

# CSEC 744 Network Security

Name : Shriram Karpoora Sundara Pandian Course

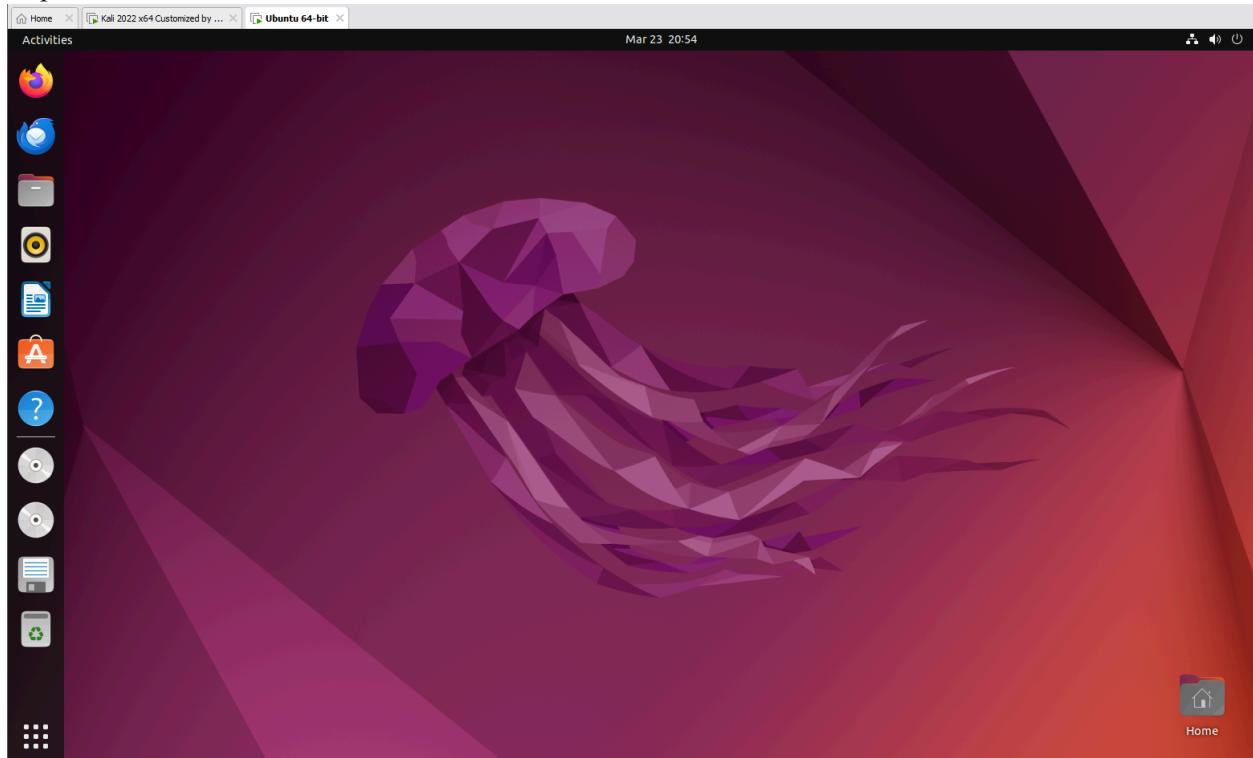
Title : Intrusion Detection Systems

Lab : 7

Chapter : 13 ( SecurityPlus )

## Exercise 13. 01 :

Step 1 :

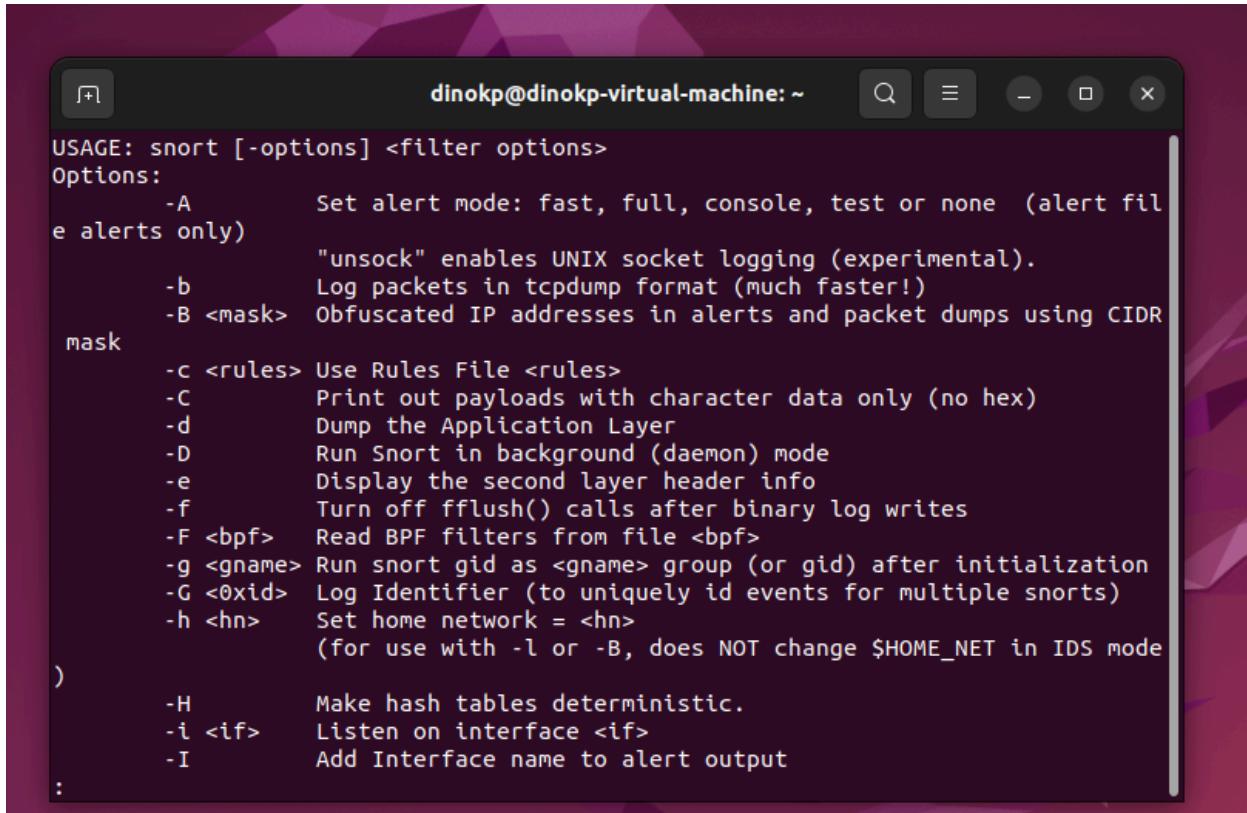


Step 2 :

```
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
dinokp@dinokp-virtual-machine:~$ sudo apt install snort
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libdaq2 libdumbnet1 libluajit-5.1-2 libluajit-5.1-common libnetfilter-queue1
    net-tools oinkmaster snort-common snort-common-libraries snort-rules-default
Suggested packages:
  snort-doc
The following NEW packages will be installed:
  libdaq2 libdumbnet1 libluajit-5.1-2 libluajit-5.1-common libnetfilter-queue1
    net-tools oinkmaster snort snort-common snort-common-libraries
      snort-rules-default
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,554 kB of archives.
After this operation, 11.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libluajit-5.1-com
mon all 2.1.0~beta3+dfsg-6 [44.3 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libluajit-5.1-2 a
md64 2.1.0~beta3+dfsg-6 [238 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 snort-common-libr
```

### Exercise 13. 02 :

A.



The screenshot shows a terminal window with the title bar "dinokp@dinokp-virtual-machine: ~". The window displays the usage information for the "snort" command. The text is as follows:

```
USAGE: snort [-options] <filter options>
Options:
  -A           Set alert mode: fast, full, console, test or none  (alert fil
e alerts only)
              "unsock" enables UNIX socket logging (experimental).
  -b           Log packets in tcpdump format (much faster!)
  -B <mask>   Obfuscated IP addresses in alerts and packet dumps using CIDR
mask
  -c <rules>  Use Rules File <rules>
  -C           Print out payloads with character data only (no hex)
  -d           Dump the Application Layer
  -D           Run Snort in background (daemon) mode
  -e           Display the second layer header info
  -f           Turn off fflush() calls after binary log writes
  -F <bpf>    Read BPF filters from file <bpf>
  -g <gname>   Run snort gid as <gname> group (or gid) after initialization
  -G <0xid>   Log Identifier (to uniquely id events for multiple snorts)
  -h <hn>     Set home network = <hn>
              (for use with -l or -B, does NOT change $HOME_NET in IDS mode
)
  -H           Make hash tables deterministic.
  -i <if>     Listen on interface <if>
  -I           Add Interface name to alert output
:
```

B.

