Reference: ASSR-2023-CB-Suricata

Master Cryptis 2023
ASSR

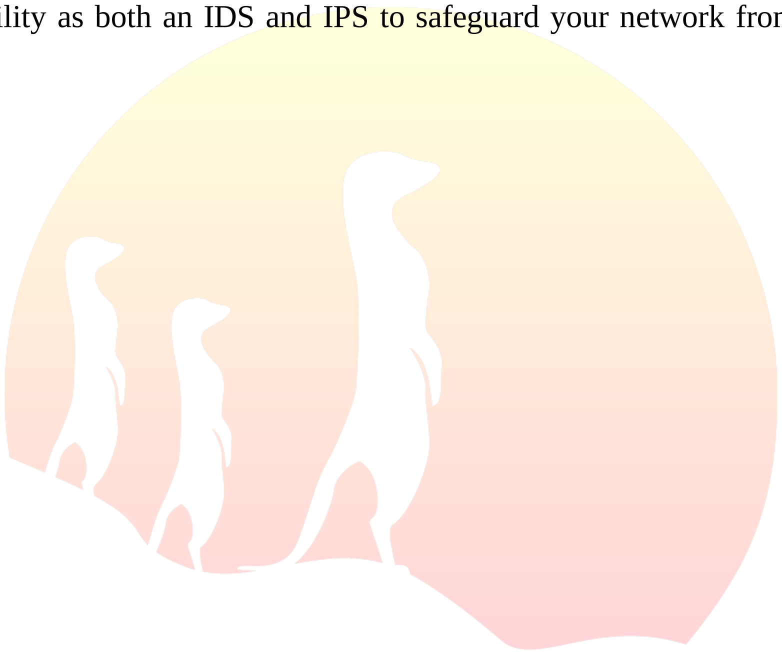
Université de Limoges
NAME : Salame Joe

Suricata Cookbook

Summary

The cyber security landscape is constantly evolving and the need for Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) has become crucial to offer real-time threat detection and proactive defense against different cyber threats.

This is within this Cookbook, we will give you an in-depth guide to the robust security open-source network Intrusion Detection and Prevention System, Suricata. We will uncover Suricata's distinctive features and advantages that strengthen defense against various cyber threats. Furthermore, we will discover how this high-performance threat detection engine excels from real time analysis to active threat recognition. The Cookbook concludes with a user-friendly, step by step implementation guide, guaranteeing a seamless deployment on a Linux environment. Finally, validate your cybersecurity strategy with Suricata through a straightforward yet impactful test to confirm its capability as both an IDS and IPS to safeguard your network from a wide range of threats.



General Presentation:

In today’s interconnected world, network security becomes a critical need. The increasing complexity of sophisticated cyber threats requires a proactive and robust defense mechanism. This is where the implementation of security systems such as Intrusion Detection Systems and Intrusion Prevention Systems become a crucial necessity.

The evolution of Intrusion Detection Systems (IDS) has been remarkable. Originating in the 1980s, IDS progressed from rule-based systems centered on auditing system logs into real-time threat detection in the 1990s with the introduction of network-based intrusion detection systems (NIDS). This era saw also the emergence of Intrusion Prevention Systems (IPS) in the 2000s marking a proactive shift in cybersecurity, actively preventing threats.

Today, IDS has become an integral layer of defense for organizations, serving as an early warning system that enables cybersecurity professionals to detect and respond to cyber threats in a proactive and timely manner. While IPS serve as a proactive shield actively preventing malicious activities.

Suricata stands at the front. It is an open-source Intrusion Detection System and Intrusion Prevention System that provides real-time monitoring and analysis of network traffic. It is a vital tool designed to detect and respond to various types of cyber threats and attacks such as malware, network intrusions, denial of service attacks and data breaches in order to safeguard the integrity and confidentiality of your network assets.

This dynamic low-cost tool emerges among security frameworks and empowers cybersecurity professionals to safeguard networks against evolving cyber threats. It employs signature detection, behavioral analysis and community-driven updates to actively identify and respond to potential security threats. It can be deployed in various environments including on-premises and in the cloud.

It supports various capture modes which allow it to analyze traffic in different ways. The main ones are AF_Packet, PCAP (Packet Capture), NFQ (Netfilter Queue), IPFW (IP Firewall), Netmap and PFRING (Packet Filtering).

