/suricata.yaml
[sudo] password for zainab:
zainab@zainab-VMware-Virtual-Platform:~$ sudo systemctl restart suricata
zainab@zainab-VMware-Virtual-Platform:~$ sudo systemctl status suricata
● suricata.service - LSB: Next Generation IDS/IPS
   Loaded: loaded (/etc/init.d/suricata; generated)
   Active: active (running) since Thu 2025-01-02 01:09:02 PKT; 45s ago
     Docs: man:systend-sysv-generator(8)
   Process: 13012 ExecStart=/etc/init.d/suricata start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 2220)
   Memory: 401.4M (peak: 401.4M swap: 2.4M swap peak: 2.4M)
      CPU: 44.535s
   FloppyDisk: /system.slice/suricata.service
             13020 /usr/bin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid --af-packet -D -v
Jan 02 01:09:02 zainab-VMware-Virtual-Platform systemd[1]: Starting suricata.service - LSB: Next Generation IDS/IPS...
Jan 02 01:09:02 zainab-VMware-Virtual-Platform suricata[13012]: Starting suricata in IDS (af-packet) mode... done.
Jan 02 01:09:02 zainab-VMware-Virtual-Platform systemd[1]: Started suricata.service - LSB: Next Generation IDS/IPS.
lines 1-14/14 (END)
```

perfect as see it's active and running so all good ..

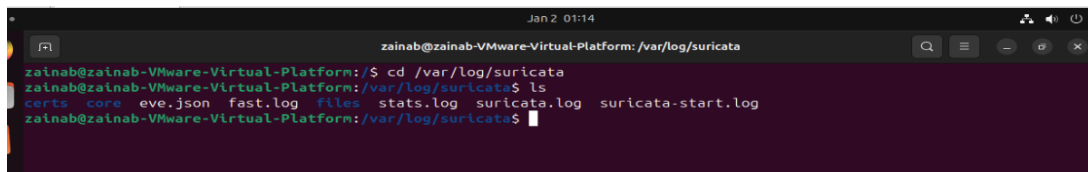


*we can also verify the different logs , that's a better way of*

verifying it as the logs are stored in a different directory.

we can try by

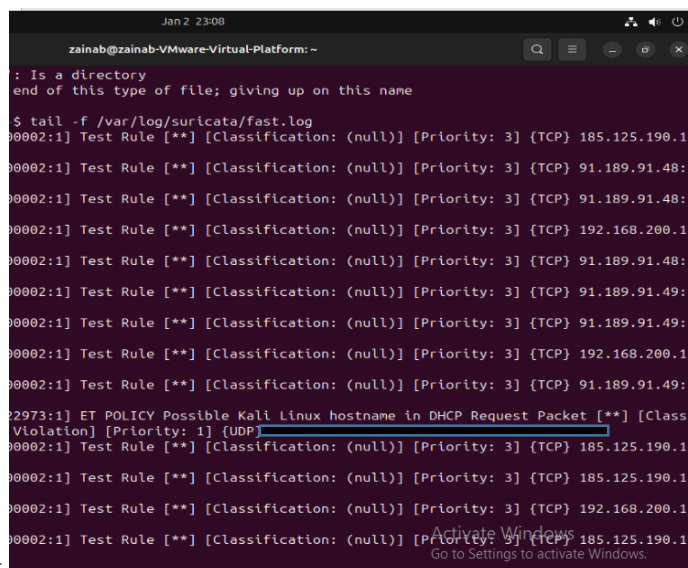
`cd /var/log/suricata`

A terminal window titled 'zainab@zainab-VMware-Virtual-Platform: /var/log/suricata' showing the command 'ls' and its output. The output lists files: 'certs', 'core', 'eve.json', 'fast.log', 'files', 'stats.log', 'suricata.log', and 'suricata-start.log'.

```
zainab@zainab-VMware-Virtual-Platform:/$ cd /var/log/suricata
zainab@zainab-VMware-Virtual-Platform:/var/log/suricata$ ls
certs  core  eve.json  fast.log  files  stats.log  suricata.log  suricata-start.log
zainab@zainab-VMware-Virtual-Platform:/var/log/suricata$
```

View Logs:

Open fast.log for quick alerts:

A terminal window titled 'zainab@zainab-VMware-Virtual-Platform: ~' showing the command 'tail -f /var/log/suricata/fast.log'. The output displays a series of 'Test Rule' messages followed by a 'Possible Kali Linux hostname in DHCP Request Packet' alert. A red box highlights the alert message.

