

**Final Year B. Tech, Sem VII 2022-23**  
**PRN – 2020BTECS00211**  
**Name – Aashita Narendra Gupta**  
**Cryptography And Network Security Lab**  
**Batch: B4**  
**Practical No – 15**

**Title:** Snort Intrusion Detection System (IDS)

In this lab we will explore the Snort IDS. This is a signature based intrusion detection system used to detect network attacks. Snort can also be used as a simple packet logger, however we won't be doing that in this lab. Snort has multiple modes of operation, for the lab we will use snort as a packet sniffer, not inline.

**Theory:**

**SNORT**

Snort is the world's foremost Open Source Intrusion Prevention System (IPS). Snort IPS uses a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generates alerts for users.

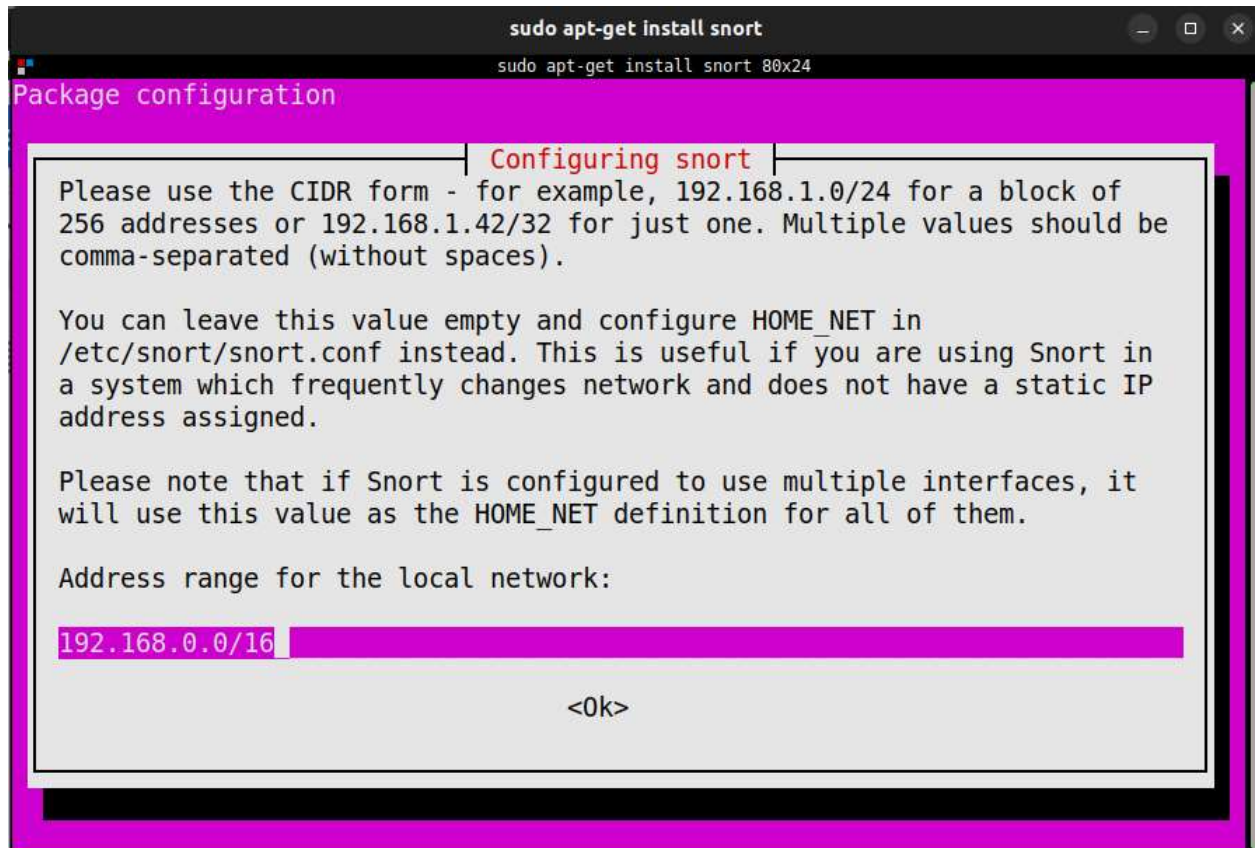
Snort can be deployed inline to stop these packets, as well. Snort has three primary uses: As a packet sniffer like tcpdump, as a packet logger – which is useful for network traffic debugging or can be used as a full-blown network intrusion prevention system. Snort can be downloaded and configured for personal and business use alike.