Suricata offers a powerful solution that is far more than IDS/IPS, it has other functionalities like Network Traffic Analysis, Protocol Analysis, Traffic Pattern Detection, Log Generation, Reporting, File Extraction and Analysis.

Definitions:

Suricata is an open-source Intrusion Detection system and Intrusion Prevention system developed in 2009 by the Open Information Security Foundation (OISF) a non-profit foundation organized to build a next generation IDS/IPS engine. Leveraging a high performance, multithreaded architecture, Suricata enables the active identification and prevention of diverse cyber threats. It works through a combination of signature-based detection, protocol and behavioral analysis to identify and respond to potential threats.

Here is some key terms and terminology you need to know:

1. **Intrusion Detection System (IDS)**:
   - A security mechanism designed to monitor and analyze network and system activities for signs of malicious activities.
2. **Intrusion Prevention System (IPS)**:
   - A security solution that detects and actively prevents potential security threats.
3. **Signature-Based Detection**:
   - A method of identifying known patterns or signatures of malicious activity based on predefined rules.
4. **Behavioral analysis**:
   - The process of monitoring and analyzing the behavior of network traffic to detect anomalies.
5. **AF_PACKET, PCAP, NFQ**:
   - Different modes of packet capture:
     i. **AF_PACKET:** Advanced Packet Socket, for copying the packets from one interface to the other. It provides High-performance capturing capabilities.
     ii. **PCAP:** Packet Capture, for intercepting and log data traffic that crossing specific point in data network.
     iii. **NFQ:** Netfilter Queue, for receiving packet from the network stack and then inspecting the packet for potential threats before allowing or blocking the packets.
6. **Multithreaded Architecture**:
   - Architectural design that allows Suricata to execute multiple threads concurrently.

Hardware and Software Prerequisites:

In order to successfully implement and test Suricata tool on a host, please ensure you have the following Hardware and Software ingredients:

Hardware Prerequisites:

1. **Server/Computer:** We will use a VirtualBox VM
2. **Storage Spaces:** At least 30GB of disk space are required for your VM.
3. **CPU:** At least 2 CPU are required for our testing environment.
4. **RAM:** At least 4-8 GB of memory.

Software Prerequisites:

1. **Linux OS:** We will use Ubuntu 22.04.3 LTS as our OS.
2. **Suricata Software:** Latest stable version 7.0.2.
3. **Text Editor:** Ensure you have nano, vi or vim on your machine for editing the configuration.
4. **Package management Tools:** APT, typically installed by default on Ubuntu. It will simplify the process of installing required software on our Linux system.

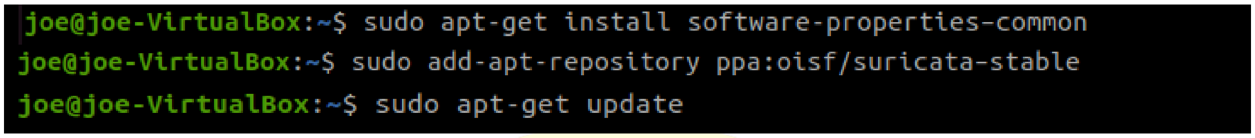
Please note that the above requirements may vary from the Suricata deployment option whether it is an HIDS (Host Based - IDS) or NIDS (Network Based – IDS) and from your production environment.

Illustrative Recipe

Step 1- Installing Suricata

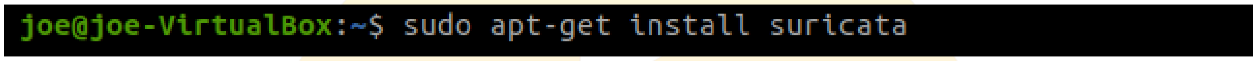
This Section provides step-by-step instructions for installing Suricata on an Ubuntu machine.