```
zainab@zainab-VMware-Virtual-Platform: ~
: Is a directory
end of this type of file; giving up on this name

$ tail -f /var/log/suricata/fast.log
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 185.125.190.1
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 91.189.91.48:
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 91.189.91.48:
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.200.1
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 91.189.91.48:
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 91.189.91.49:
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 91.189.91.49:
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.200.1
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 91.189.91.49:
02973:1] ET POLICY Possible Kali Linux hostname in DHCP Request Packet [**] [Class
Violation] [Priority: 1] {UDP}
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 185.125.190.1
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 185.125.190.1
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.200.1
00002:1] Test Rule [**] [Classification: (null)] [Priority: 3] {TCP} 185.125.190.1
```

`tail -f /var/log/suricata/fast.log`

For detailed logs, use eve.json:



```

Jan 23 23:08
zainab@zainab-VMware-Virtual-Platform:~
zainab@zainab-VMware-Virtual-Platform:~$ tail -f /var/log/suricata/eve.json
{"timestamp": "2025-01-02T23:07:04.152559+0500", "event_type": "stats", "stats": {"uptime": 343, "capture": {"kernel_packets": 95, "kernel_errors": 0, "send_errors": 0, "afpacket": {"busy_loop_avg": 0, "polls": 4313, "poll_signal": 0, "poll_timeout": 4240, "poll_data": 73, "poll_errors": 0, "send_errors": 0}}, "decoder": {"pkts": 95, "bytes": 13075, "invalid": 0, "ipv4": 53, "ipv6": 2, "ethernet": 95, "arp": 4, "unknown_ethertype": 0, "chdlc": 0, "raw": 0, "null": 0, "sl": 0, "tcp": 41, "udp": 13, "sctp": 0, "esp": 0, "icmpv4": 0, "icmpv6": 1, "ppp": 0, "pppoe": 0, "geneve": 0, "gre": 0, "vlan": 0, "vlan_qinq": 0, "vlan_qinqing": 0, "vxlan": 0, "vntag": 0, "ieee8021ah": 0, "teredo": 0, "ipv4_in_ipv6": 0, "ipv6_in_ipv6": 0, "mpls": 0, "avg_pkt_size": 137, "max_pkt_size": 1514, "max_mac_addr_src": 0, "max_mac_addr_dst": 0, "erspan": 0, "nsh": 0, "event": {"ipv4": {"pkt too small": 0, "hlen too small": 0, "iplen smaller than hlen": 0, "trunc_pkt": 0, "opt_invalid": 0, "opt_invalid_len": 0, "opt_malformed": 0, "opt_pad_required": 0, "opt_eol_required": 0, "opt_duplicate": 0, "opt_unknown": 0, "wrong_ip_version": 0, "icmpv6": 0, "frag_pkt too large": 0, "frag_overlap": 0, "frag_ignored": 0, "icmpv4": {"pkt too small": 0, "unknown_type": 0, "unknown_code": 0, "ipv4_trunc_pkt": 0, "ipv4_unknown_ver": 0, "icmpv6": {"unknown_type": 0, "unknown_code": 0, "pkt too small": 0, "ipv6_unknown_version": 0, "ipv6_trunc_pkt": 0, "mld_message_with_invalid_hl": 0, "unassigned_type": 0, "experimentation_type": 0, "ipv6": {"pkt too small": 0, "trunc_pkt": 0, "trunc_exthdr": 0, "exthdr_dupl_fh": 0, "exthdr_usele_ss_fh": 0, "exthdr_dupl_rh": 0, "exthdr_dupl_hh": 0, "exthdr_dupl_ch": 0, "exthdr_dupl_eh": 0, "exthdr_invalid_optlen": 0, "wrong_ip_version": 0, "exthdr_ah_res_not_null": 0, "hopopts_unknown_opt": 0, "hopopts_unkn_padding": 0, "dstopts_unkn_opt": 0, "dstopts_unkn_padding": 0, "rh_type": 0, "zero_len_padd": 0, "fh_non_zero_reserved_field": 0, "data_after_node_header": 0, "unknown_next_header": 0, "icmpv4": {"frag_pkt too large": 0, "frag_overlap": 0, "frag_invalid_length": 0, "frag_ignored": 0, "ipv4_in_ipv6 too small": 0, "ipv4_in_ipv6_wrong_version": 0, "ipv6_in_ipv6 too small": 0, "ipv6_in_ipv6_wrong_version": 0, "trunc": {"pkt too small": 0, "hlen too small": 0, "invalid_optlen": 0, "opt_invalid_len": 0, "opt_duplicate": 0, "opt_unknown": 0, "pkt too small": 0, "hlen too small": 0, "hlen invalid": 0, "len invalid": 0, "sl": {"pkt too small": 0, "ethernet": {"pkt too small": 0, "ppp": {"pkt too small": 0, "vju_pkt too small": 0, "ip4_pkt too small": 0, "ip6_pkt too small": 0, "wrong_type": 0, "unsupp_proto": 0, "pppoe": {"pkt too small": 0, "wrong_code": 0, "malformed_taps": 0}, "gre": {"pkt too small": 0, "wrong_version": 0, "version8_recur": 0, "version0_flags": 0, "version0_hdr too big": 0, "version0 malformed ser_hdr": 0, "version1 chksum": 0, "version1 route": 0, "version1 ssr": 0, "version1 recur": 0, "version1 flags": 0, "version1 no_key": 0, "version1 wrong_protocol": 0, "version1 malformed_ser_hdr": 0, "version1_hdr too big": 0, "vlan": {"header too small": 0, "unknown_type": 0, "too many layers": 0, "ieee8021ah": {"header too small": 0, "vntag": {"header too small": 0, "unknown_type": 0, "iraw": {"invalid_ip_version": 0, "ltnull": {"pkt too small": 0, "unsupported_type": 0, "sctp": {"pkt too small": 0, "esp": {"pkt too small": 0, "mpls": {"header too small": 0, "pkt too small": 0, "bad_label_router_alert": 0, "bad_label implicit null": 0, "bad_label reserved": 0, "unknown_payload_type": 0, "vxlan": {"unknown_payload_type": 0, "geneve": {"unknown_payload_type": 0, "erspan": {"header too small": 0, "unsupported_version": 0, "too many vlan layers": 0, "dce": {"pkt too small": 0, "chdlc": {"pkt too small": 0, "nsh": {"header too small": 0, "unsupported_version": 0, "bad_header_length": 0, "reserved_type": 0, "unsupported_type": 0, "unknown_payload_type": 0, "too many layers": 0, "tcp": {"syn": 2, "synack": 2, "rst": 0, "urg": 0, "active_sessions": 2, "sessions": 2, "ssn_mempcap_drop": 0, "ssn_from_each_ip": 0, "ssn_from_nontcp": 2, "psuedo": 2, "psuedo_failed": 0, "invalid_checksums": 0, "midstream_pir_kills": 0, "pkt on wrong thread": 0, "ack unseen data

```

Generate network traffic to test its capabilities:

## Ping your Ubuntu system

