# Snort- V2 & Snort- V3

## Contents


Snort- V2 ..... 2

**1. Download & Extract Source Files** ..... 2

**2. Install Required Dependencies** ..... 2

**3. Compile & Install DAQ** ..... 2

**4. Compile & Install Snort** ..... 2

**5. Verify Snort Installation** ..... 3

**6. Fixing Snort Binary Path** ..... 3

**7. Create Required Directories** ..... 3

**8. Copy Configuration Files** ..... 3

**9. Create a Test Rule** ..... 3

**10. Test Configuration** ..... 3

**11. Created Minimal Config (`snort-min.conf`)** ..... 4

Phase 1: Getting Started with Snort ..... 4

Phase 2: Understanding Snort Modes ..... 5

Phase 3: Rule Management ..... 10

Phase 4: Updating and Managing Rules ..... 11

Phase 5: Advanced Learning Topics ..... 11

Snort- V3 ..... 13

**Why Scripting in Snort 3 Matters** ..... 16

**1. What You'll Work With** ..... 16

**2. Start Simple — Basic Lua Example in Snort** ..... 16

**3. Custom Logic Example — Conditional Loading** ..... 17

**5. Advanced Lua Example — Custom Counters** ..... 18

**Next Level** ..... 18

# Snort- V2

## Snort Installation Report on RHEL

**Prepared by:** Saba
**Platform:** RHEL (Red Hat Enterprise Linux)
**Snort Version:** 2.9.20 GRE (Build 82)
**Installation Type:** Source-based Manual Compilation
**Interface Used:** `ens160`

### 1. Download & Extract Source Files

- **Snort source:** `snort-2.9.20.tar.gz`
- **DAQ library**
- Extracted to `/usr/src/`

### 2. Install Required Dependencies

```
sudo dnf install -y libpcap libpcap-devel pcre pcre-devel zlib zlib-devel
libdnet libdnet-devel libtirpc libtirpc-devel
sudo dnf groupinstall "Development Tools"
```

**Issue:**
`configure: error: Can't find pcap library`

✅ **Solution:** Installed `libpcap-devel`

### 3. Compile & Install DAQ

```
./configure; make; sudo make install
```

**Issue:**
`libtirpc` headers not found.

✅ **Solution:**

```
CPPFLAGS="-I/usr/include/tirpc" LDFLAGS="-ltirpc" ./configure
```

### 4. Compile & Install Snort

```
./configure --enable-sourcefire
make
sudo make install
```

**Issue:**
Wrong option used `--enablble-open-appid`

✅ **Solution:** Corrected typo to `--enable-open-appid`

## 5. Verify Snort Installation

```
sudo /usr/local/bin/snort -V
```

✅ Output showed correct version and components.

## 6. Fixing Snort Binary Path

**Issue:**
```
sudo: snort: command not found
```

✅ **Solution:** Full path used:

```
sudo /usr/local/bin/snort -v
```

## 7. Create Required Directories

```
sudo mkdir -p /etc/snort/rules
sudo mkdir /var/log/snort
sudo mkdir /usr/local/lib/snort_dynamicrules
```

## 8. Copy Configuration Files

```
sudo cp *.conf *.map *.dtd /etc/snort/
sudo cp classification.config reference.config /etc/snort/
```

## 9. Create a Test Rule

```
sudo touch /etc/snort/rules/local.rules
echo 'alert icmp any any -> any any (msg:"ICMP Packet Detected";
sid:1000001; rev:1;)' | sudo tee /etc/snort/rules/local.rules
```

## 10. Test Configuration

```
sudo /usr/local/bin/snort -T -c /etc/snort/snort.conf
```

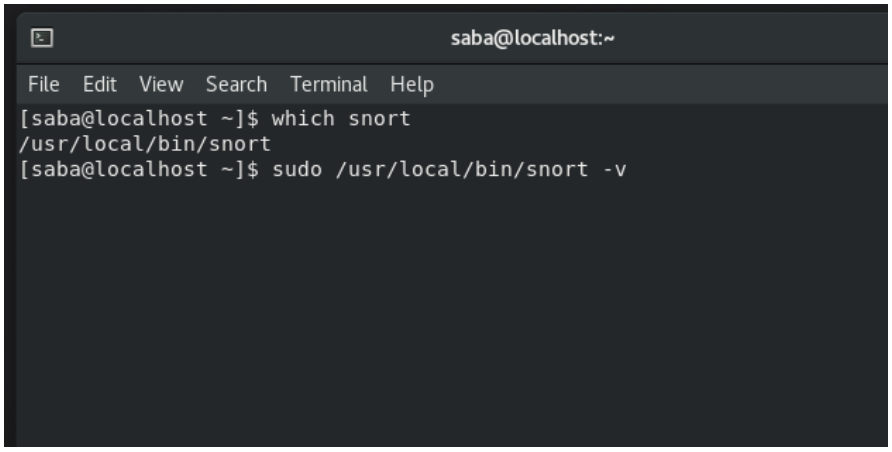
## 11. Created Minimal Config (`snort-min.conf`)

