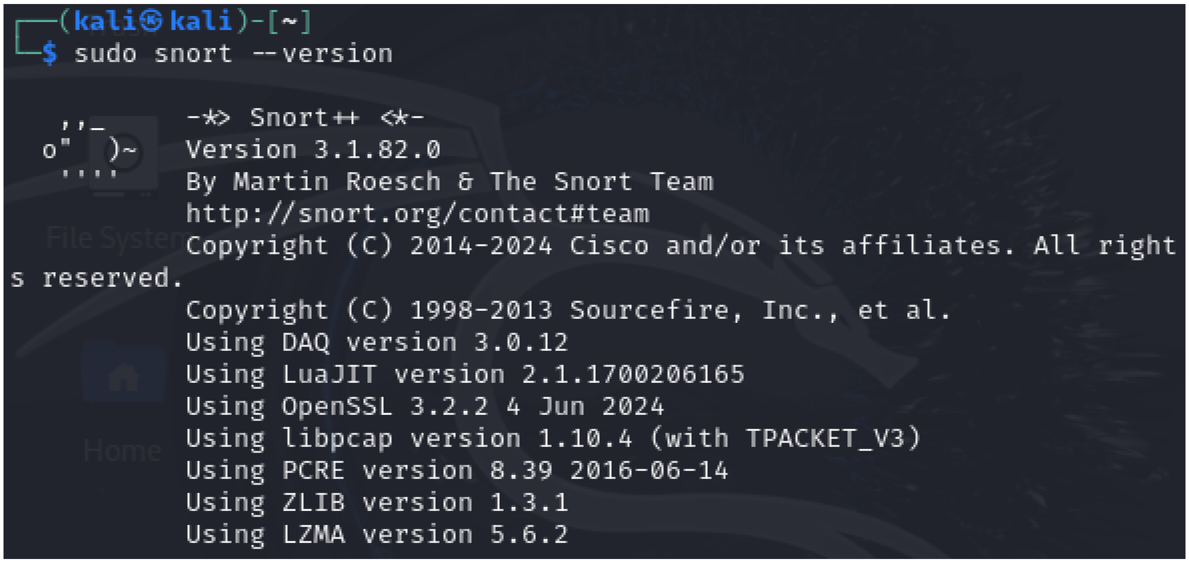
Suricata vs Snort

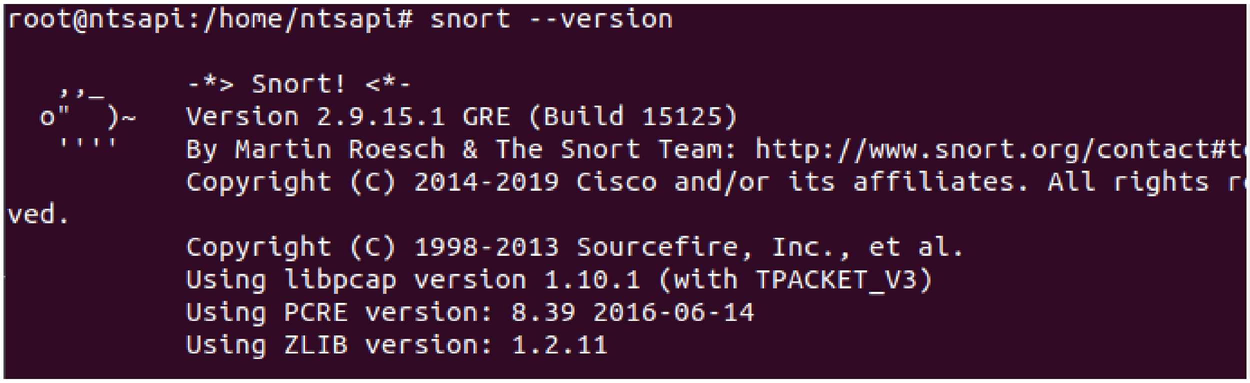
1. Setup and Basic Configuration

- Install Snort on a separate virtual machine.



```
┌──(kali㉿kali)-[~]
└─$ sudo snort --version

   ,,_     -*> Snort++ <*-
  o"  )~   Version 3.1.82.0
   ''''    By Martin Roesch & The Snort Team
           http://snort.org/contact#team
           Copyright (C) 2014-2024 Cisco and/or its affiliates. All right
s reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using DAQ version 3.0.12
           Using LuaJIT version 2.1.1700206165
           Using OpenSSL 3.2.2 4 Jun 2024
           Using libpcap version 1.10.4 (with TPACKET_V3)
           Using PCRE version 8.39 2016-06-14
           Using ZLIB version 1.3.1
           Using LZMA version 5.6.2
```



```
root@ntsapi:/home/ntsapi# snort --version

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#t
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights r
ved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11
```

- Configure Snort configuration file (snort.conf).

  Sudo nano /etc/snort/snort.conf

- Set up network interfaces for live traffic capture.

  Sudo -A console -q -c /etc/snort/snort.conf -i <interface>

- Configure logging and output options for Snort.