The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "dinokp@dinokp-virtual-machine: ~". The window contains the following content:

**DESCRIPTION**

**Snort** is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plugin architecture. Snort also has a modular real-time alerting capability, incorporating alerting and logging plugins for syslog, a ASCII text files, UNIX sockets or XML.

Snort has three primary uses. It can be used as a straight packet sniffer like **tcpdump(1)**, a packet logger (useful for network traffic debugging, etc), or as a full blown network intrusion detection system.

Snort logs packets in **tcpdump(1)** binary format or in Snort's decoded ASCII format to a hierarchy of logging directories that are named based on the IP address of the "foreign" host.

**OPTIONS**

-A alert-mode

Manual page snort(8) line 31 (press h for help or q to quit)

C.

The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "dinokp@dinokp-virtual-machine: ~". The window contains the following content:

```
dinokp@dinokp-virtual-machine:~$ man snort
dinokp@dinokp-virtual-machine:~$ 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

dinokp@dinokp-virtual-machine:~$
```

Step 2 :

A.

```
 dinokp@dinokp-virtual-machine:~$ ip a
 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
 2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
        link/ether 00:0c:29:bf:2a:83 brd ff:ff:ff:ff:ff:ff
        altname enp2s1
        inet 192.168.32.136/24 brd 192.168.32.255 scope global dynamic noprefixroute
          ens33
            valid_lft 1417sec preferred_lft 1417sec
        inet6 fe80::81ac:978b:40a8:b288/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
 dinokp@dinokp-virtual-machine:~$
```

B.

```
Decoding Ethernet
      --- Initialization Complete ---
      -*> 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

Commencing packet processing (pid=18471)
WARNING: No preprocessors configured for policy 0.
03/23-21:14:38.902406 192.168.32.128:51590 -> 5.78.62.36:123
  UDP TTL:64 TOS:0x10 ID:52548 IpLen:20 DgmLen:76 DF
  Len: 48
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

WARNING: No preprocessors configured for policy 0.
03/23-21:14:38.984041 5.78.62.36:123 -> 192.168.32.128:51590
  UDP TTL:128 TOS:0x0 ID:33371 IpLen:20 DgmLen:76
```

C.

```
03/23-21:19:06.558191 192.168.32.128 -> 192.168.32.136
ICMP TTL:64 TOS:0x0 ID:44424 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:8420 Seq:7 ECHO
=+==+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

WARNING: No preprocessors configured for policy 0.
03/23-21:19:06.558233 192.168.32.136 -> 192.168.32.128
ICMP TTL:64 TOS:0x0 ID:22605 IpLen:20 DgmLen:84
Type:0 Code:0 ID:8420 Seq:7 ECHO REPLY
=+==+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

WARNING: No preprocessors configured for policy 0.
03/23-21:19:07.557912 192.168.32.128 -> 192.168.32.136
ICMP TTL:64 TOS:0x0 ID:44531 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:8420 Seq:8 ECHO
=+==+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

WARNING: No preprocessors configured for policy 0.
03/23-21:19:07.557954 192.168.32.136 -> 192.168.32.128
ICMP TTL:64 TOS:0x0 ID:22615 IpLen:20 DgmLen:84
Type:0 Code:0 ID:8420 Seq:8 ECHO REPLY
=+==+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

D.

```
03/23-21:21:04.067782 127.0.0.1 -> 127.0.0.1
ICMP TTL:64 TOS:0x0 ID:55758 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:1 Seq:5 ECHO
=+==+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

WARNING: No preprocessors configured for policy 0.
03/23-21:21:04.067799 127.0.0.1 -> 127.0.0.1
ICMP TTL:64 TOS:0x0 ID:55759 IpLen:20 DgmLen:84
Type:0 Code:0 ID:1 Seq:5 ECHO REPLY
=+==+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

WARNING: No preprocessors configured for policy 0.
03/23-21:21:05.091901 127.0.0.1 -> 127.0.0.1
ICMP TTL:64 TOS:0x0 ID:55860 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:1 Seq:6 ECHO
=+==+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

WARNING: No preprocessors configured for policy 0.
03/23-21:21:05.091928 127.0.0.1 -> 127.0.0.1
ICMP TTL:64 TOS:0x0 ID:55861 IpLen:20 DgmLen:84
Type:0 Code:0 ID:1 Seq:6 ECHO REPLY
=+==+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```



B.

C.

### **Exercise 13. 03 :**

Step 1:

A.

```
dinokp@dinokp-virtual-machine:~$ sudo snort -l .
Running in packet logging mode

      --- Initializing Snort ---
Initializing Output Plugins!
Log directory = .
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

      --- Initialization Complete ---

      -*> 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 rese
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

Commencing packet processing (pid=18998)
```

We got the snort is opening and especially the logging directory is equal to .

```
dinokp@dinokp-virtual-machine: ~/Desktop$ sudo snort -l log-dir
[sudo] password for dinokp:
Running in packet logging mode

     --- Initializing Snort ---
Initializing Output Plugins!
Log directory = log-dir
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

     --- Initialization Complete ---

      ,,-      -*> 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

Commencing packet processing (pid=19186)
```

Now i am trying to save the logs to log-dir in desktop and in next step i will ping with windows machine.

B. Pinging Ubuntu :

```
root@kali:~# ping 192.168.32.136
PING 192.168.32.136 (192.168.32.136) 56(84) bytes of data.
64 bytes from 192.168.32.136: icmp_seq=1 ttl=64 time=3.86 ms
64 bytes from 192.168.32.136: icmp_seq=2 ttl=64 time=1.82 ms
64 bytes from 192.168.32.136: icmp_seq=3 ttl=64 time=3.86 ms
64 bytes from 192.168.32.136: icmp_seq=4 ttl=64 time=3.11 ms
64 bytes from 192.168.32.136: icmp_seq=5 ttl=64 time=1.36 ms
64 bytes from 192.168.32.136: icmp_seq=6 ttl=64 time=1.54 ms
64 bytes from 192.168.32.136: icmp_seq=7 ttl=64 time=1.81 ms
64 bytes from 192.168.32.136: icmp_seq=8 ttl=64 time=2.10 ms
64 bytes from 192.168.32.136: icmp_seq=9 ttl=64 time=1.28 ms
64 bytes from 192.168.32.136: icmp_seq=10 ttl=64 time=3.71 ms
64 bytes from 192.168.32.136: icmp_seq=11 ttl=64 time=1.77 ms
64 bytes from 192.168.32.136: icmp_seq=12 ttl=64 time=1.52 ms
64 bytes from 192.168.32.136: icmp_seq=13 ttl=64 time=2.61 ms
64 bytes from 192.168.32.136: icmp_seq=14 ttl=64 time=2.01 ms
64 bytes from 192.168.32.136: icmp_seq=15 ttl=64 time=2.43 ms
64 bytes from 192.168.32.136: icmp_seq=16 ttl=64 time=1.35 ms
64 bytes from 192.168.32.136: icmp_seq=17 ttl=64 time=2.15 ms
64 bytes from 192.168.32.136: icmp_seq=18 ttl=64 time=1.47 ms
64 bytes from 192.168.32.136: icmp_seq=19 ttl=64 time=1.12 ms
64 bytes from 192.168.32.136: icmp_seq=20 ttl=64 time=1.31 ms
64 bytes from 192.168.32.136: icmp_seq=21 ttl=64 time=2.34 ms
=====
Run time for packet processing was 107.542077 seconds
Snort processed 223 packets.
Snort ran for 0 days 0 hours 1 minutes 47 seconds
    Pkts/min:          223
    Pkts/sec:          2
=====
Memory usage summary:
    Total non-mmapped bytes (arena):      790528
    Bytes in mapped regions (hblkhd):    21590016
    Total allocated space (uordblks):     687184
    Total free space (fordblks):         103344
    Topmost releasable block (keepcost):   99936
=====
Packet I/O Totals:
    Received:          225
    Analyzed:          223 ( 99.111%)
    Dropped:           0 ( 0.000%)
    Filtered:          0 ( 0.000%)
    Outstanding:       2 ( 0.889%)
    Injected:          0
=====
```