Only referenced working rules:

```
include /etc/snort/rules/local.rules
```

**Passed validation**

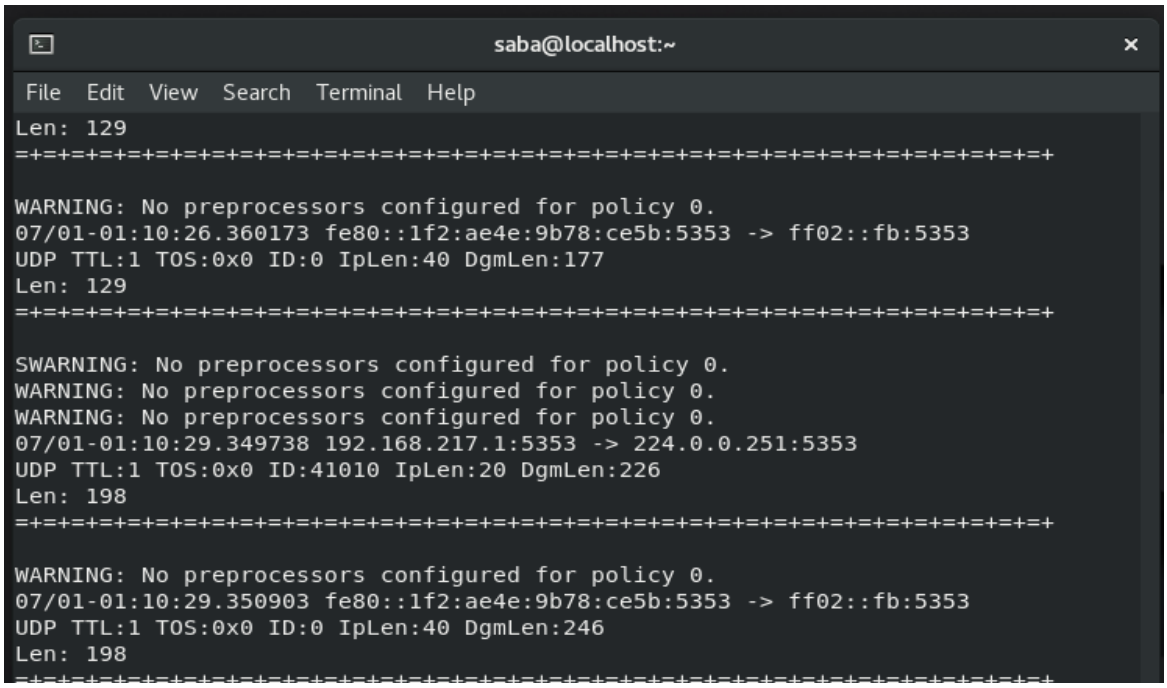
```
sudo /usr/local/bin/snort -T -c /etc/snort/snort-min.con
```




# Phase 1: Getting Started with Snort

### 1.1. Install Snort (Already Done)

sudo /usr/local/bin/snort -V


saba@localhost:~
File Edit View Search Terminal Help
Len: 129
=+=+=+

### 1.2. Snort Directory Structure (Default paths when built from source)

- Config: /etc/snort/snort.conf

```
[saba@localhost ~]$ ls /etc/snort/snort.conf
/etc/snort/snort.conf
[saba@localhost ~]$ ls /etc/snort
attribute_table.dtd    gen-msg.map       snort.conf      unicode.map
classification.config  reference.config  snort-min.conf
file_magic.conf        rules             threshold.conf
[saba@localhost ~]$
```

- Rules: /etc/snort/rules/

```
[saba@localhost ~]$ cd /etc/snort
[saba@localhost snort]$ cd rules
[saba@localhost rules]$ ls
app-detect.rules        botnet-cnc.rules       browser-plugins.rules
attack-responses.rules  browser-chrome.rules   browser-webkit.rules
backdoor.rules          browser-firefox.rules  chat.rules
bad-traffic.rules       browser-ie.rules       local.rules
blacklist.rules         browser-other.rules
[saba@localhost rules]$
```

- Logs: /var/log/snort/

```
[saba@localhost ~]$ ls /var/log/snort
snort.log.1750917524  snort.log.1750918602
[saba@localhost ~]$
```

- Executable: /usr/local/bin/snort

```
[saba@localhost ~]$ ls /usr/local/bin/snort
/usr/local/bin/snort
[saba@localhost ~]$ ls /usr/local/bin
appid_detector_builder.sh  snort   u2openappid  u2streamer
daq-modules-config         u2boat  u2spewfoo
[saba@localhost ~]$
```

## Phase 2: Understanding Snort Modes

### 2.1. Sniffer Mode

View real-time network packets (like tcpdump)

**Command:**

sudo snort -v -i ens160

- -v: verbose output
- -i ens160: specify interface

```
saba@localhost:~
File  Edit  View  Search  Terminal  Help
[saba@localhost ~]$ sudo snort -v -i ens160
[sudo] password for saba:
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens160".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reser
ved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.42 2018-03-20
           Using ZLIB version: 1.2.11
```

```
saba@localhost:~
File  Edit  View  Search  Terminal  Help
UDP TTL:1 TOS:0x0 ID:41364 IpLen:20 DgmLen:283
Len: 255
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
07/01-01:49:39.700296 fe80::1f2:ae4e:9b78:ce5b:5353 -> ff02::fb:5353
UDP TTL:1 TOS:0x0 ID:0 IpLen:40 DgmLen:303
Len: 255
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
07/01-01:49:39.702030 192.168.217.1:5353 -> 224.0.0.251:5353
UDP TTL:1 TOS:0x0 ID:41365 IpLen:20 DgmLen:230
Len: 202
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
07/01-01:49:39.704339 fe80::1f2:ae4e:9b78:ce5b:5353 -> ff02::fb:5353
UDP TTL:1 TOS:0x0 ID:0 IpLen:40 DgmLen:250
Len: 202
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

**Advanced Tip:** Use -d -e for decoding and printing headers:

sudo snort -dev -i ens160