```
ping <Ubuntu_IP>
```

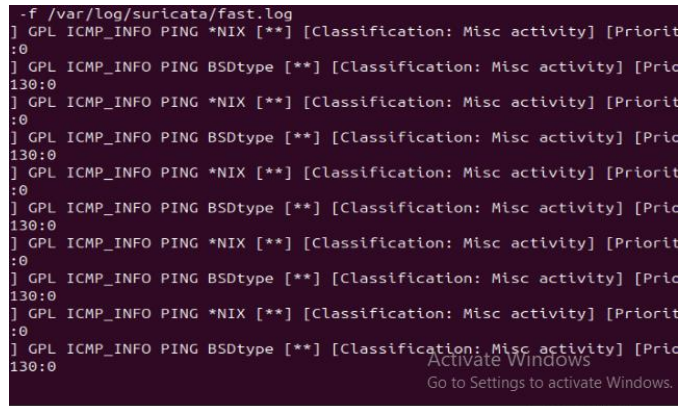
```
msfadmin@metasploitable:~$ ping 192.168.200.130
PING 192.168.200.130: 56(84) bytes of data:
64 bytes from 192.168.200.130: icmp_seq=1 ttl=64 time=7.48 ms
64 bytes from 192.168.200.130: icmp_seq=2 ttl=64 time=1.06 ms
64 bytes from 192.168.200.130: icmp_seq=3 ttl=64 time=0.953 ms
64 bytes from 192.168.200.130: icmp_seq=4 ttl=64 time=0.339 ms
64 bytes from 192.168.200.130: icmp_seq=5 ttl=64 time=1.47 ms
64 bytes from 192.168.200.130: icmp_seq=6 ttl=64 time=1.40 ms
64 bytes from 192.168.200.130: icmp_seq=7 ttl=64 time=1.64 ms
64 bytes from 192.168.200.130: icmp_seq=8 ttl=64 time=1.59 ms
64 bytes from 192.168.200.130: icmp_seq=9 ttl=64 time=1.50 ms
64 bytes from 192.168.200.130: icmp_seq=10 ttl=64 time=0.537 ms
64 bytes from 192.168.200.130: icmp_seq=11 ttl=64 time=1.46 ms
64 bytes from 192.168.200.130: icmp_seq=12 ttl=64 time=0.321 ms
64 bytes from 192.168.200.130: icmp_seq=13 ttl=64 time=1.43 ms
64 bytes from 192.168.200.130: icmp_seq=14 ttl=64 time=1.52 ms

--- ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13007ms
rtt min/avg/max/mdev = 0.321/1.625/7.480/1.684 ms
msfadmin@metasploitable:~$
```

### Check Suricata logs

Run the following command on your Ubuntu system to see the logs:

```
tail -f /var/log/suricata/fast.log
```



```
-f /var/log/suricata/fast.log
] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 0]
] GPL ICMP_INFO PING BSDtype [**] [Classification: Misc activity] [Priority: 130:0]
] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 0]
] GPL ICMP_INFO PING BSDtype [**] [Classification: Misc activity] [Priority: 130:0]
] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 0]
] GPL ICMP_INFO PING BSDtype [**] [Classification: Misc activity] [Priority: 130:0]
] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 0]
] GPL ICMP_INFO PING BSDtype [**] [Classification: Misc activity] [Priority: 130:0]
] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 0]
] GPL ICMP_INFO PING BSDtype [**] [Classification: Misc activity] [Priority: 130:0]
```

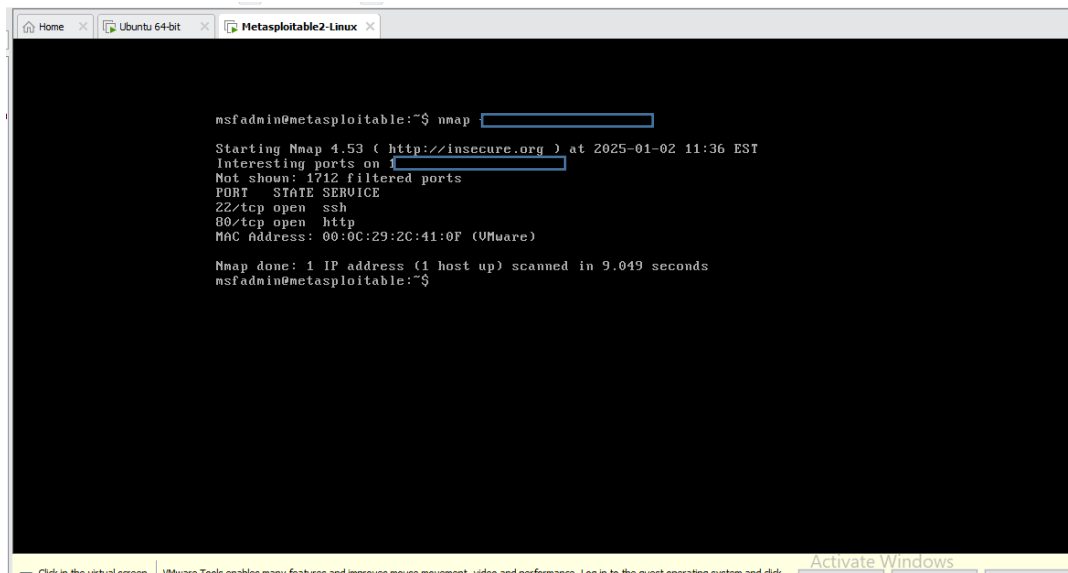
Look for entries related to ICMP traffic.

### Port Scan

#### Run an nmap scan

From another device, use **nmap** to perform a SYN scan on your Ubuntu machine: `bash`

```
nmap -sS <Ubuntu_IP>
```



```
msfadmin@metasploitable:~$ nmap [redacted]
Starting Nmap 4.53 ( http://insecure.org ) at 2025-01-02 11:36 EST
Interesting ports on [redacted]:
Not shown: 1712 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:0C:29:2C:41:0F (VMware)