To get started, you need to add the OISF software repository information to your Ubuntu system. Run the following commands to add the repository to your system and update the list of available packages:



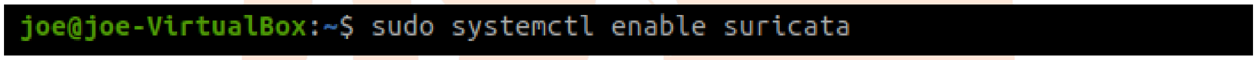
```
joe@joe-VirtualBox:~$ sudo apt-get install software-properties-common
joe@joe-VirtualBox:~$ sudo add-apt-repository ppa:oisf/suricata-stable
joe@joe-VirtualBox:~$ sudo apt-get update
```

Now you can install the latest stable version of Suricata package using ***apt*** command:



```
joe@joe-VirtualBox:~$ sudo apt-get install suricata
```

Since the package is installed now, run the following command to enable the ***suricata.service*** so it will run when your system restart:



```
joe@joe-VirtualBox:~$ sudo systemctl enable suricata
```

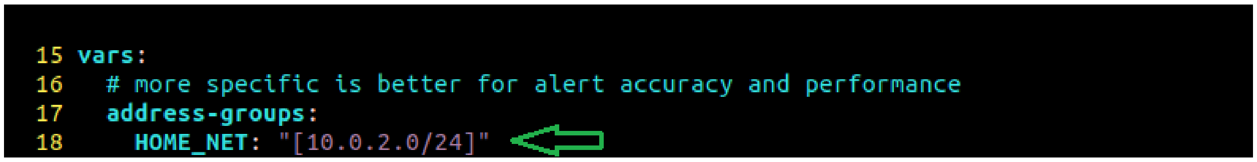
Step 2- Suricata Deployment

In this section, we will cover some configuration file modifications that need to be done in order to be able to test Suricate's IDS and IPS functionalities.
The default mode for Suricata is IDS mode, so no traffic is dropped, it is only logged.
Please note that the following modifications have been made to our environment based on our testing needs.

<u>Configuration file modifications</u>

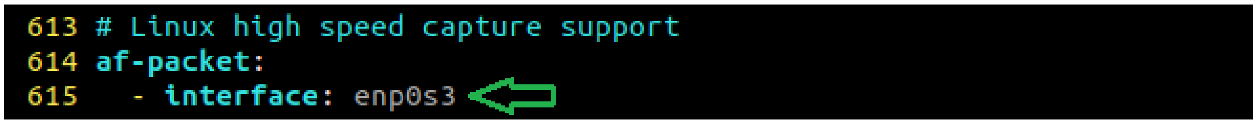
In order to increase the IDS and IPS performance the following changesets in Suricata's configuration file ***/etc/suricata/suricata.yaml*** must be done:

1. Put the subnet address or the IP address of the Suricata's machine in the ***HOME_NET*** variable within the configuration file ***/etc/suricata/suricata.yaml***. In our case we used the subnet address ***10.0.2.0/24***. To do so, open the above configuration file using ***vi*** or your preferred editor and change the value:



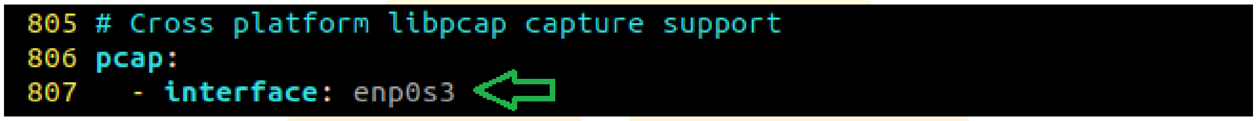
```
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[10.0.2.0/24]"
```

2. Specify the network interface name that you would like Suricata to inspect traffic on in the ***af-packet*** attribute. As we mentioned, this ensures high-performance packet capturing and processing capabilities, thereby enhancing the efficiency of packet analysis:



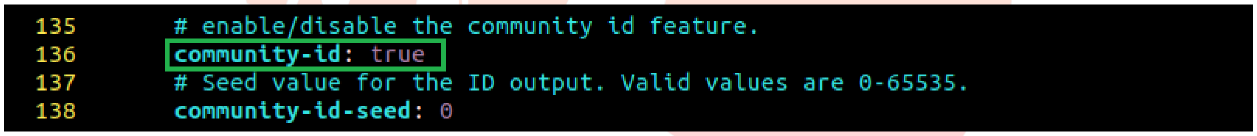
```
# Linux high speed capture support
af-packet:
  - interface: enp0s3
```

