(3) Rules to alert & block traffic

In the previous section with multi-detect we defined location of configurations with rules for each of the namespaces. Tenant configuration file will contain the location of the rules which will be executed by suricata, they would look like this
Net1.yaml

```
%YAML 1.1
---
rule-files:
  - /home/test/rules/set-A.rules
```

Net2.yaml

```
%YAML 1.1
---
rule-files:
  - /home/test/rules/set-B.rules
```

Net3.yaml

```
%YAML 1.1
---
rule-files:
  - /home/test/rules/set-C.rules
```

Net4.yaml

```
%YAML 1.1
---
rule-files:
  - /home/test/rules/set-D.rules
```

Net5.yaml

```
%YAML 1.1
---
rule-files:
  - /home/test/rules/set-E.rules
```

Now all these point to rule sets which look like this

set-A.rules

```
alert tcp 10.0.1.2 any -> 10.0.6.2 22 (msg:"Net1: SSH Connection Attempt";
sid:1000001; rev:1;)
```

set-B.rules

```
alert http 10.0.2.2 any -> 10.0.6.2 80 (msg:"Net2: HTTP Web Access";
sid:1000002; rev:1;)
```

set-C.rules

```
alert tcp 10.0.3.2 any -> 10.0.6.2 21 (msg:"Net3: FTP Login Attempt";
sid:1000003; rev:1;)
```

set-D.rules

```
alert tcp 10.0.4.2 any -> 10.0.6.2 445 (msg:"Net4: SMB/Samba Traffic
Detected"; sid:1000004; rev:1;)
```

set-E.rules

```
alert icmp 10.0.5.2 any -> 10.0.6.2 any (msg:"Net5: ICMP Ping Detected";
sid:1000005; rev:1;)
```

As this is a demonstration of the rules creating alerts the commands used to trigger the alert and the alert itself are simple to understand the fundamental concept.

Triggering Alerts

First we will start suricata to monitor traffic using

```
sudo ip netns exec Suricata suricata -c /home/test/suricata.yaml --af-packet
```

This starts up suricata, it listens on the pre-configured interfaces for any network traffic. Next we have to start a listener in namespace N6 which acts as a service listening for incoming requests, for which Netcat will be used.

Alert 1 : SSH

```
sudo ip netns exec N6 nc -nvlp 22
```

Now that a listener has started in N6 namespace we now start an ssh request from N1 namespace using

```
sudo ip netns exec N1 ssh 10.0.6.2
```

Alert 2 : HTTP

```
sudo ip netns exec N6 nc -nvlp 80
```

http request from namespace N2 using

```
sudo ip netns exec N2 curl http://10.0.6.2
```

Alert 3 : FTP

```
sudo ip netns exec N6 nc -nvlp 21
```

FTP request from N3

```
sudo ip netns exec N3 nc -zv 21
```

Alert 4 : SMB

```
sudo ip netns exec N6 nc -nvlp 445
```

SMB request from namespace N4

```
sudo ip netns exec N4 nc -zv 445
```

Alert 5 : ICMP ping

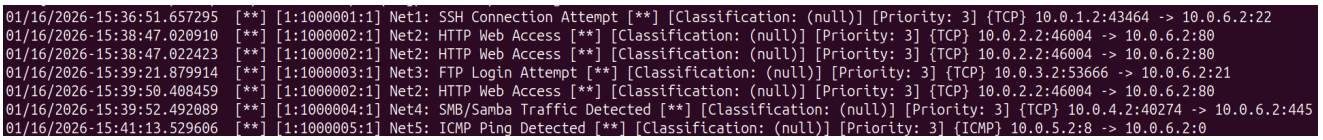
For this no listener is required as firewall in Namespace N6 is not configured to block pings so its open to ping

ping request from namespace N5

```
sudo ip netns exec N5 ping 10.0.6.2
```

Logs

When we check the logs located in the directory `/var/log/suricata`, first we "cat" the logs as it is the easiest to read and needs no pre-processing which is the "fast.log". A screenshot of the alerts detected captured in fast.log looks like this -



A copy of the file will also be attached

The other log source which is also created is eve.json , this type of file can either be read by using any SIEM tool to read it or use "jq" to display the contents of the file. This type of file required specific queries to be readable. The jq command used in this case is

```
cat /var/log/suricata/eve.json | jq -r 'select(.event_type=="alert")'
```

To show the result of this command we can use "tail" to show a part of it which looks like this


```
{
  "timestamp": "2026-01-16T15:41:13.529606+0530",
  "flow_id": 304317710694226,
  "in_iface": "veth5-su",
  "event_type": "alert",
  "src_ip": "10.0.5.2",
  "dest_ip": "10.0.6.2",
  "proto": "ICMP",
  "ip_v": 4,
  "icmp_type": 8,
  "icmp_code": 0,
  "pkt_src": "wire/pcap",
  "tenant_id": 5,
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 1000005,
    "rev": 1,
    "signature": "Net5: ICMP Ping Detected",
    "category": "",
    "severity": 3,
    "tenant_id": 5
  },
  "direction": "to_server",
  "flow": {
    "pkts_toserver": 1,
    "pkts_toclient": 0,
    "bytes_toserver": 98,
    "bytes_toclient": 0,
    "start": "2026-01-16T15:41:13.529606+0530",
    "src_ip": "10.0.5.2",
    "dest_ip": "10.0.6.2"
  }
}
```