## Step 2 :

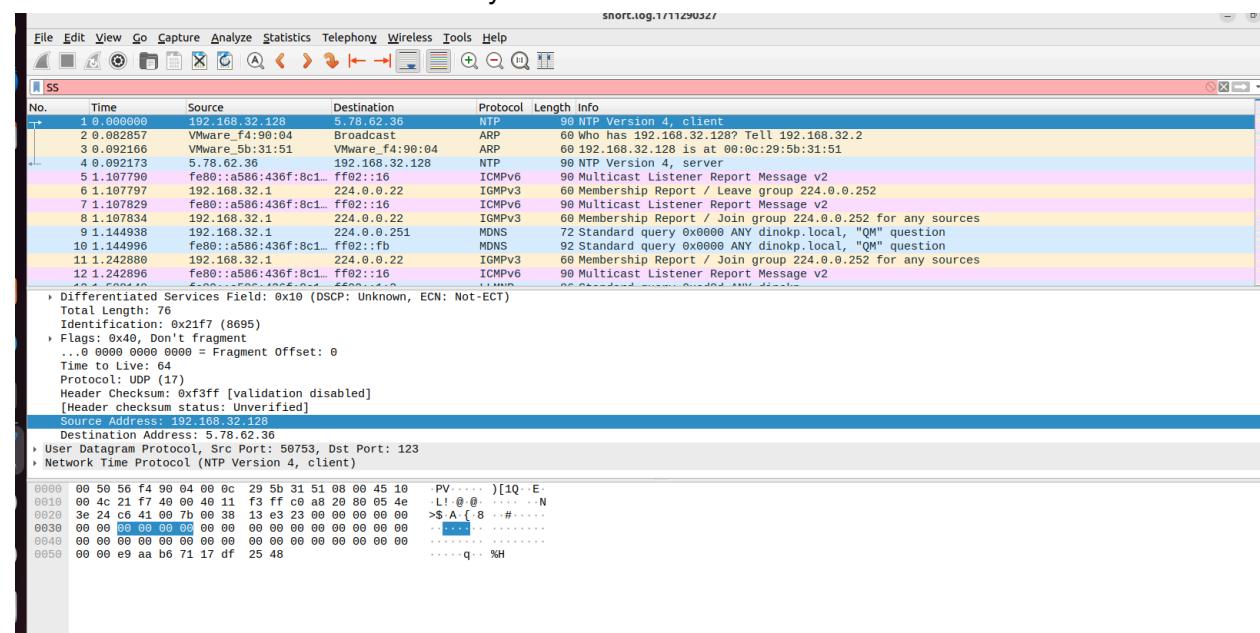
A.

Snort log is successfully saved :

```
l-machine:~/Desktop/log-dir$  
dinokp@dinokp-virtual-machine:~/Desktop/log-dir$ ls  
snort.log.1711290327  
dinokp@dinokp-virtual-machine:~/Desktop/log-dir$
```

B.

We can see the source address of my kali machine.



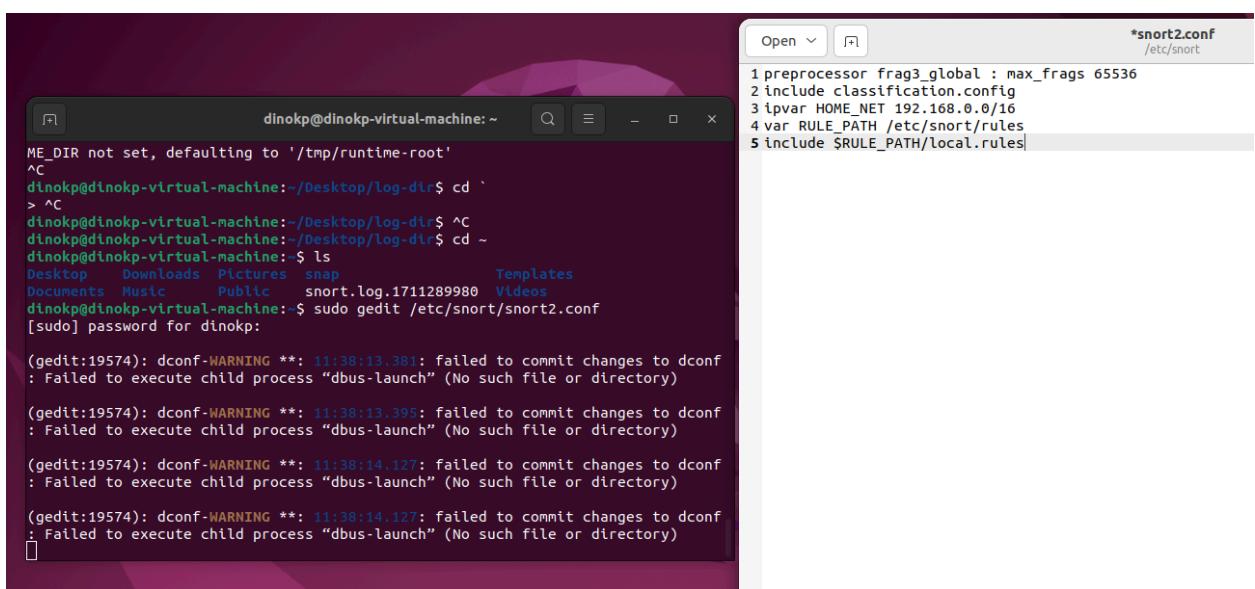
No.	Time	Source	Destination	Protocol	Length	Info
27	15.838768	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=1/256, ttl=64 (reply in 30)
30	15.948712	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=1/256, ttl=64 (request in 27)
31	16.948033	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=2/512, ttl=64 (reply in 32)
32	16.949130	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=2/512, ttl=64 (request in 31)
33	17.942364	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=3/768, ttl=64 (reply in 34)
34	17.942424	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=3/768, ttl=64 (request in 33)
35	18.942477	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=4/1024, ttl=64 (reply in 36)
36	18.942536	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=4/1024, ttl=64 (request in 35)
55	19.943547	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=5/1280, ttl=64 (reply in 56)
56	19.943579	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=5/1280, ttl=64 (request in 55)
57	20.946489	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=6/1536, ttl=64 (reply in 58)
58	20.946447	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=6/1536, ttl=64 (request in 57)
61	21.948296	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=7/1792, ttl=64 (reply in 62)
62	21.948344	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=7/1792, ttl=64 (request in 61)
63	22.949821	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=8/2048, ttl=64 (reply in 64)
64	22.949876	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=8/2048, ttl=64 (request in 63)
65	23.950748	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=9/2304, ttl=64 (reply in 66)
66	23.950782	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=9/2304, ttl=64 (request in 65)
67	24.954699	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=10/2560, ttl=64 (reply in 68)
68	24.954755	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=10/2560, ttl=64 (request in 67)
69	25.954659	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=11/2816, ttl=64 (reply in 70)
70	25.954659	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=11/2816, ttl=64 (request in 69)
71	26.955498	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=12/3072, ttl=64 (reply in 72)
72	26.955546	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=12/3072, ttl=64 (request in 71)
73	27.957966	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=13/3328, ttl=64 (reply in 74)
74	27.958223	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=13/3328, ttl=64 (request in 73)
75	28.961598	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=14/3584, ttl=64 (reply in 76)
76	28.961650	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=14/3584, ttl=64 (request in 75)
77	29.963698	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=15/3840, ttl=64 (reply in 78)
78	29.963755	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=15/3840, ttl=64 (request in 77)
79	30.965011	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=16/4096, ttl=64 (reply in 80)
80	30.965056	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=16/4096, ttl=64 (request in 79)
81	31.966340	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=17/4352, ttl=64 (reply in 82)
82	31.966383	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=17/4352, ttl=64 (request in 81)
85	32.968688	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=18/4608, ttl=64 (reply in 86)
86	32.968753	192.168.32.136	192.168.32.128	ICMP	98	Echo (ping) reply id=0xc5c0, seq=18/4608, ttl=64 (request in 85)
87	33.970451	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) request id=0xc5c0, seq=19/4864, ttl=64 (reply in 88)
88	33.970451	192.168.32.128	192.168.32.136	ICMP	98	Echo (ping) reply id=0xc5c0, seq=19/4864, ttl=64 (request in 87)

We can see the icmp request and replies above.

### Exercise 13. 04 :

Step 1 :