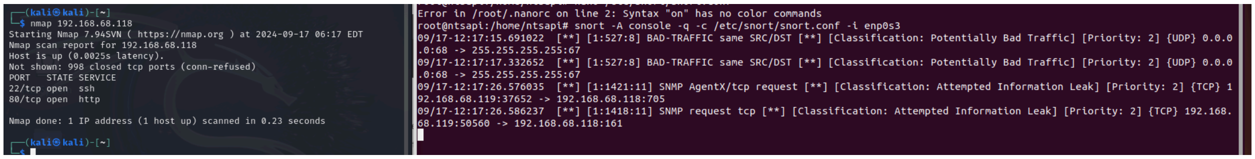
  output log_tcpdump: tcpdump.log

  output alert_fast: snort.alert

  Etc

2. Testing and Verification:

- Start Snort in live mode and verify that it is capturing traffic.



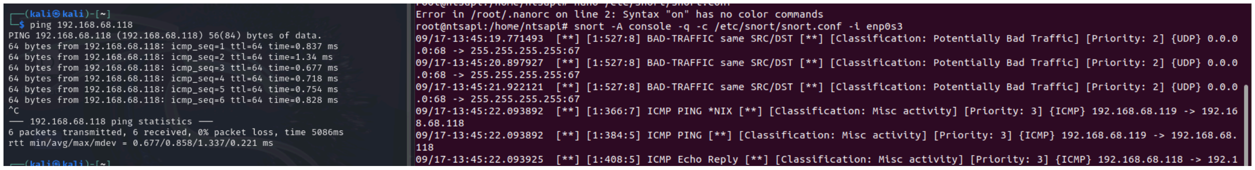
```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.68.118
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-17 06:17 EDT
Nmap scan report for 192.168.68.118
Host is up (0.0025s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

```
Error in /root/.nanorc on line 2: Syntax "on" has no color commands
root@ntsapi:/home/ntsapi# snort -A console -q -c /etc/snort/snort.conf -i enp0s3
09/17-12:17:15.691022  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0
.0:68 -> 255.255.255.255:67
09/17-12:17:17.332652  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0
.0:68 -> 255.255.255.255:67
09/17-12:17:26.576035  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 1
92.168.68.119:37652 -> 192.168.68.118:705
09/17-12:17:26.586237  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.
68.119:50560 -> 192.168.68.118:161
```

- Generate test traffic using tools like ping, nmap, or other network utilities.



```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.68.118
PING 192.168.68.118 (192.168.68.118) 56(84) bytes of data.
64 bytes from 192.168.68.118: icmp_seq=1 ttl=64 time=0.837 ms
64 bytes from 192.168.68.118: icmp_seq=2 ttl=64 time=1.34 ms
64 bytes from 192.168.68.118: icmp_seq=3 ttl=64 time=0.677 ms
64 bytes from 192.168.68.118: icmp_seq=4 ttl=64 time=0.718 ms
64 bytes from 192.168.68.118: icmp_seq=5 ttl=64 time=0.754 ms
64 bytes from 192.168.68.118: icmp_seq=6 ttl=64 time=0.828 ms
^C
─── 192.168.68.118 ping statistics ───
6 packets transmitted, 6 received, 0% packet loss, time 5086ms
rtt min/avg/max/mdev = 0.677/0.858/1.337/0.221 ms

┌──(kali㉿kali)-[~]
```

```
root@ntsapi:/home/ntsapi# nano /etc/snort/snort.conf
Error in /root/.nanorc on line 2: Syntax "on" has no color commands
root@ntsapi:/home/ntsapi# snort -A console -q -c /etc/snort/snort.conf -i enp0s3
09/17-13:45:19.771493  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0
.0:68 -> 255.255.255.255:67
09/17-13:45:20.897927  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0
.0:68 -> 255.255.255.255:67
09/17-13:45:21.922121  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0
.0:68 -> 255.255.255.255:67
09/17-13:45:22.093892  [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.68.119 -> 192.16
8.68.118
09/17-13:45:22.093892  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.68.119 -> 192.168.68.
118
09/17-13:45:22.093925  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.68.118 -> 192.1
```

3. Rule Creation and Customization

- Review the rule syntax and structure for Snort.
  Done
- Develop a set of custom rules that detect specific network behaviors or threats (e.g., SQL injection attempts, malware traffic patterns).

- Test each rule by generating/downloading appropriate test traffic to ensure they trigger correctly in both Suricata and Snort.

4. Rule Optimization (Optional)

- Optimize the custom rules to reduce false positives and improve detection accuracy.

5. Deliverables

- Screenshots or text files showing the configuration of Snort.
- Logs demonstrating both Suricata and Snort successfully capturing and logging test traffic.
- A brief comparison report highlighting differences in configuration and initial observations.
- A comparison report on the effectiveness and performance of custom rules between Suricata and Snort -A brief report on the process of optimization and any challenges encountered (optional).