```
saba@localhost:~
File  Edit  View  Search  Terminal  Help
32 64 61 39 36 32 30 64 64 66 34 05 6C 6F 63 61  2da9620ddf4.loca
6C 00 00 2F 80 01 00 00 00 78 00 05 C0 46 00 01  l../.....x...F..
40                                               @

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
07/01-01:51:26.914476 00:50:56:C0:00:08 -> 33:33:00:00:00:FB type:0x86DD len:0xBF
fe80::1f2:ae4e:9b78:ce5b:5353 -> ff02::fb:5353 UDP TTL:1 TOS:0x0 ID:0 IpLen:40 DgmL
en:177
Len: 129
00 00 84 00 00 00 00 01 00 00 00 01 24 37 32 63  ............$72c
33 34 30 63 31 2D 34 30 63 30 2D 34 66 32 66 2D  340c1-40c0-4f2f-
61 63 64 61 2D 37 32 64 61 39 36 32 30 64 64 66  acda-72da9620ddf
34 05 6C 6F 63 61 6C 00 00 01 80 01 00 00 00 78  4.local........x
00 04 0A 0A C9 81 24 37 32 63 33 34 30 63 31 2D  ......$72c340c1-
34 30 63 30 2D 34 66 32 66 2D 61 63 64 61 2D 37  40c0-4f2f-acda-7
32 64 61 39 36 32 30 64 64 66 34 05 6C 6F 63 61  2da9620ddf4.loca
6C 00 00 2F 80 01 00 00 00 78 00 05 C0 46 00 01  l../.....x...F..
40                                               @

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

## 2.2. Packet Logger Mode

Save captured packets to disk

**Command:**

sudo snort -dev -i ens160 -l /var/log/snort

- -l: log directory
- Output is stored in tcpdump-style format

```
saba@localhost:~
File  Edit  View  Search  Terminal  Help
        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved
.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.42 2018-03-20
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=5795)
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
```

**View logs:**

ls /var/log/snort/

new file created

```
[saba@localhost ~]$ ls /var/log/snort/
snort.log.1750917524  snort.log.1750918602  snort.log.1751349636
[saba@localhost ~]$
```

Setting up rules:

```
[saba@localhost ~]$ cd /etc/snort/
[saba@localhost snort]$ ls
attribute_table.dtd    gen-msg.map       snort.conf      unicode.map
classification.config  reference.config  snort-min.conf
file_magic.conf        rules             threshold.conf
[saba@localhost snort]$ cp snort.conf snort.conf.back
cp: cannot create regular file 'snort.conf.back': Permission denied
[saba@localhost snort]$ sudo cp snort.conf snort.conf.back
[saba@localhost snort]$ ls
attribute_table.dtd    gen-msg.map       snort.conf       threshold.conf
classification.config  reference.config  snort.conf.back  unicode.map
file_magic.conf        rules             snort-min.conf
[saba@localhost snort]$ sudo nano /etc/snort/snort.conf
[saba@localhost snort]$ sudo vi /etc/snort/snort.conf

[4]+  Stopped                 sudo vi /etc/snort/snort.conf
[saba@localhost snort]$ sudo gedit /etc/snort/snort.conf


** (gedit:8017): WARNING **: 02:53:29.692: Set document metadata failed: Setting at
tribute metadata::gedit-spell-language not supported

** (gedit:8017): WARNING **: 02:53:29.693: Set document metadata failed: Setting at
tribute metadata::gedit-encoding not supported

** (gedit:8017): WARNING **: 02:53:33.636: Set document metadata failed: Setting at
tribute metadata::gedit-position not supported
[saba@localhost snort]$
[saba@localhost snort]$
[saba@localhost snort]$
[saba@localhost snort]$ ls
```

We will configure rules in local.rules only so remove extra rules from snort.conf

**2.3. NIDS/NIPS Mode**

Use rules to detect or block malicious traffic

**Command:**

sudo snort -c /etc/snort/snort.conf -i ens160

- -c: config file with rule path, preprocessors, output options

```
saba@
File  Edit  View  Search  Terminal  Help
[saba@localhost ~]$ sudo snort -c /etc/snort/snort.conf -i ens160
[sudo] password for saba:
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414
144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118
43 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1024:65535 ]
PortVar 'SSH_PORTS' defined :  [ 22 ]
PortVar 'FTP_PORTS' defined :  [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined :  [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined :  [ 80:81 110 143 311 383 591 593
8 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 80
 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'GTP_PORTS' defined :  [ 2123 2152 3386 ]
Detection:
   Search-Method = AC-Full-Q
    Split Any/Any group = enabled
    Search-Method-Optimizations = enabled
```

```
saba@localhost:~
File  Edit  View  Search  Terminal  Help
Reload thread starting...
Reload thread started, thread 0x7f16ef292700 (8601)
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.42 2018-03-20
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.2  <Build 1>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_S7COMMPLUS  Version 1.0  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
Commencing packet processing (pid=8598)
```

sudo tail -f  /var/log/snort/alert

```
[saba@localhost rules]$
[saba@localhost rules]$ sudo tail -f /var/log/snort/alert
07/01-04:01:21.201217 192.168.217.1 -> 192.168.217.148
ICMP TTL:128 TOS:0x0 ID:14820 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:59  ECHO

[**] [1:1000001:1] - ICMP Packet Detected - [**]
[Priority: 0]
07/01-04:01:21.201323 192.168.217.148 -> 192.168.217.1
ICMP TTL:64 TOS:0x0 ID:30049 IpLen:20 DgmLen:60
Type:0  Code:0  ID:1  Seq:59  ECHO REPLY
```