A.



```

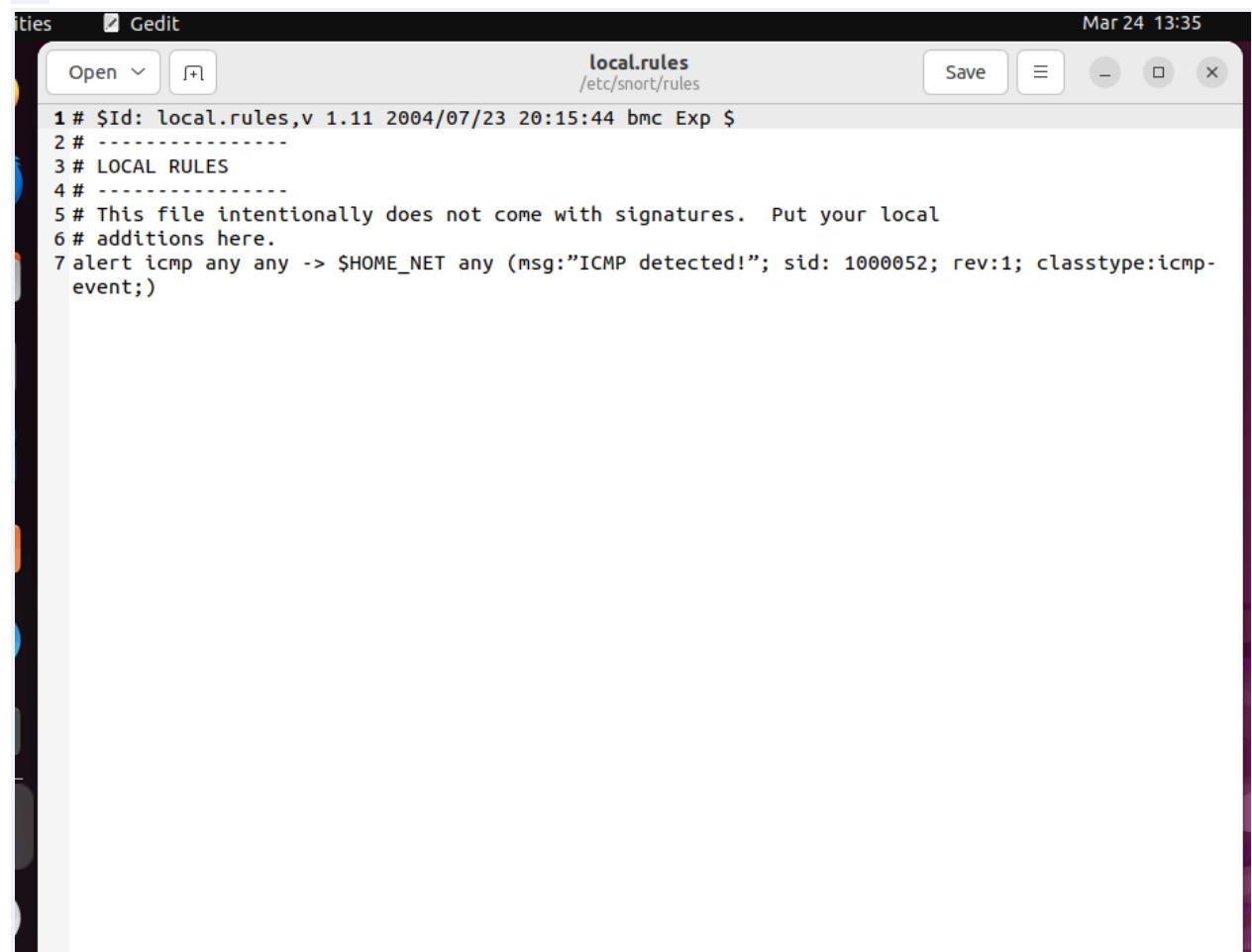
dino@dinokp-virtual-machine:~$ gedit /etc/snort/snort2.conf
[sudo] password for dino@dinokp-virtual-machine:
*snort2.conf
/etc/snort
1 preprocessor frag3_global : max_frags 65536
2 include classification.config
3 tpcvar HOME_NET 192.168.0.0/16
4 var RULE_PATH /etc/snort/rules
5 include $RULE_PATH/local.rules

ME_DIR not set, defaulting to '/tmp/runtime-root'
^C
dino@dinokp-virtual-machine:~/Desktop/log-dir$ cd `
```

Gedit config file :

```
gedit /etc/snort/classification.config
```

B.



The screenshot shows a Gedit text editor window with the following details:

- Title Bar:** Gedit
- File Path:** local.rules /etc/snort/rules
- Date/Time:** Mar 24 13:35
- Toolbar Buttons:** Open, Save, Minimize, Maximize, Close
- Text Content:**

```
1 # $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $  
2 # -----  
3 # LOCAL RULES  
4 # -----  
5 # This file intentionally does not come with signatures. Put your local  
6 # additions here.  
7 alert icmp any any -> $HOME_NET any (msg:"ICMP detected!"; sid: 1000052; rev:1; classtype:icmp-  
event;)
```

classification.config [Read-Only]  
/etc/snort

```
1 # $Id$  
2 # The following includes information for prioritizing rules  
3 #  
4 # Each classification includes a shortname, a description, and a default  
5 # priority for that classification.  
6 #  
7 # This allows alerts to be classified and prioritized. You can specify  
8 # what priority each classification has. Any rule can override the default  
9 # priority for that rule.  
10 #  
11 # Here are a few example rules:  
12 #  
13 #   alert TCP any any -> any 80 (msg: "EXPLOIT ntpdx overflow";  
14 #       dsize: > 128; classtype:attempted-admin; priority:10;  
15 #  
16 #   alert TCP any any -> any 25 (msg:"SMTP expn root"; flags:A+; \  
17 #           content:"expn root"; nocase; classtype:attempted-recon;)  
18 #  
19 # The first rule will set its type to "attempted-admin" and override  
20 # the default priority for that type.  
21 #  
22 # The second rule set its type to "attempted-recon" and set its  
23 # priority to the default for that type.  
24 #  
25 #  
26 #  
27 # config classification:shortname,short description,priority  
28 #  
29 #  
30 config classification: not-suspicious,Not Suspicious Traffic,3  
31 config classification: unknown,Unknown Traffic,3  
32 config classification: bad-unknown,Potentially Bad Traffic, 2  
33 config classification: attempted-recon,Attempted Information Leak,2  
34 config classification: successful-recon-limited,Information Leak,2  
35 config classification: successful-recon-largescale,Large Scale Information Leak,2  
36 config classification: attempted-dos,Attempted Denial of Service,2  
37 config classification: successful-dos,Denial of Service,2
```

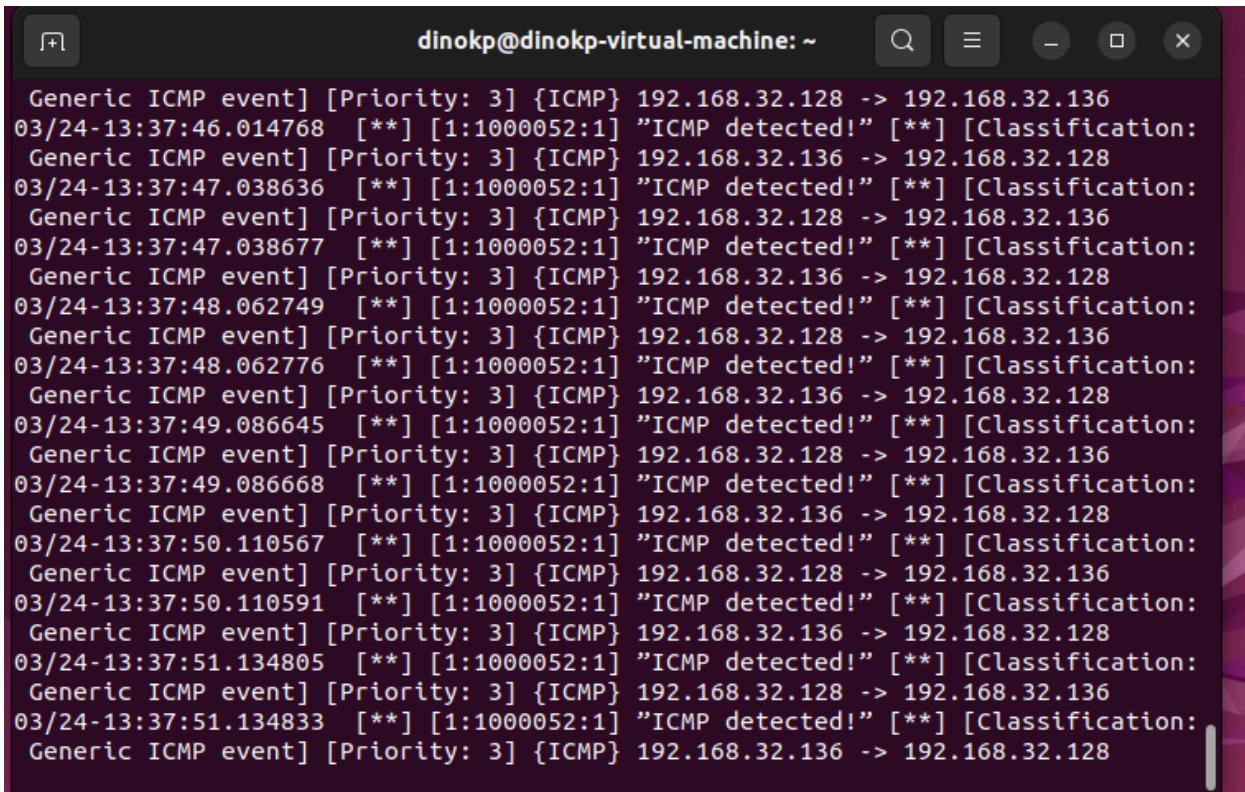
C.

```
sudo snort -A console -A fast -c /etc/snort/snort2.conf -i ens33
Running in IDS mode

      === Initializing Snort ===
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort2.conf"
Tagged Packet Limit: 256
Log directory = /var/log/snort
Frag3 global config:
  Max frags: 65536
  Fragment memory cap: 4194304 bytes

+++++
Initializing rule chains...
1 Snort rules read
  1 detection rules
  0 decoder rules
  0 preprocessor rules
1 Option Chains linked into 1 Chain Headers
+++++
```

D.