3. Keep your editor open and add the interface name in the ***pcap*** attribute to apply control over memory:



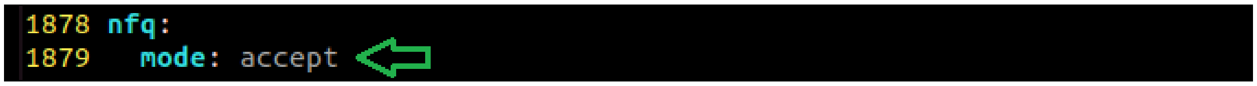
```
# Cross platform libpcap capture support
pcap:
  - interface: enp0s3
```

4. We also need to enable the ***community-id*** in the **Community Flow ID** section. This will add a ***community_id*** to EVE records, ensuring a predictable flow ID. This ID is important for correlating records across various tools (like Zeek or Elasticsearch), offering a more comprehensive understanding of network events:



```
      # enable/disable the community id feature.
      community-id: true
      # Seed value for the ID output. Valid values are 0-65535.
      community-id-seed: 0
```

5. Finally, to use, later on, in our testing Suricata as IPS we need to enable the **NFQ** mode. Uncomment the ***mode*** attribute in ***nfq*** section and set it to ***accept***:



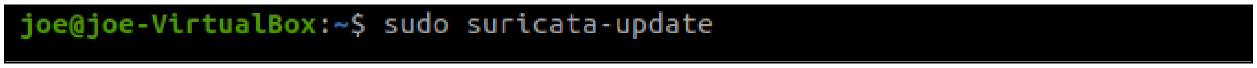
```
nfq:
  mode: accept
```

We have 3 modes in **NFQ**:

- **accept**: It's the default mode where the packet will be accepted or dropped by Suricata after being inspected.
- **repeat:** The packets will be marked by Suricata and then re-injected at the first rule of iptables.
- **route:** The packet will be sent to another tool after being processed by Suricata.

Step 3- Updating Suricata Rulesets

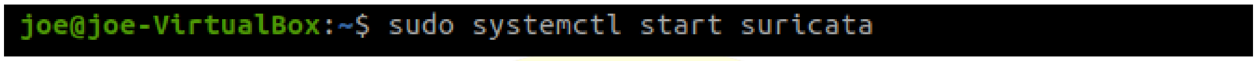
Suricata includes a tool called ***suricata-update*** that will fetch rulesets from the Suricata project's rule repository and load them in ***/var/lib/suricata/rules/***.
Run the command as below to load the latest set of signatures:



```
joe@joe-VirtualBox:~$ sudo suricata-update
```

Step 4- Running Suricata

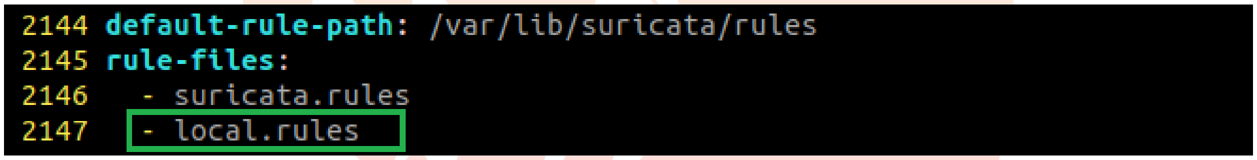
After loading the rulesets, you can start Suricata by running the following command:



```
joe@joe-VirtualBox:~$ sudo systemctl start suricata
```

Step 5- Create custom rule

Users can also write their own custom rules. To achieve this, create a file named ***local.rules*** under ***/var/lib/suricata/rules***. This file will be used afterwards in the testing section. Please note the name of the file can be random but the extension must be **.rules**. Proceed by adding the file name in the ***/etc/suricata/suricata.yaml*** configuration file as follow:



```
default-rule-path: /var/lib/suricata/rules
rule-files:
  - suricata.rules
  - local.rules
```