Nmap done: 1 IP address (1 host up) scanned in 9.049 seconds
msfadmin@metasploitable:~$
```

### Check Suricata logs

On your Ubuntu system, monitor the logs again for port scan alerts:

```
tail -f /var/log/suricata/fast.log
```

You should see alerts related to the port scanning activity.

# What did you learn about Suricata rules?

## Custom Rules for Threat Detection:

**Objective:** This demonstrates Suricata's flexibility, allowing us to detect specific attack patterns such as port scans, suspicious network traffic, or even application layer vulnerabilities.

Learn the structure of Suricata rules:

Action (e.g., alert)

Protocol (e.g., TCP, UDP)

Source IP, destination IP

Source port, destination port

Payload conditions (patterns in the data)

Steps to Implement Custom Rules

### Step 1: Understand Suricata's Rule Syntax:

Suricata uses a simple and intuitive rule syntax to identify malicious or suspicious traffic patterns. These rules help detect everything from basic port scans to more complex attack vectors. The first step in our project is to familiarize ourselves with this syntax to create custom rules for detecting specific threats.

### Step 2: Identify a Threat:

Next, we'll identify a network threat that we want Suricata to detect. This could be something simple, like a port scan

### Step 3: Create a Custom Suricata Rule:

With Suricata, we can write rules to detect specific types of traffic. For detecting a port scan, we would write a rule that looks for an unusually high number of connection attempts to different ports in a short time frame. This rule would alert Suricata whenever this pattern is detected."

Basic structure of a Suricata rule:

```
alert <protocol> <source_ip> <source_port> -> <destination_ip> <destination_port>  
(msg:"<alert_message>"; <options>; sid:<unique_id>;)
```

### Step 4: Working

Develop custom Suricata rules to detect specific threats (e.g., port scans, suspicious traffic, or unauthorized access attempts) and test their effectiveness using a simulated environment.

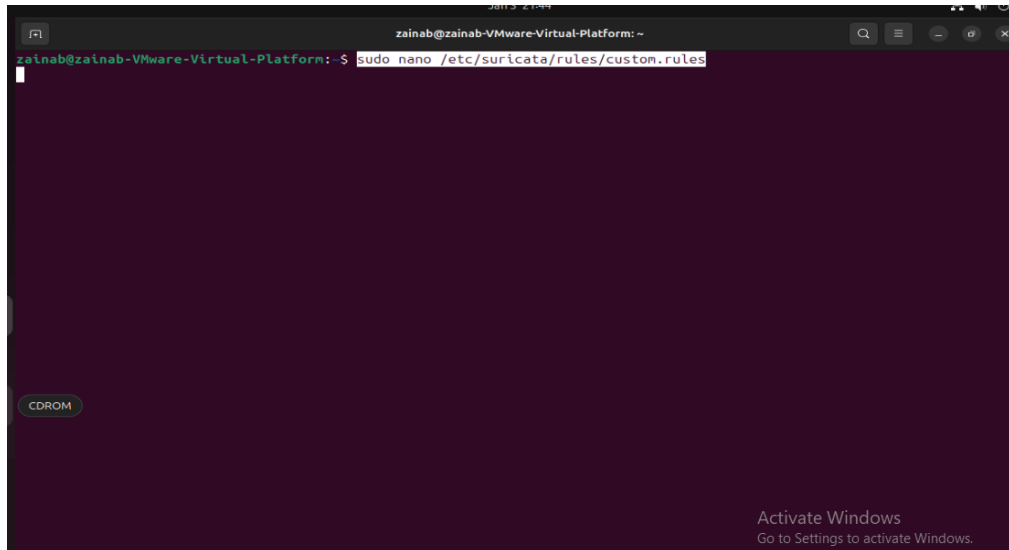
Test the Rules

Add the Rules to Suricata:

Create a new custom rules file:

```
sudo nano /etc/suricata/rules/custom.rules
```

Add the rules to this file.



□ Example 1: Detect Port Scans:

```
alert tcp any any -> $HOME_NET any (msg:"Possible Port Scan"; flags:S; threshold:type both, track by_dst, count 10, seconds 5; sid:100001;)
```



Update Suricata Configuration:

Edit the Suricata configuration file:

```
sudo nano /etc/suricata/suricata.yaml
```

```
Jan 3 21:51
zainab@zainab-VMware-Virtual-Platform: ~
zainab@zainab-VMware-Virtual-Platform:~$ sudo nano /etc/suricata/rules/custom.rules
[sudo] password for zainab:
zainab@zainab-VMware-Virtual-Platform:~$ sudo nano /etc/suricata/rules/custom.rules
zainab@zainab-VMware-Virtual-Platform:~$ sudo nano /etc/suricata/suricata.yaml
```

Ensure the custom rules file is included:

rule-files:

- custom.rules

```
ntu 64-bit
Jan 4 00:34
zainab@zainab-VMware-Virtual-Platform: ~
GNU nano 7.2 /etc/suricata/suricata.yaml
# See Nmaptech NTPL documentation other hashmodes and details on their use.
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hashStuplesorted

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

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

rule-files:
- "custom.rules"

## Auxiliary configuration files.
##

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

##
## Include other configs
##

# Includes: Files included here will be handled as if they were in-l
```

set address-groups as below:

```
CNU 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 generated by Suricata 7.0.7.
suricata-version: "7.0"

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

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: [REDACTED]
    #HOME_NET: [REDACTED]
    #HOME_NET: [REDACTED]
    CDROM "FE_NET: "any"
    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"
    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"

[Help] [Exit] [Write Out] [Read File] [Where Is] [Replace] [Cut] [Paste] [Read 2189 lines] [Execute] [Justify] [Location] [Go To Line] [Undo] [Redo] [Activate Window] [Set Mark] [Copy] [Paste]
```

Restart Suricata:

sudo systemctl restart suricata

```
Jan 3 21:59
zainab@zainab-VMware-Virtual-Platform: ~
zainab@zainab-VMware-Virtual-Platform:~$ sudo nano /etc/suricata/rules/custom.rules
[sudo] password for zainab:
zainab@zainab-VMware-Virtual-Platform:~$ sudo nano /etc/suricata/rules/custom.rules
zainab@zainab-VMware-Virtual-Platform:~$ sudo nano /etc/suricata/suricata.yaml
zainab@zainab-VMware-Virtual-Platform:~$ sudo nano /etc/suricata/suricata.yaml
zainab@zainab-VMware-Virtual-Platform:~$ sudo systemctl restart suricata
```

Simulate Attacks

On Metasploitable, scan port

Nmap Port Scan:

nmap -p 1-100 192.168.x.x

```
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ nmap 192.168.200.1

Starting Nmap 4.53 ( http://insecure.org ) at 2025-01-03 12:02 EST
Interesting ports on 192.168.200.1:
Not shown: 1712 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:0C:29:2C:41:0F (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.608 seconds
msfadmin@metasploitable:~$ nmap -p 1-100 192.168.200.100

Starting Nmap 4.53 ( http://insecure.org ) at 2025-01-03 12:07 EST
Interesting ports on 192.168.200.100:
Not shown: 98 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:0C:29:2C:41:0F (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.391 seconds
msfadmin@metasploitable:~$ _
```

## Analyze Logs

View alerts:

```
tail -f /var/log/suricata/fast.log
```

Check for matches with your custom rules:

Alerts for port scans should include:

text

```
[**] [1:100001:1] Possible Port Scan [**]
```

*How to Match the Expected Port Scan Alert:*

☐ Custom Rule Match:

The rule with sid:100001 is matching and generating alerts for possible port scans:

```
[1:100001:0] Possible Port Scan
```

This indicates that traffic matching the flags:S and threshold conditions is being detected as a port scan.

☐ Alerts in Logs:

These lines from your logs confirm the rule is firing:

```
01/03/2025-22:06:55.167636 [**] [1:100001:0] Possible Port Scan [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.200.1:49309 -> 192.168.200.100:95
```

☐ Other Rules:

The logs also show other rules matching, such as the ET SCAN rules:

```
[1:2010937:3] ET SCAN Suspicious inbound to mySQL port 3306
```



### **Conclusion:**

By completing these steps, created a custom Suricata rule, tested it using nmap to generate a port scan, and monitored Suricata's logs for the detection. This is a simple demonstration of how flexible Suricata is in detecting specific threats using custom rules.

### **Conclusion:**

By following these steps can:

1. Set up Suricata as an IDS/IPS.
2. Create and test custom rules.
3. Analyze logs for threat detection.

This demonstrates the flexibility and power of Suricata for real-time threat detection and network security.