## Phase 3: Rule Management

### 3.1. Understand Rule Syntax

Snort rule format:

action protocol src_ip src_port -> dst_ip dst_port (options)

**Example:**

alert tcp any any -> any 80 (msg:"HTTP access"; sid:1000001; rev:1;)

### 3.2. Create a Custom Rule

Open local.rules

Write own rule

```
[saba@localhost rules]$ sudo gedit local.rules
```

Open | local.rules /etc/snort/rules | Save

```
alert icmp any any -> any any (msg:"ICMP Packet Detected"; sid:1000001; rev:2;)
alert tcp any any -> any 80 (msg:"HTTP Traffic Detected"; sid:1000002; rev:1;)
```

Next:

```
[saba@localhost ~]$ sudo snort -c /etc/snort/snort.conf -i ens160
```

```
            Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
            Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
            Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
            Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
            Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
            Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
            Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
            Preprocessor Object: SF_POP  Version 1.0  <Build 1>
            Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
Commencing packet processing (pid=14751)
```

Check alert

```
[saba@localhost rules]$ sudo tail -f /var/log/snort/alert
[**] [1:1000001:2] ICMP Packet Detected [**]
[Priority: 0]
07/01-04:11:17.257127 fe80::1f2:ae4e:9b78:ce5b -> ff02::16
IPV6-ICMP TTL:1 TOS:0x0 ID:256 IpLen:40 DgmLen:76

[**] [1:1000001:1] - ICMP Packet Detected - [**]
[Priority: 0]
07/01-04:11:17.257127 fe80::1f2:ae4e:9b78:ce5b -> ff02::16
IPV6-ICMP TTL:1 TOS:0x0 ID:256 IpLen:40 DgmLen:76

[**] [1:1000002:1] HTTP Traffic Detected [**]
[Priority: 0]
07/01-04:16:04.689893 192.168.217.148:60518 -> 34.107.221.82:80
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x5DE60FDF  Ack: 0x2CC718F9  Win: 0x7210  TcpLen: 20

[**] [1:1000002:1] HTTP Traffic Detected [**]
[Priority: 0]
07/01-04:16:22.008601 192.168.217.148:55140 -> 192.178.24.163:80
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
```

## Phase 4: Updating and Managing Rules

### 4.1. Use PulledPork or Snort.org

- Register at https://snort.org
- Download community or subscription rules
- Automate rule updates using **PulledPork**

## Phase 5: Advanced Learning Topics

### 5.1. Preprocessors

- Inspect traffic before rule engine
- Examples: frag3, http_inspect, stream5
- Configured in snort.conf
- Log in unified2, pcap, or alert_fast format
- Integration with **Barnyard2** for log parsing and SIEM forwarding

### 5.3. Inline Mode (IPS)

Requires NFQUEUE or AF_PACKET DAQ

Example (inline drop):

sudo snort -Q --daq afpacket --daq-mode inline -c /etc/snort/snort.conf -i ens160:ens161

### 5.4. Integration with SIEM (e.g., Splunk or QRadar)

- Use log forwarding tools like Barnyard2 or syslog
- Parse Snort alerts into your SIEM for correlation

| Feature | Snort 2 | Snort 3 | Advantage in Snort 3 |
|---|---|---|---|
| **Speed** | Runs on **one CPU core** | Uses **multiple CPU cores** | Much **faster**, can handle more traffic without dropping packets |
| **Rules** | Basic rule options, depends on preprocessors | Advanced rule keywords (e.g., `service:http;`, `tls;`, `file_data;`) | Rules are **smarter, detect modern attacks** |
| **Configuration** | Hard to manage (`snort.conf`) | Easy & flexible (`snort.lua` with Lua scripting) | **Easier to tune & automate** |
| **Protocol Support** | Limited (basic HTTP, DNS, TCP/UDP, ICMP) | Modern protocols (HTTP/2, TLS, SSH, SIP, SCADA, etc.) | Can detect attacks in **new apps & encrypted traffic** |
| **Scalability** | Struggles with large networks | Built for high-performance, large environments | Works better in **enterprise-level traffic** |
| **Future Updates** | Old, maintenance only | Actively developed by Cisco/Talos | **Future-proof** with new rules/features |
| **Integration** | Limited logging formats | Supports JSON & modern outputs | **Better SIEM/SOAR integration** |