Snapshots:

### Snort Installation:

```
sudo apt-get install snort
sudo apt-get install snort 80x24

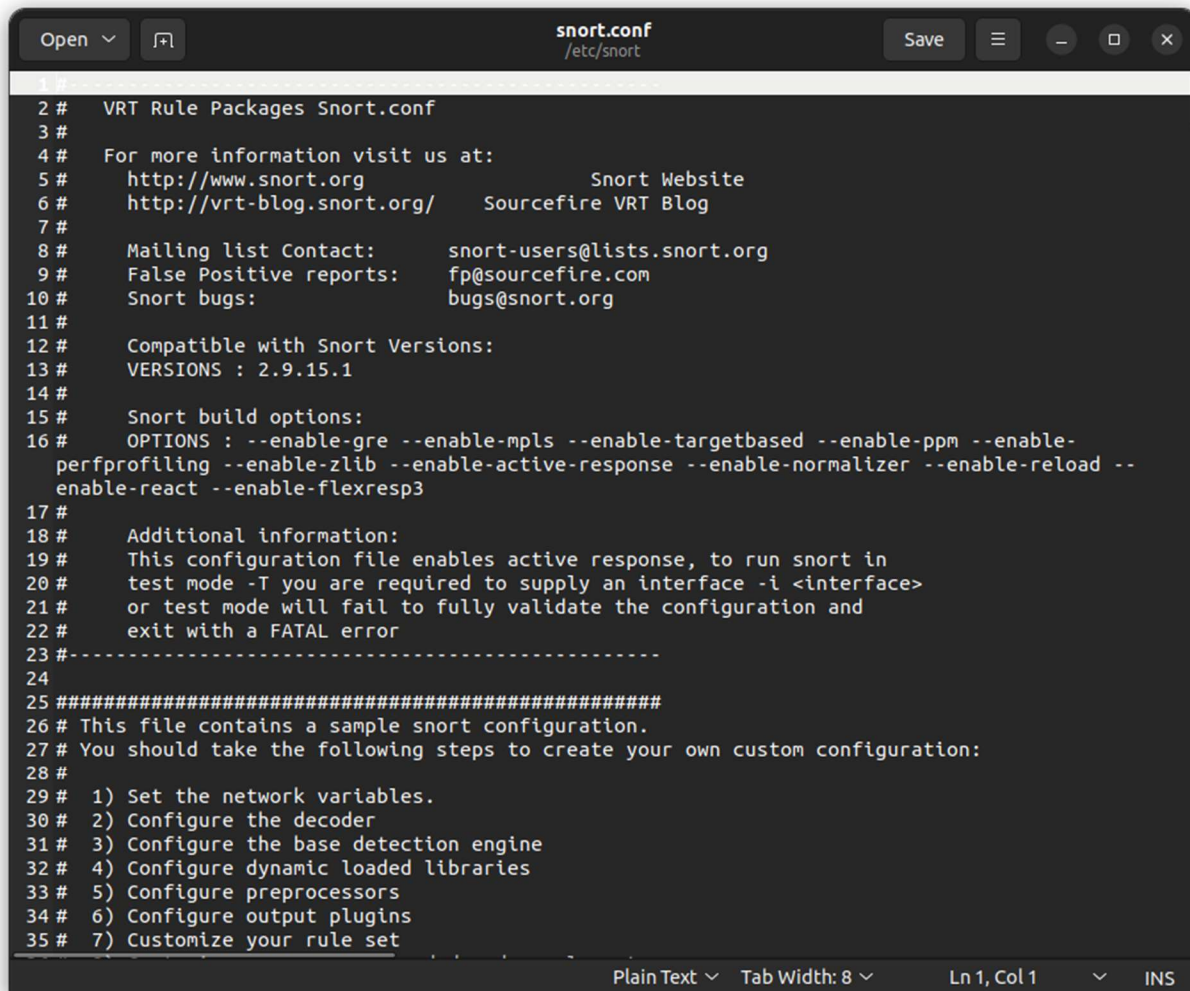
vinayak at Gladiator in ~
$ sudo apt-get install snort
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libdaq2 libdumbnet1 libnetfilter-queue1 oinkmaster snort-common
  snort-common-libraries snort-rules-default
Suggested packages:
  snort-doc
The following NEW packages will be installed:
  libdaq2 libdumbnet1 libnetfilter-queue1 oinkmaster snort snort-common
  snort-common-libraries snort-rules-default
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/2,067 kB of archives.
After this operation, 9,836 kB of additional disk space will be used.
```



```
vinayak@Gladiator:~  
vinayak@Gladiator:~ 80x24  
vinayak at Gladiator in ~  
$ snort -V  
  
    ,,_      -*> Snort! <*-  
   o" _ )~  Version 2.9.15.1 GRE (Build 15125)  
  ' ' '    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.  
  
           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  
  
vinayak at Gladiator in ~  
$
```

```
vinayak at Gladiator in ~  
$ sudo gedit /etc/snort/snort.conf
```