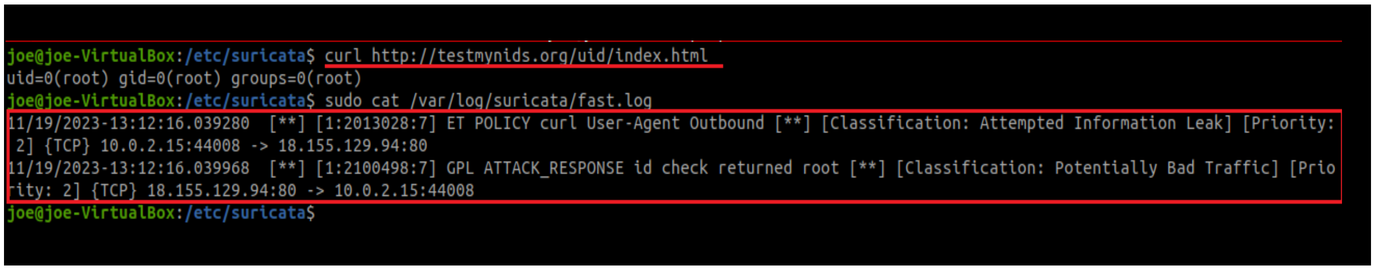
N.B: In case the custom rules file was created in a different location than the **default-rule-path**, the full file path name must be provided **(/path-to-the-file/local.rules**).

Step 6- Minimal testing

Now, we will proceed to perform comprehensive tests on both the IDS and IPS functionalities to validate their proper functioning in detecting and preventing potential security threats.

IDS signatures test

To verify the effectiveness of the IDS signatures and ensure they are operating as expected, we will send a GET request using ***curl*** tool to http://testmynids.org/uid/index.html, a designated website designed for IDS testing.
The response from this website should trigger alerts that will be recorded in **/var/log/suricata/fast.log**:



```
joe@joe-VirtualBox:/etc/suricata$ curl http://testmynids.org/uid/index.html
uid=0(root) gid=0(root) groups=0(root)
joe@joe-VirtualBox:/etc/suricata$ sudo cat /var/log/suricata/fast.log
11/19/2023-13:12:16.039280  [**] [1:2013028:7] ET POLICY curl User-Agent Outbound [**] [Classification: Attempted Information Leak] [Priority:
 2] {TCP} 10.0.2.15:44008 -> 18.155.129.94:80
11/19/2023-13:12:16.039968  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentially Bad Traffic] [Prio
rity: 2] {TCP} 18.155.129.94:80 -> 10.0.2.15:44008
joe@joe-VirtualBox:/etc/suricata$
```

As noticed in the above capture two alerts were detected: **ET POLICY curl User-Agent Outbound** and **GPL ATTACK_RESPONSE ID check returned root**.

IDS custom rule test

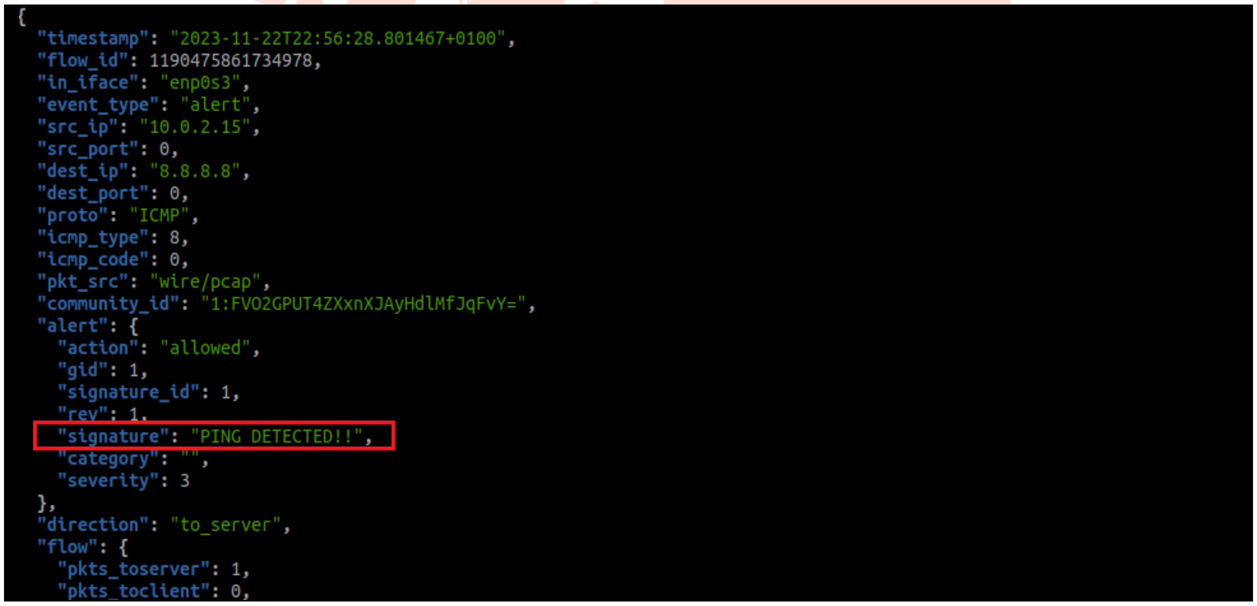
In this test, add the below rule in the ***local.rules*** file created in **Step 5**. This rule detects pings executed from the ***HOME_NET*** to any external address to confirm the proper IDS functionality:

```
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"PING DETECTED!!"; sid:1; rev:1;)
```