A screenshot of a terminal window titled "dinokp@dinokp-virtual-machine:~". The window displays a continuous stream of log messages from the Snort IDS. The messages are all identical, indicating "Generic ICMP event" with priority 3, ICMP type 0 (echo request), source 192.168.32.128, and destination 192.168.32.136. Each message is preceded by a timestamp and followed by "[\*\*] [1:1000052:1]" and the classification "ICMP detected!". The terminal has a dark theme with light-colored text and standard window controls at the top.

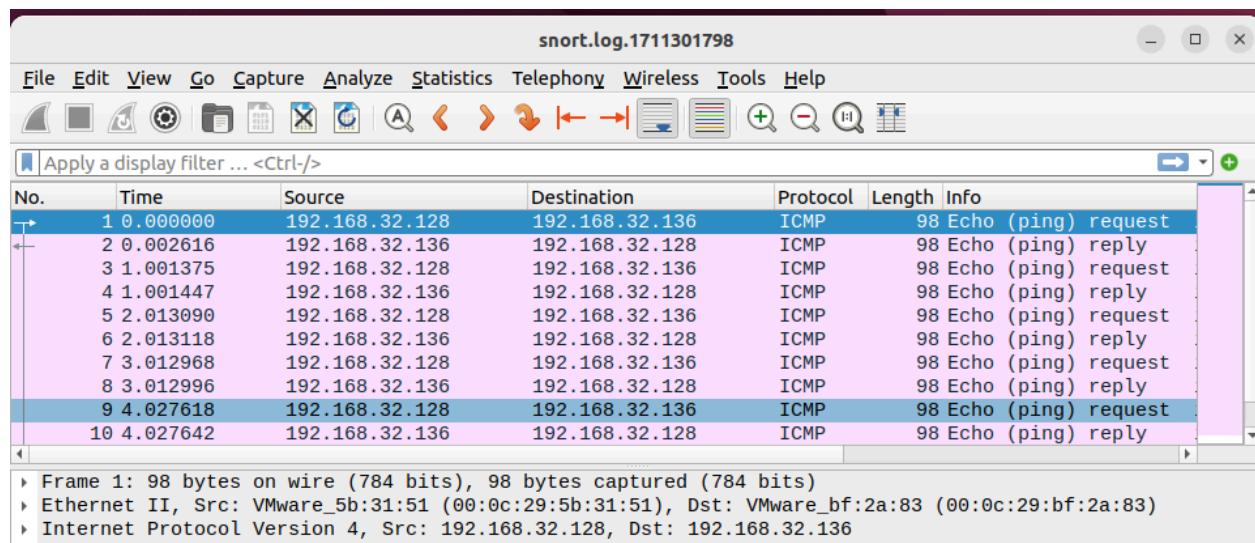
```
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:37:46.014768 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:37:47.038636 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:37:47.038677 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:37:48.062749 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:37:48.062776 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:37:49.086645 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:37:49.086668 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:37:50.110567 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:37:50.110591 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:37:51.134805 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:37:51.134833 [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
```

```
=====
Packet I/O Totals:
  Received:          80
  Analyzed:         79 ( 98.750%)
  Dropped:           0 ( 0.000%)
  Filtered:          0 ( 0.000%)
Outstanding:        1 ( 1.250%)
  Injected:          0
```

E.

```
-----  
Snort exiting  
dinokp@dinokp-virtual-machine:~$ sudo ls -l /var/log/snort  
total 588  
-rw-r--r-- 1 root adm 5247 Mar 24 13:37 alert  
-rw-r----- 1 snort adm 53820 Mar 24 13:38 snort.alert  
-rw-r--r-- 1 root adm 168611 Mar 24 13:38 snort.alert.fast  
-rw-r----- 1 snort adm 345028 Mar 24 13:38 snort.log  
-rw----- 1 root adm 3786 Mar 24 13:37 snort.log.1711301798  
dinokp@dinokp-virtual-machine:~$
```

F.



G.

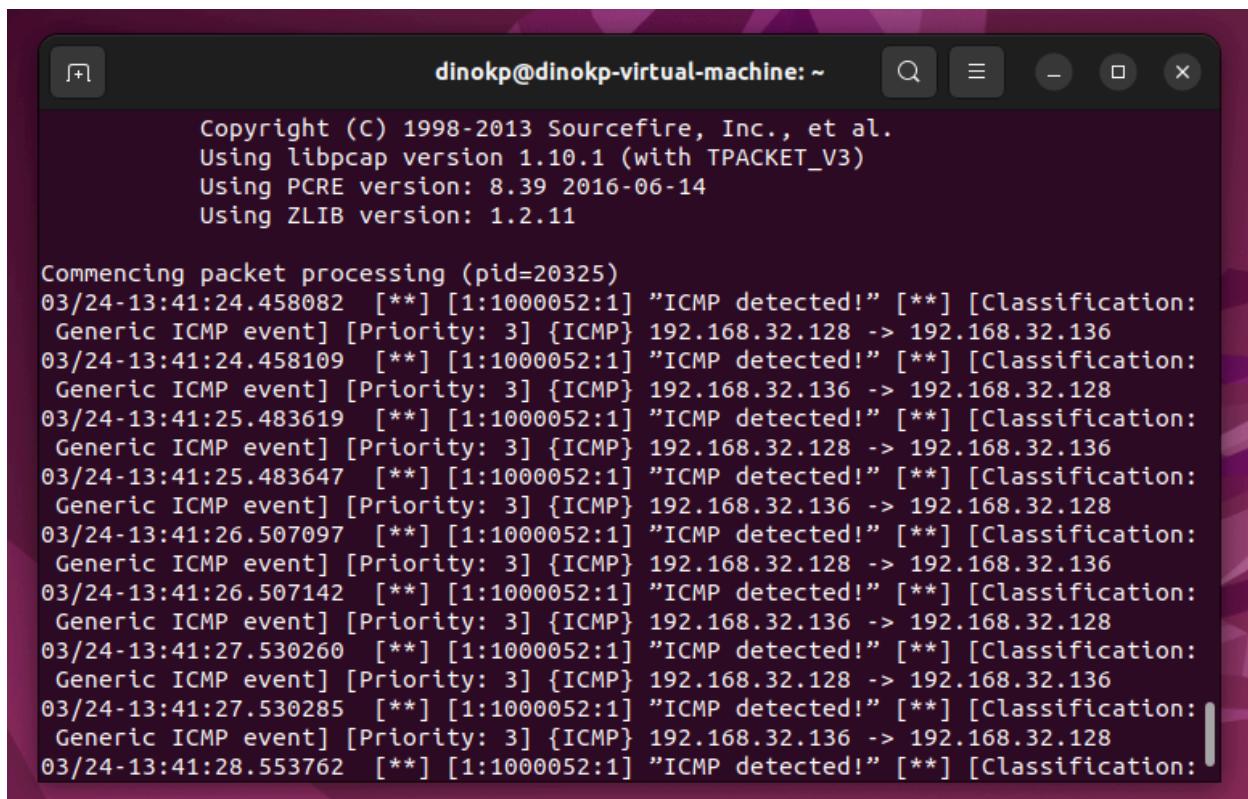
Open alert /var/log/snort

Save

```
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
16 03/24-13:37:49.086668 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
17 03/24-13:37:50.110567 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
18 03/24-13:37:50.110591 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
19 03/24-13:37:51.134805 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
20 03/24-13:37:51.134833 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
21 03/24-13:37:52.158563 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
22 03/24-13:37:52.158593 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
23 03/24-13:37:53.160275 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
24 03/24-13:37:53.160302 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
25 03/24-13:37:54.161789 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
26 03/24-13:37:54.161816 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
27 03/24-13:37:55.167384 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
28 03/24-13:37:55.167413 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
29 03/24-13:37:56.191941 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
30 03/24-13:37:56.191972 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
31 03/24-13:37:57.215913 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
32 03/24-13:37:57.215938 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
33 03/24-13:37:58.240307 [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP
event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
```

Plain Text Tab Width: 8 Line 1 Col 1 INIC

H.