## SNORT Configuration

A screenshot of a text editor window titled 'snort.conf' with the path '/etc/snort'. The window has a dark theme and includes standard window controls (Open, Save, etc.) at the top. The main area displays the contents of the snort.conf file, which is a configuration file for Snort. The file contains various settings, including VRT Rule Packages, contact information, compatible versions, build options, and a list of steps to create a custom configuration. The status bar at the bottom indicates 'Plain Text', 'Tab Width: 8', 'Ln 1, Col 1', and 'INS' mode.

```
1 #
2 # VRT Rule Packages Snort.conf
3 #
4 # For more information visit us at:
5 #   http://www.snort.org           Snort Website
6 #   http://vrt-blog.snort.org/     Sourcefire VRT Blog
7 #
8 #   Mailing list Contact:      snort-users@lists.snort.org
9 #   False Positive reports:    fp@sourcefire.com
10 #   Snort bugs:               bugs@snort.org
11 #
12 #   Compatible with Snort Versions:
13 #   VERSIONS : 2.9.15.1
14 #
15 #   Snort build options:
16 #   OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-
perfprofiling --enable-zlib --enable-active-response --enable-normalizer --enable-reload --
enable-react --enable-flexresp3
17 #
18 #   Additional information:
19 #   This configuration file enables active response, to run snort in
20 #   test mode -T you are required to supply an interface -i <interface>
21 #   or test mode will fail to fully validate the configuration and
22 #   exit with a FATAL error
23 #-----
24
25 #####
26 # This file contains a sample snort configuration.
27 # You should take the following steps to create your own custom configuration:
28 #
29 # 1) Set the network variables.
30 # 2) Configure the decoder
31 # 3) Configure the base detection engine
32 # 4) Configure dynamic loaded libraries
33 # 5) Configure preprocessors
34 # 6) Configure output plugins
35 # 7) Customize your rule set
```