Open a second terminal, connect to your server and run the below command before proceeding:

```
joe@joe-VirtualBox:~$ sudo tail -f /var/log/suricata/eve.json | jq 'select(.event_type=="alert")'
```

Return to your first terminal and execute a ***ping*** to **8.8.8.8.** Immediately the ping alert will be detected on your second terminal where we opened the **eve.json** log file as below:



```
{
  "timestamp": "2023-11-22T22:56:28.801467+0100",
  "flow_id": 1190475861734978,
  "in_iface": "enp0s3",
  "event_type": "alert",
  "src_ip": "10.0.2.15",
  "src_port": 0,
  "dest_ip": "8.8.8.8",
  "dest_port": 0,
  "proto": "ICMP",
  "icmp_type": 8,
  "icmp_code": 0,
  "pkt_src": "wire/pcap",
  "community_id": "1:FVO2GPUT4ZXxnXJAyHdlMfJqFvY=",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 1,
    "rev": 1,
    "signature": "PING DETECTED!!",
    "category": "",
    "severity": 3
  },
  "direction": "to_server",
  "flow": {
    "pkts_toserver": 1,
    "pkts_toclient": 0,
```

IPS functionality test

Now, in this section we are going to test Suricata as IPS. As stated earlier, NFQ mode must be enabled. This was done in **Step2 - Configuration file modifications (5)**.

Add the below rule in the ***local.rules*** file created in **Step 5**. The rule detects and blocks pings executed from the **HOME_NET** to any external address:

```

drop icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"ICMP detected and blocked!!"; sid:1; rev:1;)
```

To send traffic to Suricata for inspection add the below rules in Iptables:

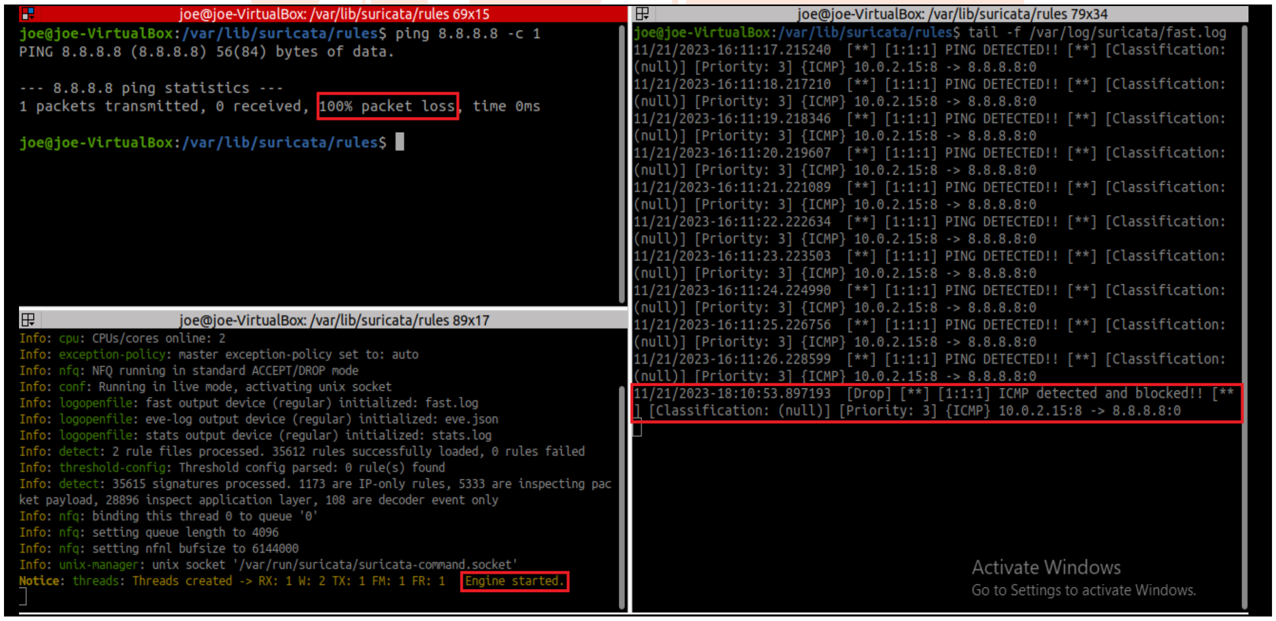
```
joe@joe-VirtualBox:~$ sudo iptables -I INPUT -j NFQUEUE
joe@joe-VirtualBox:~$ sudo iptables -I OUTPUT -j NFQUEUE
```