A screenshot of a terminal window titled "dinokp@dinokp-virtual-machine: ~". The window displays network traffic analysis output. At the top, it shows copyright information: "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", and "Using ZLIB version: 1.2.11". Below this, the message "Commencing packet processing (pid=20325)" is displayed. The main content is a log of ICMP detection events from March 24, 2013, at 13:41:24. The log shows multiple entries for ICMP detected events between two hosts, with source and destination IP addresses being 192.168.32.128 and 192.168.32.136 respectively, and various timestamps and priority levels (Priority: 3).

```
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

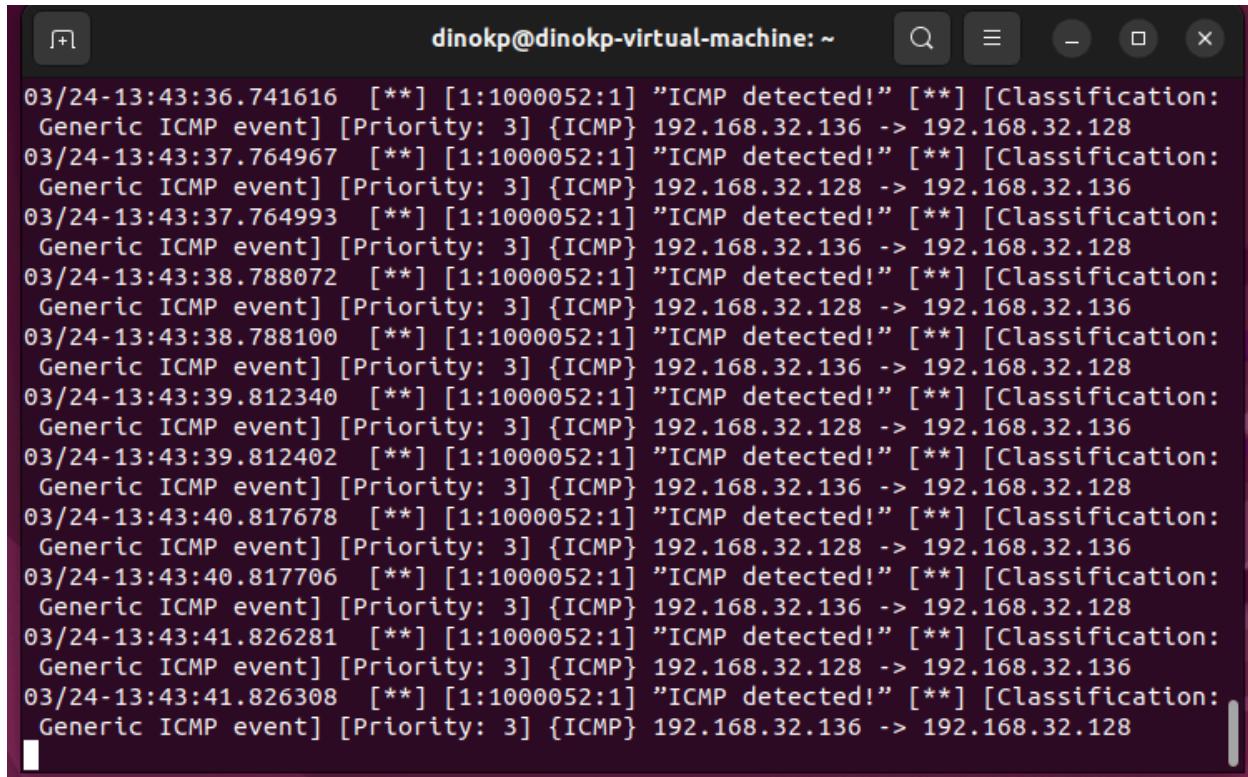
Commencing packet processing (pid=20325)
03/24-13:41:24.458082  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:41:24.458109  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:41:25.483619  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:41:25.483647  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:41:26.507097  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:41:26.507142  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:41:27.530260  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:41:27.530285  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:41:28.553762  [**] [1:1000052:1] "ICMP detected!" [**] [Classification:
```

I.

```
Open + alert /var/log/snort Save ⌂ ⌃ ⌄ ⌅
56 [Classification: Generic ICMP event] [Priority: 3]
57 03/24-13:41:28.553789 192.168.32.136 -> 192.168.32.128
58 ICMP TTL:64 TOS:0x0 ID:18800 IpLen:20 DgmLen:84
59 Type:0 Code:0 ID:34115 Seq:224 ECHO REPLY
60
61 [**] [1:1000052:1] "ICMP detected!" [**]
62 [Classification: Generic ICMP event] [Priority: 3]
63 03/24-13:41:29.577378 192.168.32.128 -> 192.168.32.136
64 ICMP TTL:64 TOS:0x0 ID:22609 IpLen:20 DgmLen:84 DF
65 Type:8 Code:0 ID:34115 Seq:225 ECHO
66
67 [**] [1:1000052:1] "ICMP detected!" [**]
68 [Classification: Generic ICMP event] [Priority: 3]
69 03/24-13:41:29.577404 192.168.32.136 -> 192.168.32.128
70 ICMP TTL:64 TOS:0x0 ID:18958 IpLen:20 DgmLen:84
71 Type:0 Code:0 ID:34115 Seq:225 ECHO REPLY
72
73 [**] [1:1000052:1] "ICMP detected!" [**]
74 [Classification: Generic ICMP event] [Priority: 3]
75 03/24-13:41:30.601004 192.168.32.128 -> 192.168.32.136
76 ICMP TTL:64 TOS:0x0 ID:22763 IpLen:20 DgmLen:84 DF
77 Type:8 Code:0 ID:34115 Seq:226 ECHO
78
79 [**] [1:1000052:1] "ICMP detected!" [**]
80 [Classification: Generic ICMP event] [Priority: 3]
81 03/24-13:41:30.601032 192.168.32.136 -> 192.168.32.128
82 ICMP TTL:64 TOS:0x0 ID:19197 IpLen:20 DgmLen:84
83 Type:0 Code:0 ID:34115 Seq:226 ECHO REPLY
84
85 [**] [1:1000052:1] "ICMP detected!" [**]
86 [Classification: Generic ICMP event] [Priority: 3]
87 03/24-13:41:31.624255 192.168.32.128 -> 192.168.32.136
88 ICMP TTL:64 TOS:0x0 ID:22908 IpLen:20 DgmLen:84 DF
89 Type:8 Code:0 ID:34115 Seq:227 ECHO
90
91 [**] [1:1000052:1] "ICMP detected!" [**]
92 [Classification: Generic ICMP event] [Priority: 3]
93 03/24-13:41:31.624284 192.168.32.136 -> 192.168.32.128
```

Plain Text ▾ Tab Width: 8 ▾ In 1 Col 1 ▾ INC

J.



A screenshot of a terminal window titled "dinokp@dinokp-virtual-machine: ~". The window contains a log of ICMP detection events. The log entries are as follows:

```
03/24-13:43:36.741616  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:43:37.764967  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:43:37.764993  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:43:38.788072  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:43:38.788100  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:43:39.812340  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:43:39.812402  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:43:40.817678  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:43:40.817706  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-13:43:41.826281  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-13:43:41.826308  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
```

K.



The screenshot shows a terminal window with the following details:

- Title Bar:** alert /var/log/snort
- Buttons:** Open, Save, Minimize, Maximize, Close
- Text Content:** A log of ICMP events. Lines 30 through 66 are shown, each starting with a line number and followed by an alert message. The messages indicate ICMP detected events between two hosts, with details like timestamp, source IP, destination IP, TTL, TOS, ID, length, and sequence numbers.
- Bottom Status Bar:** Plain Text, Tab Width: 8, Ln 19. Col 29, INS

```
30
31 [**] [1:1000052:1] "ICMP detected!" [**]
32 [Classification: Generic ICMP event] [Priority: 3]
33 03/24-13:43:37.764993 192.168.32.136 -> 192.168.32.128
34 ICMP TTL:64 TOS:0x0 ID:35292 IpLen:20 DgmLen:84
35 Type:0 Code:0 ID:34115 Seq:351 ECHO REPLY
36
37 [**] [1:1000052:1] "ICMP detected!" [**]
38 [Classification: Generic ICMP event] [Priority: 3]
39 03/24-13:43:38.788072 192.168.32.128 -> 192.168.32.136
40 ICMP TTL:64 TOS:0x0 ID:39271 IpLen:20 DgmLen:84 DF
41 Type:8 Code:0 ID:34115 Seq:352 ECHO
42
43 [**] [1:1000052:1] "ICMP detected!" [**]
44 [Classification: Generic ICMP event] [Priority: 3]
45 03/24-13:43:38.788100 192.168.32.136 -> 192.168.32.128
46 ICMP TTL:64 TOS:0x0 ID:35371 IpLen:20 DgmLen:84
47 Type:0 Code:0 ID:34115 Seq:352 ECHO REPLY
48
49 [**] [1:1000052:1] "ICMP detected!" [**]
50 [Classification: Generic ICMP event] [Priority: 3]
51 03/24-13:43:39.812340 192.168.32.128 -> 192.168.32.136
52 ICMP TTL:64 TOS:0x0 ID:39415 IpLen:20 DgmLen:84 DF
53 Type:8 Code:0 ID:34115 Seq:353 ECHO
54
55 [**] [1:1000052:1] "ICMP detected!" [**]
56 [Classification: Generic ICMP event] [Priority: 3]
57 03/24-13:43:39.812402 192.168.32.136 -> 192.168.32.128
58 ICMP TTL:64 TOS:0x0 ID:35544 IpLen:20 DgmLen:84
59 Type:0 Code:0 ID:34115 Seq:353 ECHO REPLY
60
61 [**] [1:1000052:1] "ICMP detected!" [**]
62 [Classification: Generic ICMP event] [Priority: 3]
63 03/24-13:43:40.817678 192.168.32.128 -> 192.168.32.136
64 ICMP TTL:64 TOS:0x0 ID:39434 IpLen:20 DgmLen:84 DF
65 Type:8 Code:0 ID:34115 Seq:354 ECHO
66
```