## Checking configurations for the wireless interface

```
vinayak@Gladiator:~  
vinayak@Gladiator:~ 80x24  
vinayak at Gladiator in ~  
$ sudo snort -T -c /etc/snort/snort.conf -i wlp0s20f3  
Running in Test 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 1741 1830  
2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777  
7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300  
8800 8888 8899 9000 9060 9080 9090:9091 9443 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 : [ 80:81 110 143 311 383 591 593 901 1220 14  
14 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:71  
45 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 82  
43 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444  
41080 50002 55555 ]
```

```
dnyaneshwar@fedora:/usr/src/daq-2.0.7/os-daq-modules  
$ ifconfig  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 70 bytes 11117 (10.8 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 70 bytes 11117 (10.8 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.98.204 netmask 255.255.255.0 broadcast 192.168.98.255  
    inet6 fe80::1a87:291e:f8fb:59a7 prefixlen 64 scopeid 0x20<link>  
    inet6 2409:4042:271f:a19a:e65c:a22c:67f5:569f prefixlen 64 scopeid 0x0<global>  
    ether 28:d0:ea:dd:5f:ae txqueuelen 1000 (Ethernet)  
    RX packets 595163 bytes 365759299 (348.8 MiB)  
    RX errors 0 dropped 14194 overruns 0 frame 0  
    TX packets 90583 bytes 23613007 (22.5 MiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
dnyaneshwar@fedora:/usr/src/daq-2.0.7/os-daq-modules  
$
```

**Attacking device Information:**



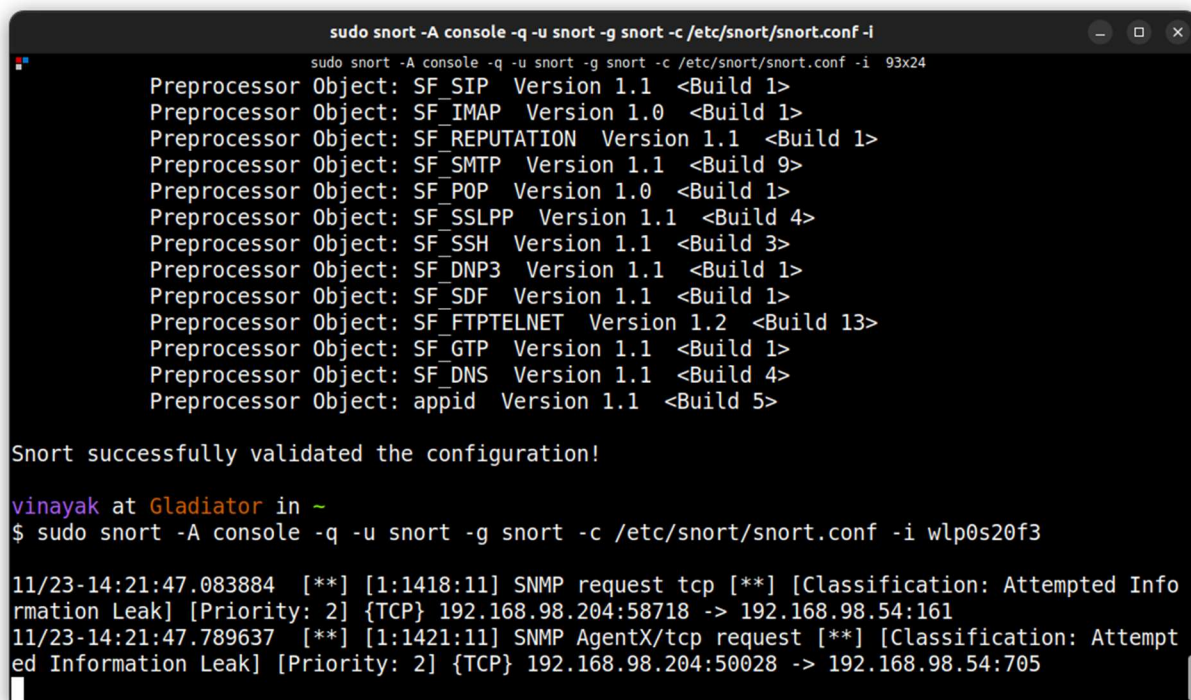
## Running NMAP from another device



```
dnyaneshwar@fedora ~/Downloads
$ nmap 192.168.98.54
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-23 14:21 IST
Nmap scan report for 192.168.98.54
Host is up (0.0088s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 16.70 seconds
dnyaneshwar@fedora ~/Downloads
$
```