# Snort- V3

**sudo apt update && sudo apt upgrade -y**

```
Session  Actions  Edit  View  Help
Get:363 http://http.kali.org/kali kali-rolling/main amd64 libqt6opengl6 amd64 6.8.2+dfsg-10.1 [440 kB]
Get:364 http://http.kali.org/kali kali-rolling/main amd64 libqt6widgets6 amd64 6.8.2+dfsg-10.1 [2,803 kB]
Get:365 http://http.kali.org/kali kali-rolling/main amd64 libqt6openglwidgets6 amd64 6.8.2+dfsg-10.1 [51.3 kB]
Get:366 http://http.kali.org/kali kali-rolling/main amd64 libqt6printsupport6 amd64 6.8.2+dfsg-10.1 [229 kB]
Get:367 http://http.kali.org/kali kali-rolling/main amd64 libqt6sql6 amd64 6.8.2+dfsg-10.1 [154 kB]
Get:368 http://http.kali.org/kali kali-rolling/main amd64 libqt6sql6-sqlite amd64 6.8.2+dfsg-10.1 [68.8 kB]
Get:369 http://http.kali.org/kali kali-rolling/main amd64 libqt6test6 amd64 6.8.2+dfsg-10.1 [200 kB]
Get:370 http://http.kali.org/kali kali-rolling/main amd64 libqt6xml6 amd64 6.8.2+dfsg-10.1 [90.8 kB]
Get:371 http://kali.download/kali kali-rolling/main amd64 libtag-c2 amd64 2.1.1-2 [38.7 kB]
Get:372 http://kali.download/kali kali-rolling/main amd64 libtbb12 amd64 2022.1.0-3 [93.1 kB]
Get:373 http://kali.download/kali kali-rolling/main amd64 libtbbmalloc2 amd64 2022.1.0-3 [51.0 kB]
Get:374 http://kali.download/kali kali-rolling/main amd64 libtbbbind-2-5 amd64 2022.1.0-3 [14.0 kB]
Get:375 http://kali.download/kali kali-rolling/main amd64 libtext-iconv-perl amd64 1.7-8.1 [14.3 kB]
Get:376 http://kali.download/kali kali-rolling/main amd64 thunar amd64 4.20.5-1 [414 kB]
Get:377 http://kali.download/kali kali-rolling/main amd64 thunar-data all 4.20.5-1 [918 kB]
Get:378 http://kali.download/kali kali-rolling/main amd64 libthunarx-3-0 amd64 4.20.5-1 [124 kB]
Get:379 http://kali.download/kali kali-rolling/main amd64 libunbound8 amd64 1.24.0-2 [614 kB]
Get:380 http://http.kali.org/kali kali-rolling/main amd64 libwayland-server0 amd64 1.24.0-2+b1 [35.5 kB]
Get:381 http://kali.download/kali kali-rolling/main amd64 libwebkitgtk-6.0-4 amd64 2.48.6-1 [22.2 MB]
Get:382 http://kali.download/kali kali-rolling/main amd64 wireplumber amd64 0.5.11-1 [116 kB]
Get:384 http://kali.download/kali kali-rolling/main amd64 libwsutil16 amd64 4.4.9-1 [127 kB]
Get:385 http://kali.download/kali kali-rolling/main amd64 libwiretap15 amd64 4.4.9-1 [267 kB]
Get:386 http://kali.download/kali kali-rolling/main amd64 libwireshark18 amd64 4.4.9-1 [21.2 MB]
Get:387 http://kali.download/kali kali-rolling/main amd64 tshark amd64 4.4.9-1 [175 kB]
Get:388 http://kali.download/kali kali-rolling/main amd64 wireshark amd64 4.4.9-1 [4,626 kB]
Get:389 http://kali.download/kali kali-rolling/main amd64 wireshark-common amd64 4.4.9-1 [598 kB]
Get:390 http://kali.download/kali kali-rolling/main amd64 libwireshark-data all 4.4.9-1 [449 kB]
Get:391 http://kali.download/kali kali-rolling/main amd64 libxfce4ui-utils amd64 4.20.2-1 [90.2 kB]
Get:392 http://kali.download/kali kali-rolling/main amd64 libxfce4windowing-common all 4.20.4-1 [55.6 kB]
Get:393 http://kali.download/kali kali-rolling/main amd64 libxfce4windowing-0-0 amd64 4.20.4-1 [112 kB]
Get:394 http://kali.download/kali kali-rolling/main amd64 libxklavier16 amd64 5.4-6 [43.3 kB]
Get:395 http://kali.download/kali kali-rolling/main amd64 libyelp-1-0 amd64 49.0-3 [159 kB]
Get:396 http://kali.download/kali kali-rolling/main amd64 m4 amd64 1.4.20-1 [324 kB]
Get:397 http://kali.download/kali kali-rolling/main amd64 mesa-vdpau-drivers amd64 25.2.3-1 [20.7 kB]
Get:398 http://kali.download/kali kali-rolling/main amd64 mesa-vulkan-drivers amd64 25.2.3-1 [16.7 MB]
Get:399 http://kali.download/kali kali-rolling/main amd64 mingw-w64-common all 13.0.0-1 [6,146 kB]
Get:400 http://kali.download/kali kali-rolling/main amd64 mingw-w64-i686-dev all 13.0.0-1 [4,533 kB]
Get:401 http://kali.download/kali kali-rolling/main amd64 mingw-w64-x86-64-dev all 13.0.0-1 [5,176 kB]
Get:402 http://kali.download/kali kali-rolling/main amd64 ncompress amd64 5.0-4 [23.9 kB]
Get:403 http://kali.download/kali kali-rolling/main amd64 nginx amd64 1.28.0-6 [672 kB]
Get:404 http://kali.download/kali kali-rolling/main amd64 nginx-common all 1.28.0-6 [110 kB]
Get:405 http://http.kali.org/kali kali-rolling/main amd64 ngrep amd64 1.47+ds1-7 [37.0 kB]
Get:407 http://kali.download/kali kali-rolling/main amd64 offsec-awae-python2 amd64 2025.4.0 [1,040 kB]
Get:408 http://http.kali.org/kali kali-rolling/main amd64 openjdk-21-jdk amd64 21.0.9~5ea-1 [3,580 kB]
Get:409 http://http.kali.org/kali kali-rolling/main amd64 openjdk-21-jdk-headless amd64 21.0.9~5ea-1 [83.5 MB]
Get:410 http://http.kali.org/kali kali-rolling/main amd64 openjdk-21-jre amd64 21.0.9~5ea-1 [219 kB]
Get:411 http://http.kali.org/kali kali-rolling/main amd64 openjdk-21-jre-headless amd64 21.0.9~5ea-1 [42.2 MB]
94% [411 openjdk-21-jre-headless 40.7 MB/42.2 MB 96%]
```