C. Which five categories are the most interesting?

The most interesting categories, I felt are follows :

- Backdoor.rules : As steganography and phishing emails are growing I found it as interesting and useful one.
- Attack\_responses rules : It is very important to manage the attack and manage it, so this is the main one.
- Dns.rules : It is mandatory for any company
- DDOS rules : Especially for financial services this is very important and interesting one i felt.
- Finally ICMP rules : This is great one, while someone tries to ping our server, having this is very important likewise

D. Which five individual rules are the most interesting?

- alert icmp any any -> any any (msg:"ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited"; icode:10; itype:3; classtype:misc-activity; sid:486; rev:4;)
- alert tcp \$EXTERNAL\_NET any -> \$HOME\_NET 53 (msg:"DNS TCP inverse query"; flow:to\_server,established; byte\_test:1,<,16,2; byte\_test:1,&,8,2; reference:bugtraq,2302; reference:cve,2001-0010; classtype:attempted-recon; sid:2922; rev:1;)
- alert tcp \$EXTERNAL\_NET any -> \$TELNET\_SERVERS 23 (msg:"BACKDOOR MISC Linux rootkit attempt"; flow:to\_server,established; content:"wh00t!"; classtype:attempted-admin; sid:213; rev:4;)
- alert tcp \$HOME\_NET any -> \$EXTERNAL\_NET 25 (msg:"VIRUS OUTBOUND bad file attachment"; flow:to\_server,established; content:"Content-Disposition|3A|"; nocase; pcre:"/filename\s\*=\s\*.\*?\.(?=[abcdehijlmnoprsvwx])(a(d[ep]|s[dfx])|c([ho]m|l|md|p|d|iz|l|ot)|e(m[f]|xe)|h(lp|sq|ta)|jse?|m(d[abew]|s[ip])|p(p[st]|if|[lm]|ot)|r(eg|tf)|s(cr|f|n|s|u|v|w|z|\_|)|t(e|m|n|o|p|s|u|v|w|z|\_|)|y(e|m|n|o|p|s|u|v|w|z|\_|)|z(e|m|n|o|p|s|u|v|w|z|\_|)/"); rev:1;)

```
|[hy]s|wf)|v(b[es]?|cf|xd)|w(m[dfsz]|p[dmsz]|s[cfh])|x|[tw]|bat|ini|lnk|nws|ocx)|\x27\x  
22\n\r\s]/iR"; classtype:suspicious-filename-detect; sid:721; rev:8;)
```

- alert icmp \$EXTERNAL\_NET any -> \$HOME\_NET any (msg:"DDOS  
Stacheldraht client spoofworks"; icmp\_id:1000; itype:0; content:"spoofworks";  
reference:arachnids,192; classtype:attempted-dos; sid:227; rev:6;)

These five rules from different categories I feel are interesting because they are very important rules for those particular categories, depending on them.

Step 3 :

A.

```
dinokp@dinokp-Virtual-Machine: ~$ sudo /etc/snort/rules/temp.rules  
dinokp@dinokp-Virtual-Machine: ~$ sudo cp /etc/snort/snort.conf /etc/snort/snort3  
.conf  
[sudo] password for dinokp:  
dinokp@dinokp-Virtual-Machine: ~$
```

B

The screenshot shows a terminal window with a dark theme. In the background, there are several lines of text from a dconf dump command, including:

```
1 root root 36661
1 root root 1437
okp-virtual-machine
okp-virtual-machine

sword for dinokp:
okp-virtual-machine

48): dconf-WARNING : o execute child proc...
```

In the foreground, a text editor window titled "snort3.conf" is open. The file contains Snort configuration code. A specific line, "65 ipvar HOME\_NET 192.168.0.0/16", is highlighted in red, indicating it is selected or being edited.

```
42 # for a specific interface
43 ##### If you want to run Snort in Debian using different
44 #
45 # instances each handling a different interface and
46 # a different configuration you can copy this file to
47 # /etc/snort/snort.$interface.conf (where '$interface' is the name of your
48 # network interface) and adjust the value there.
49 #
50 #
51 # The Debian init.d script is defined in such a way
52 # that you can run multiple instances.
53 #
54 #####
55 # Step #1: Set the network variables. For more information, see README.variables
56 #####
57 #
58 # Setup the network addresses you are protecting
59 #
60 # Note to Debian users: this value is overridden when starting
61 # up the Snort daemon through the init.d script by the
62 # value of DEBIAN_SNORT_HOME_NET is defined in the
63 # /etc/snort/snort.debian.conf configuration file
64 #
65 ipvar HOME_NET 192.168.0.0/16
66 #
67 # Set up the external network addresses. Leave as "any" in most situations
68 ipvar EXTERNAL_NET any
69 # If HOME_NET is defined as something other than "any", alternative, you can
70 # use this definition if you do not want to detect attacks from your internal
71 # IP addresses:
72 #ipvar EXTERNAL_NET !$HOME_NET
73 #
74 # List of DNS servers on your network
75 ipvar DNS_SERVERS $HOME_NET
76 #
77 # List of SMTP servers on your network
78 ipvar SMTP_SERVERS $HOME_NET
79
```

At the bottom of the editor window, there are status indicators: "Plain Text" (dropdown), "Tab Width: 8" (dropdown), "Ln 65. Col 1" (text), and "INS" (text).

C.

```
root@kali: ~
64 bytes from 192.168.32.136: icmp_seq=3375 ttl=64 time=1.08 ms
64 bytes from 192.168.32.136: icmp_seq=3376 ttl=64 time=1.07 ms
64 bytes from 192.168.32.136: icmp_seq=3377 ttl=64 time=2.39 ms
64 bytes from 192.168.32.136: icmp_seq=3378 ttl=64 time=6.50 ms
64 bytes from 192.168.32.136: icmp_seq=3379 ttl=64 time=1.87 ms
64 bytes from 192.168.32.136: icmp_seq=3380 ttl=64 time=1.08 ms
64 bytes from 192.168.32.136: icmp_seq=3381 ttl=64 time=1.05 ms
64 bytes from 192.168.32.136: icmp_seq=3382 ttl=64 time=1.26 ms
64 bytes from 192.168.32.136: icmp_seq=3383 ttl=64 time=2.30 ms
64 bytes from 192.168.32.136: icmp_seq=3384 ttl=64 time=1.77 ms
64 bytes from 192.168.32.136: icmp_seq=3385 ttl=64 time=0.965 ms
64 bytes from 192.168.32.136: icmp_seq=3386 ttl=64 time=1.09 ms
64 bytes from 192.168.32.136: icmp_seq=3387 ttl=64 time=0.803 ms
64 bytes from 192.168.32.136: icmp_seq=3388 ttl=64 time=1.01 ms
64 bytes from 192.168.32.136: icmp_seq=3389 ttl=64 time=1.59 ms
64 bytes from 192.168.32.136: icmp_seq=3390 ttl=64 time=1.74 ms
64 bytes from 192.168.32.136: icmp_seq=3391 ttl=64 time=2.17 ms
64 bytes from 192.168.32.136: icmp_seq=3392 ttl=64 time=1.87 ms
64 bytes from 192.168.32.136: icmp_seq=3393 ttl=64 time=2.25 ms
64 bytes from 192.168.32.136: icmp_seq=3394 ttl=64 time=1.95 ms
64 bytes from 192.168.32.136: icmp_seq=3395 ttl=64 time=1.02 ms
64 bytes from 192.168.32.136: icmp_seq=3396 ttl=64 time=1.91 ms
```

```
dinokp@dinokp-virtual-machine: ~
03/24-18:11:28.243580  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-18:11:29.246316  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-18:11:29.246347  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-18:11:30.249702  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-18:11:30.249756  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-18:11:31.252899  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-18:11:31.252936  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-18:11:32.255748  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-18:11:32.255796  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
03/24-18:11:33.267931  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.128 -> 192.168.32.136
03/24-18:11:33.267989  [**] [1:1000052:1] "ICMP detected!" [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.32.136 -> 192.168.32.128
```

D.