N.B: Please note this is configuration is for host installation. If you use Suricata on a gateway run the following command instead: **sudo iptables -I FORWARD -j NFQUEUE**.

Now run Suricata as IPS and start the engine as follow:

```
joe@joe-VirtualBox:~$ sudo suricata -c /etc/suricata/suricata.yaml -q 0 -v
```

Once started, open another terminal as below to tail the **fast.log** and then execute a **ping** to **8.8.8.8**:


joe@joe-VirtualBox: /var/lib/suricata/rules 69x15
joe@joe-VirtualBox: /var/lib/suricata/rules 89x17
joe@joe-VirtualBox: /var/lib/suricata/rules 79x34
Activate Windows
Go to Settings to activate Windows.

```
joe@joe-VirtualBox:/var/lib/suricata/rules$ ping 8.8.8.8 -c 1
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

joe@joe-VirtualBox:/var/lib/suricata/rules$
```

```
Info: cpu: CPUs/cores online: 2
Info: exception-policy: master exception-policy set to: auto
Info: nfq: NFQ running in standard ACCEPT/DROP mode
Info: conf: Running in live mode, activating unix socket
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. 35612 rules successfully loaded, 0 rules failed
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 35615 signatures processed. 1173 are IP-only rules, 5333 are inspecting pac
ket payload, 28896 inspect application layer, 108 are decoder event only
Info: nfq: binding this thread 0 to queue '0'
Info: nfq: setting queue length to 4096
Info: nfq: setting nfnl bufsize to 6144000
Info: unix-manager: unix socket '/var/run/suricata/suricata-command.socket'
Notice: threads: Threads created -> RX: 1 W: 2 TX: 1 FM: 1 FR: 1  Engine started.
```

```
joe@joe-VirtualBox:/var/lib/suricata/rules$ tail -f /var/log/suricata/fast.log
11/21/2023-16:11:17.215240  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-16:11:18.217210  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-16:11:19.218346  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-16:11:20.219607  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-16:11:21.221089  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-16:11:22.222634  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-16:11:23.223503  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-16:11:24.224990  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-16:11:25.226756  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-16:11:26.228599  [**] [1:1:1] PING DETECTED!! [**] [Classification:
(null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
11/21/2023-18:10:53.897193  [Drop] [**] [1:1:1] ICMP detected and blocked!! [**
] [Classification: (null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 8.8.8.8:0
```

As noticed, after executing the ping, an alert was displayed on the right side of the capture indicating that a ping was detected and blocked. Furthermore, the packet loss is 100% indicating that the ICMP packet was blocked. This means Suricata detected and blocked the traffic.

References/Documentations:

1. The Evolution of Intrusion Detection Systems: A Historical Overview: https://ts2.space/en/the-evolution-of-intrusion-detection-systems-a-historical-overview/#gsc.tab=0
2. Open Information Security Foundation (OISF). Suricata: Open-Source IDS/IPS Engine: https://suricata.io/
3. Introduction To Suricata IDS- HackerSploit: https://www.youtube.com/watch?v=91i7InHVOso
4. Suricata basic Functionality: https://medium.com/@kennithlowy/suricata-basic-functionality-868135dac7cd
5. Suricata Installation Guide: https://docs.suricata.io/en/latest/quickstart.html#installation
6. Suricata Overview: https://www.rapid7.com/blog/post/2017/02/21/suricata-overview/
7. Installing & Configuring Suricata- HackerSploit: https://www.youtube.com/watch?v=UXKbh0jPPpg