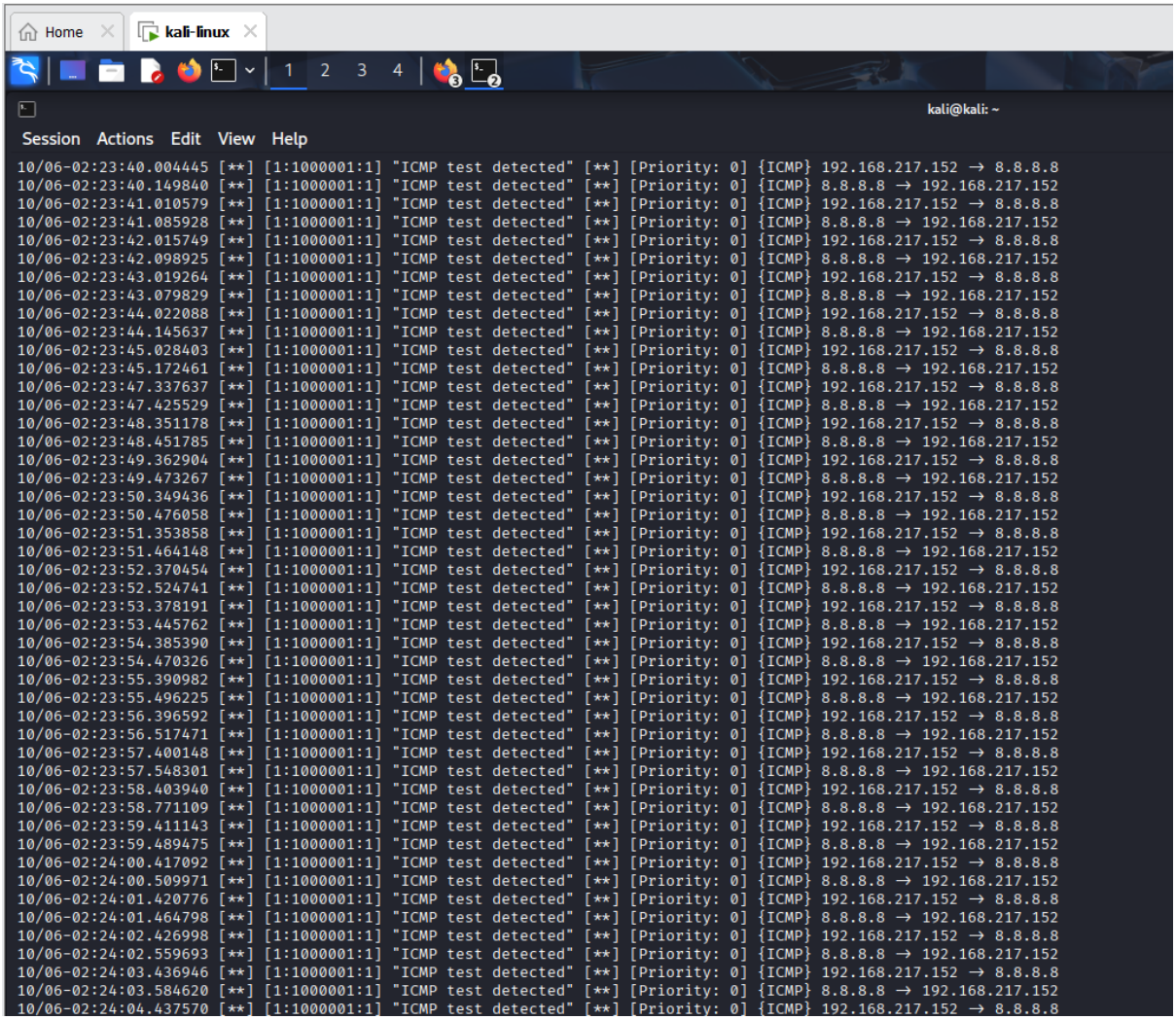
**sudo apt install -y build-essential autotools-dev libtool autoconf **
**cmake pkg-config git libpcap-dev libpcre3-dev libdumbnet-dev **
**zlib1g-dev liblzma-dev libssl-dev libluajit-5.1-dev lua5.1 **
**libnspr4-dev libglib2.0-dev bison flex cpputest libcmocka-dev **
**python3 python3-pip**

```
Setting up libpcre2-dev:amd64 (10.46-1) ...
Setting up libtool (2.5.4-5) ...
Setting up libselinux1-dev:amd64 (3.8.1-1) ...
Setting up libpcre3-dev:amd64 (2:8.39-15.1) ...
Setting up pkgconf:amd64 (1.8.1-4) ...
Setting up libltdl-dev:amd64 (2.5.4-5) ...
Setting up libdbus-1-dev:amd64 (1.16.2-2) ...
Setting up pkg-config:amd64 (1.8.1-4) ...
Setting up cmake (4.1.1+really3.31.6-2) ...
Setting up libmount-dev:amd64 (2.41.1-1) ...
Setting up libglib2.0-dev-bin (2.84.4-3) ...
Setting up libgio-2.0-dev:amd64 (2.84.4-3) ...
Processing triggers for sgml-base (1.31+nmu1) ...
Processing triggers for kali-menu (2025.4.1) ...
Processing triggers for doc-base (0.11.2) ...
Processing 2 added doc-base files...
Setting up libpcap0.8-dev:amd64 (1.10.5-2) ...
Processing triggers for libc-bin (2.41-12) ...
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for libglib2.0-0t64:amd64 (2.84.4-3) ...
Setting up libglib2.0-dev:amd64 (2.84.4-3) ...
Setting up libpcap-dev:amd64 (1.10.5-2) ...

┌──(kali㉿kali)-[~]
└─$
```