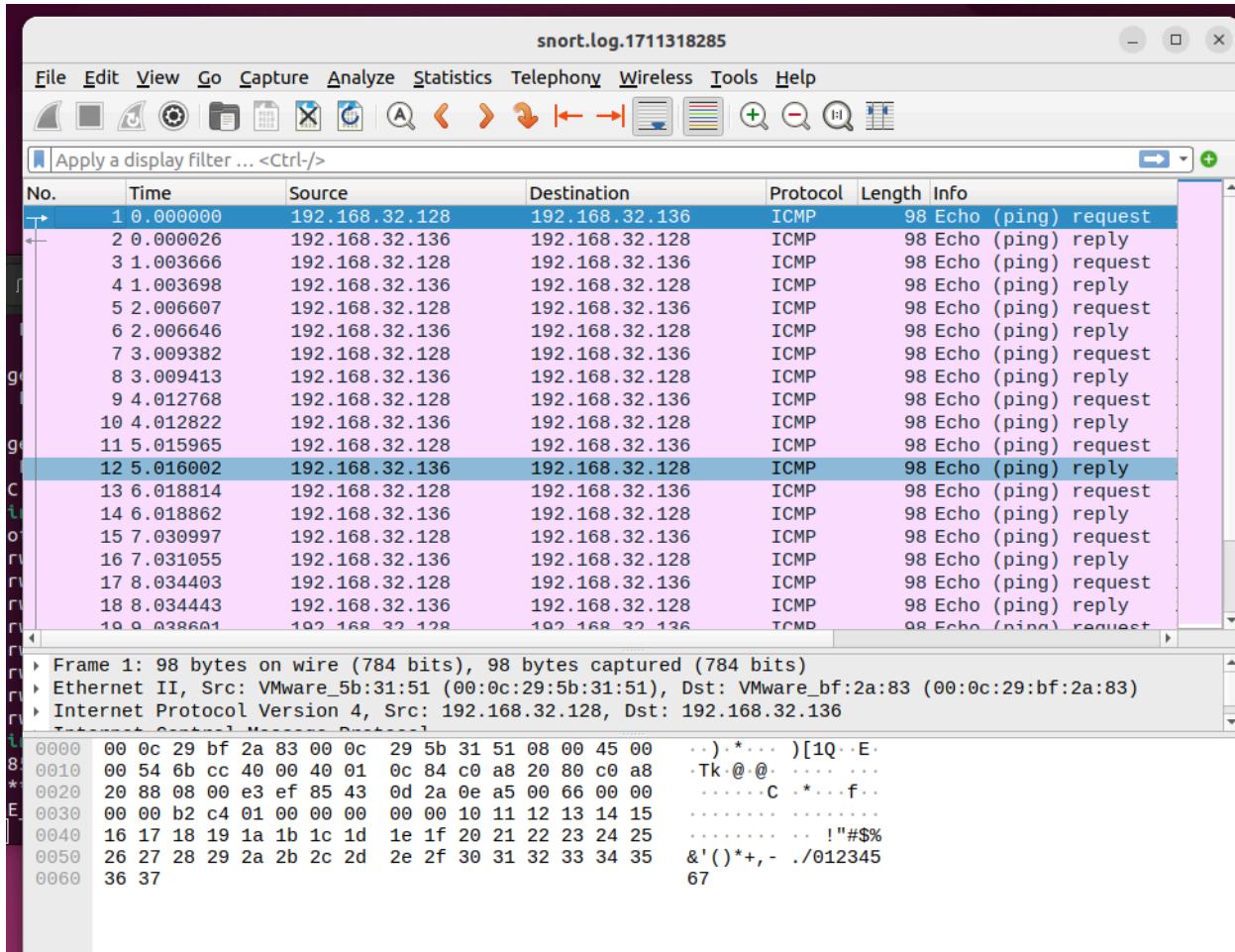
The screenshot shows a terminal window titled "alert" with the path "/var/log/snort". The window contains a list of 37 ICMP detection alerts. Each alert entry includes a line number, a double asterisk (\*\*), a timestamp, a classification ("Generic ICMP event"), a priority (3), and detailed network information (source IP 192.168.32.128, destination IP 192.168.32.136, TTL 64, TOS 0x0, ID 34115, sequence numbers 349 and 350, and type 8). The alerts are grouped by source IP and sequence number, indicating a two-way ICMP echo exchange between the two hosts.

```
1 [**] [1:1000052:1] "ICMP detected!" [**]
2 [Classification: Generic ICMP event] [Priority: 3]
3 03/24-13:43:35.718451 192.168.32.128 -> 192.168.32.136
4 ICMP TTL:64 TOS:0x0 ID:38836 IpLen:20 DgmLen:84 DF
5 Type:8 Code:0 ID:34115 Seq:349 ECHO
6
7 [**] [1:1000052:1] "ICMP detected!" [**]
8 [Classification: Generic ICMP event] [Priority: 3]
9 03/24-13:43:35.718477 192.168.32.136 -> 192.168.32.128
10 ICMP TTL:64 TOS:0x0 ID:34872 IpLen:20 DgmLen:84
11 Type:0 Code:0 ID:34115 Seq:349 ECHO REPLY
12
13 [**] [1:1000052:1] "ICMP detected!" [**]
14 [Classification: Generic ICMP event] [Priority: 3]
15 03/24-13:43:36.741591 192.168.32.128 -> 192.168.32.136
16 ICMP TTL:64 TOS:0x0 ID:38919 IpLen:20 DgmLen:84 DF
17 Type:8 Code:0 ID:34115 Seq:350 ECHO
18
19 [**] [1:1000052:1] "ICMP detected!" [**]
20 [Classification: Generic ICMP event] [Priority: 3]
21 03/24-13:43:36.741616 192.168.32.136 -> 192.168.32.128
22 ICMP TTL:64 TOS:0x0 ID:35055 IpLen:20 DgmLen:84
23 Type:0 Code:0 ID:34115 Seq:350 ECHO REPLY
24
25 [**] [1:1000052:1] "ICMP detected!" [**]
26 [Classification: Generic ICMP event] [Priority: 3]
27 03/24-13:43:37.764967 192.168.32.128 -> 192.168.32.136
28 ICMP TTL:64 TOS:0x0 ID:39111 IpLen:20 DgmLen:84 DF
29 Type:8 Code:0 ID:34115 Seq:351 ECHO
30
31 [**] [1:1000052:1] "ICMP detected!" [**]
32 [Classification: Generic ICMP event] [Priority: 3]
33 03/24-13:43:37.764993 192.168.32.136 -> 192.168.32.128
34 ICMP TTL:64 TOS:0x0 ID:35292 IpLen:20 DgmLen:84
35 Type:0 Code:0 ID:34115 Seq:351 ECHO REPLY
36
37 [**] [1:1000052:1] "ICMP detected!" [**]
```

E.

The screenshot shows a terminal window with the command "ls -l /var/log/snort" run by user "dinokp" on a virtual machine. The output lists several log files with their permissions, sizes, and timestamps. The files include "alert", "snort.alert", "snort.alert.fast", "snort.log", "snort.log.1711301798", "snort.log.1711302083", "snort.log.1711302214", and "snort.log.1711318285".

```
dinokp@dinokp-virtual-machine:~$ ls -l /var/log/snort
total 820
-rw-r--r-- 1 root adm 30102 Mar 24 18:12 alert
-rw-r----- 1 snort adm 72684 Mar 24 18:14 snort.alert
-rw-r--r-- 1 root adm 224539 Mar 24 18:14 snort.alert.fast
-rw-r----- 1 snort adm 460780 Mar 24 18:14 snort.log
-rw----- 1 root adm 3786 Mar 24 13:37 snort.log.1711301798
-rw----- 1 root adm 7662 Mar 24 13:41 snort.log.1711302083
-rw----- 1 root adm 3672 Mar 24 13:43 snort.log.1711302214
-rw----- 1 root adm 10284 Mar 24 18:12 snort.log.1711318285
```



## Lab Analysis :

- Three modes include Sniffer mode, which simply shows the packets from the stream, next is Packet Logger mode, which will save the logs of the packets. Finally Network Intrusion Detection mode, which helps to analyze the network packets and traffic.
- Most often used mode is obviously the NIDS mode, which is used to monitor the network and stop the malicious network based on the rules configured. This NIDS from snort will be combined with security infrastructure of the companies to give better protection, like working along with SIEM tools, firewalls, Network Monitoring tools etc. In this mode we can write our custom rule to prevent the traffic for safeguarding the network.
- Categories of rules snort use :
  - Web based attacks : Rules that will be used to prevent url attacks and hex modification on urls and browser attacks
  - Malware : Has some signature of the malwares to find it and stop their activity.

- DDOS : To prevent ddos attacks, this rule can identify and monitors the network the activity of the DDOS attacks and multiple request from lot of different variation to find the pattern.

### **KEY TERM QUIZ**

1. Rules
2. Configuration
3. Log