Starting SNORT in detection mode:



```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i 93x24
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: appid Version 1.1 <Build 5>

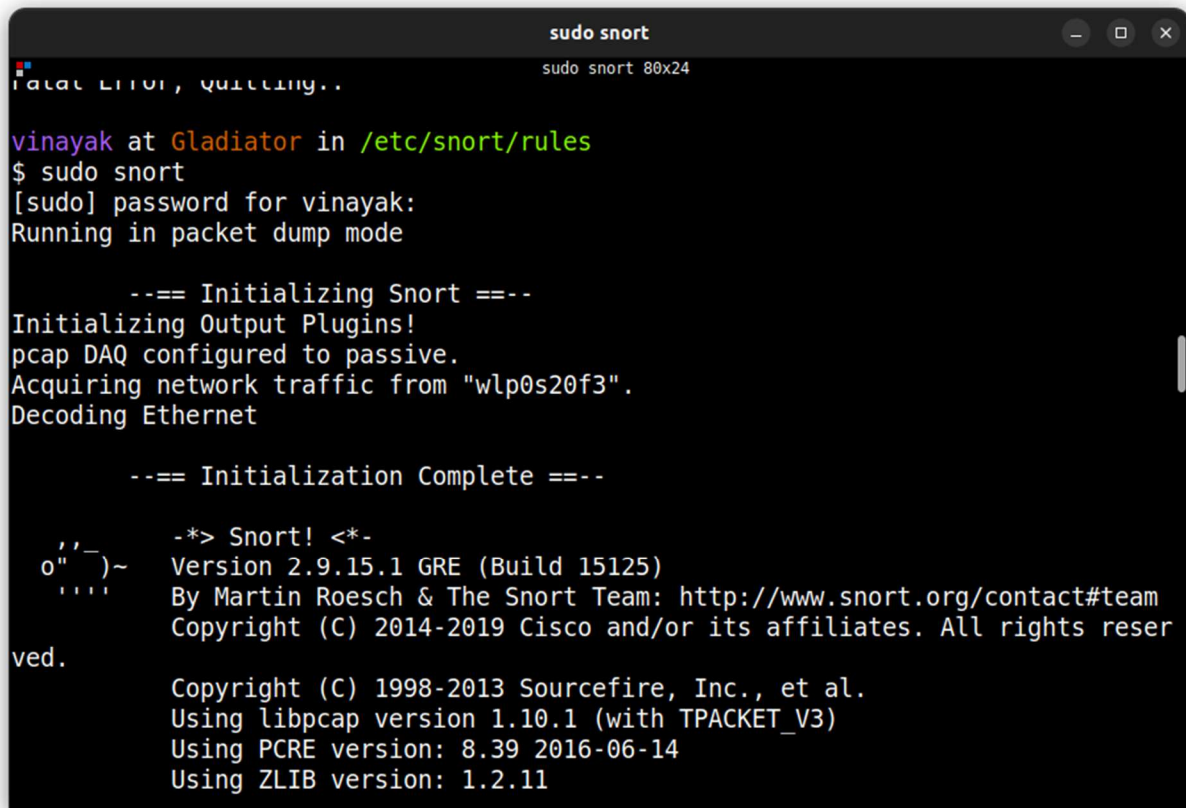
Snort successfully validated the configuration!

vinayak at Gladiator in ~
$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i wlp0s20f3

11/23-14:21:47.083884  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Info
rmation Leak] [Priority: 2] {TCP} 192.168.98.204:58718 -> 192.168.98.54:161
11/23-14:21:47.789637  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempt
ed Information Leak] [Priority: 2] {TCP} 192.168.98.204:50028 -> 192.168.98.54:705
```

The above screenshot shows that SNORT has successfully detected the NMAP attack and an alert has been generated

## Running snort in sniffing mode:



```
sudo snort
Fatal Error, quitting..
sudo snort 80x24

vinayak at Gladiator in /etc/snort/rules
$ sudo snort
[sudo] password for vinayak:
Running in packet dump mode

--== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "wlp0s20f3".
Decoding Ethernet

--== Initialization Complete ==--

    ,,_
o"  )~  -*> Snort! <*-
    ' '  Version 2.9.15.1 GRE (Build 15125)
          By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
          Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.

          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
```