**cd /opt**
**sudo git clone https://github.com/snort3/libdaq.git**
**cd libdaq**
**sudo ./bootstrap**
**sudo ./configure**
**sudo make**
**sudo make install**

**ping 8.8.8.8**

**sudo snort -c /usr/local/etc/snort/snort.lua -R /usr/local/etc/snort/rules/local.rules -i eth0 -A alert_fast**



## Open: sudo nano /usr/local/etc/snort/rules/local.rules

```
GNU nano 8.6                                                    /usr/local/etc/snort/rules/local.rules
alert icmp any any -> any any (msg:"ICMP Packet Detected"; sid:1000001; rev:1;)
alert udp any any -> any 53 (msg:"DNS Query Detected"; sid:1000002; rev:1;)
alert tcp any any -> any 80 (msg:"HTTP Traffic Detected"; sid:1000003; rev:1;)
alert tcp any any -> any 443 (msg:"HTTPS Traffic Detected"; sid:1000004; rev:1;)
alert tcp any any -> any 22 (msg:"SSH Connection Detected"; sid:1000005; rev:1;)
alert tcp any any -> any 21 (msg:"FTP Connection Detected"; sid:1000006; rev:1;)
alert tcp any any -> any 25 (msg:"SMTP Email Traffic Detected"; sid:1000007; rev:1;)
alert tcp any any -> any 23 (msg:"TELNET Connection Detected"; sid:1000008; rev:1;)
alert tcp any any -> any any (msg:"General TCP Packet Detected"; sid:1000009; rev:1;)
```

```
search engine (ac_bnfa)
                instances: 2
                 patterns: 438
            pattern chars: 2602
               num states: 1832
          num match states: 392
              memory scale: KB
             total memory: 71.2812
          pattern memory: 19.6484
         match list memory: 28.4375
         transition memory: 22.9453
appid: MaxRss diff: 2944
appid: patterns loaded: 300

pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
10/06-02:50:40.473133 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 104.18.11.224:443 -> 192.168.217.152:43194
10/06-02:50:40.473151 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 104.18.11.224:443 -> 192.168.217.152:43194
10/06-02:50:40.847319 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 104.18.11.224:443 -> 192.168.217.152:43194
10/06-02:50:40.848121 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 192.168.217.152:43194 -> 104.18.11.224:443
10/06-02:50:40.848121 [**] [1:1000004:1] "HTTPS Traffic Detected" [**] [Priority: 0] {TCP} 192.168.217.152:43194 -> 104.18.11.224:443
10/06-02:50:41.872746 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:39440
10/06-02:50:41.874562 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:39440
10/06-02:50:42.170433 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:39440
10/06-02:50:42.172402 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:39440
10/06-02:50:42.486142 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:39440
10/06-02:50:42.170433 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:39440
10/06-02:50:42.486215 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 192.168.217.152:39440 -> 44.228.249.3:80
10/06-02:50:42.486215 [**] [1:1000003:1] "HTTP Traffic Detected" [**] [Priority: 0] {TCP} 192.168.217.152:39440 -> 44.228.249.3:80
10/06-02:50:59.973311 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:34110
10/06-02:50:59.974832 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:34110
10/06-02:51:00.269693 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:34110
10/06-02:51:00.273046 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:34110
10/06-02:51:00.853111 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:34110
10/06-02:51:00.269693 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:34110
10/06-02:51:00.853189 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 192.168.217.152:34110 -> 44.228.249.3:80
10/06-02:51:00.853189 [**] [1:1000003:1] "HTTP Traffic Detected" [**] [Priority: 0] {TCP} 192.168.217.152:34110 -> 44.228.249.3:80
10/06-02:51:09.478434 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:51420
10/06-02:51:09.479923 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:51420
10/06-02:51:09.904179 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:51420
10/06-02:51:10.296419 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:51420
10/06-02:51:10.296521 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 192.168.217.152:51420 -> 44.228.249.3:80
10/06-02:51:10.296521 [**] [1:1000003:1] "HTTP Traffic Detected" [**] [Priority: 0] {TCP} 192.168.217.152:51420 -> 44.228.249.3:80
10/06-02:51:19.867140 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:34132
10/06-02:51:19.868835 [**] [1:1000009:1] "General TCP Packet Detected" [**] [Priority: 0] {TCP} 44.228.249.3:80 -> 192.168.217.152:34132
```

# Why Scripting in Snort 3 Matters

Snort 3 replaced Snort 2's static `.conf` files with a full **Lua scripting environment**.
That means you can now:

- Dynamically enable/disable preprocessors and modules
- Adjust performance tuning variables (buffers, threads, limits)
- Create **custom inspection logic**
- Control outputs, alerts, and rule loading dynamically
- Write automation logic (e.g., rotate logs, load rules by time of day, etc.)

# 1. What You'll Work With

Main file:

```
/usr/local/etc/snort/snort.lua
```

Other optional Lua configuration modules:

```
/usr/local/etc/snort/snort_defaults.lua
/usr/local/etc/snort/tweaks.lua
/usr/local/etc/snort/rules/*.lua
```

# 2. Start Simple — Basic Lua Example in Snort

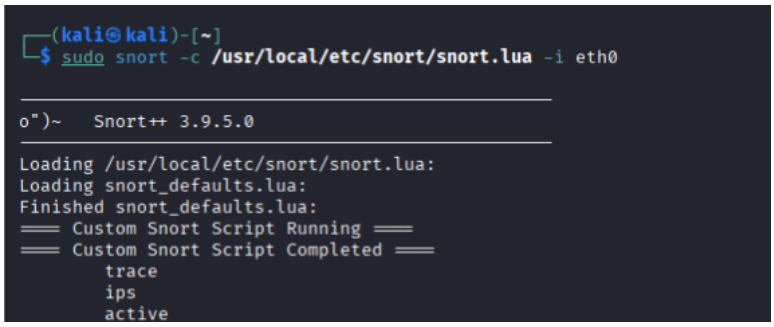
Let's start by adding a **custom Lua block** to your `snort.lua`:

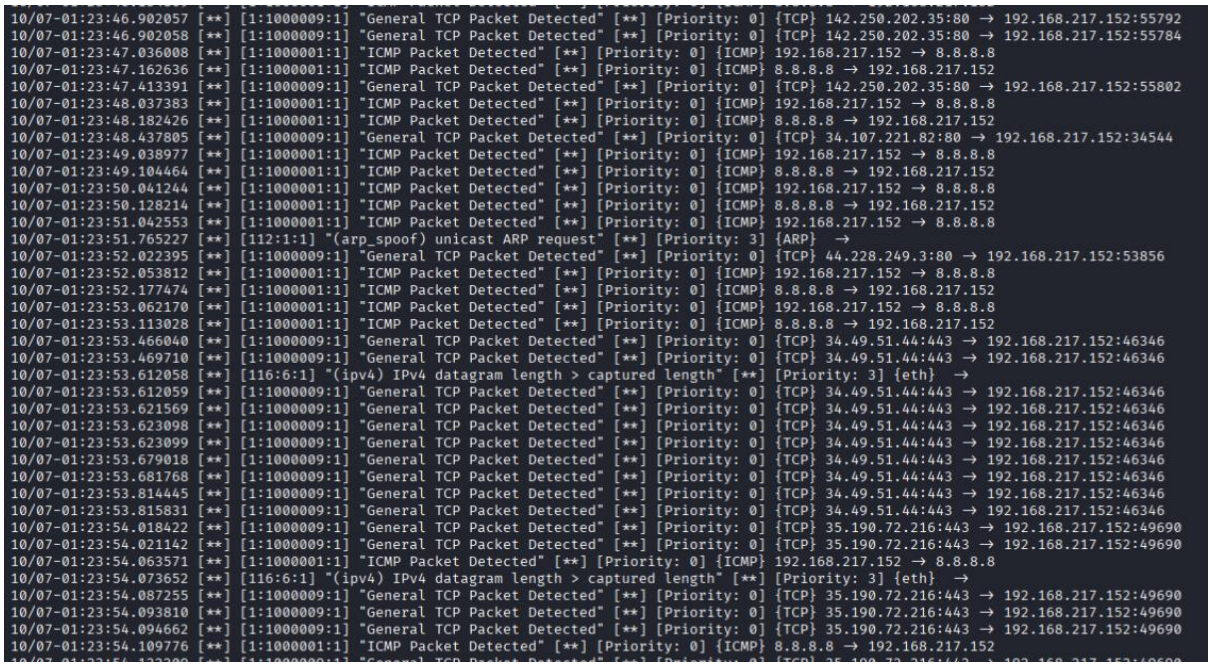
*This sets internal **performance and threading parameters** for Snort. This also activates Snort's **Intrusion Prevention System (IPS)** module with the built-in variables and default rules.*

```
print("==== Custom Snort Script Running ====")

tweaks =
{
   num_threads = 4,
   max_queue = 1024,
}

ips =
{
   enable_builtin_rules = true,
   variables = default_variables
}

print("==== Custom Snort Script Completed ====")
```

Run it with:

```
sudo snort -c /usr/local/etc/snort/snort.lua -i eth0
```




# 3. Custom Logic Example — Conditional Loading

You can use Lua logic to decide **which rule sets** to load dynamically:

```
local hour = os.date("*t").hour

if hour < 12 then
    print("Loading morning rules...")
    include("/usr/local/etc/snort/rules/morning.rules")
else
    print("Loading evening rules...")
    include("/usr/local/etc/snort/rules/evening.rules")
end
```

This allows **time-based adaptive rule loading**, which is impossible in Snort 2. The purpose of this script is to make Snort 3 context-aware and adaptive — meaning it can automatically decide which rule set to load based on conditions such as time of day, network interface, or environment.

# 5. Advanced Lua Example — Custom Counters

You can even track and print metrics:

```
packet_count = 0

function preprocess(pkt)
    packet_count = packet_count + 1
    if packet_count % 100 == 0 then
        print("Processed " .. packet_count .. " packets")
    end
end
```

You can load this as a separate Lua module:

```
/usr/local/etc/snort/custom/preprocessor.lua
```

and include it in `snort.lua`:

```
dofile("/usr/local/etc/snort/custom/preprocessor.lua")
```

# Next Level

Once comfortable:

- Learn about **Snort Lua APIs** (`snort --help-module <module>`).
- Try writing a **custom app-id detector** in Lua.
- Automate rule updates or alert classification.
- Export Snort Lua data directly into **Splunk via HTTP Event Collector (HEC)**.