Sniffing result:



```
vinayak@Gladiator:/etc/snort/rules
vinayak@Gladiator:/etc/snort/rules 80x40

=====
Run time for packet processing was 18.415611 seconds
Snort processed 443 packets.
Snort ran for 0 days 0 hours 0 minutes 18 seconds
  Pkts/sec:          24
=====

Memory usage summary:
  Total non-mmapped bytes (arena):      790528
  Bytes in mapped regions (hblkhd):     21590016
  Total allocated space (uordblks):     685808
  Total free space (fordblks):          104720
  Topmost releasable block (keepcost):  102624
=====

Packet I/O Totals:
  Received:          449
  Analyzed:          443 ( 98.664%)
  Dropped:           0 (  0.000%)
  Filtered:           0 (  0.000%)
  Outstanding:       6 (  1.336%)
  Injected:           0
=====

Breakdown by protocol (includes rebuilt packets):
  Eth:               443 (100.000%)
  VLAN:              0 (  0.000%)
  IP4:               97 ( 21.896%)
  Frag:              0 (  0.000%)
  ICMP:              0 (  0.000%)
  UDP:               4 (  0.903%)
  TCP:              93 ( 20.993%)
  IP6:              344 ( 77.652%)
  IP6 Ext:           344 ( 77.652%)
  IP6 Opts:          0 (  0.000%)
  Frag6:             0 (  0.000%)
  ICMP6:             2 (  0.451%)
  UDP6:             342 ( 77.201%)
  TCP6:             0 (  0.000%)
  Teredo:            0 (  0.000%)
  ICMP-IP:           0 (  0.000%)
  IP4/IP4:           0 (  0.000%)
  IP4/IP6:           0 (  0.000%)
  IP6/IP4:           0 (  0.000%)
```

```
vinayak@Gladiator:/etc/snort/rules
vinayak@Gladiator:/etc/snort/rules 80x40
UDP6:      342 ( 77.201%)
TCP6:      0 ( 0.000%)
Teredo:    0 ( 0.000%)
ICMP-IP:   0 ( 0.000%)
IP4/IP4:   0 ( 0.000%)
IP4/IP6:   0 ( 0.000%)
IP6/IP4:   0 ( 0.000%)
IP6/IP6:   0 ( 0.000%)
GRE:       0 ( 0.000%)
GRE Eth:   0 ( 0.000%)
GRE VLAN:  0 ( 0.000%)
GRE IP4:   0 ( 0.000%)
GRE IP6:   0 ( 0.000%)
GRE IP6 Ext: 0 ( 0.000%)
GRE PPTP:  0 ( 0.000%)
GRE ARP:   0 ( 0.000%)
GRE IPX:   0 ( 0.000%)
GRE Loop:  0 ( 0.000%)
MPLS:     0 ( 0.000%)
ARP:       2 ( 0.451%)
IPX:       0 ( 0.000%)
Eth Loop:  0 ( 0.000%)
Eth Disc:  0 ( 0.000%)
IP4 Disc:  0 ( 0.000%)
IP6 Disc:  0 ( 0.000%)
TCP Disc:  0 ( 0.000%)
UDP Disc:  0 ( 0.000%)
ICMP Disc: 0 ( 0.000%)
All Discard: 0 ( 0.000%)
Other:     0 ( 0.000%)
Bad Chk Sum: 271 ( 61.174%)
Bad TTL:   0 ( 0.000%)
S5 G 1:    0 ( 0.000%)
S5 G 2:    0 ( 0.000%)
Total:     443
=====
Snort exiting

vinayak at Gladiator in /etc/snort/rules
$
```

**Conclusion:**

Snort is referred to as a packet sniffer that monitors network traffic, scrutinizing each packet closely to detect a dangerous payload or suspicious anomalies.