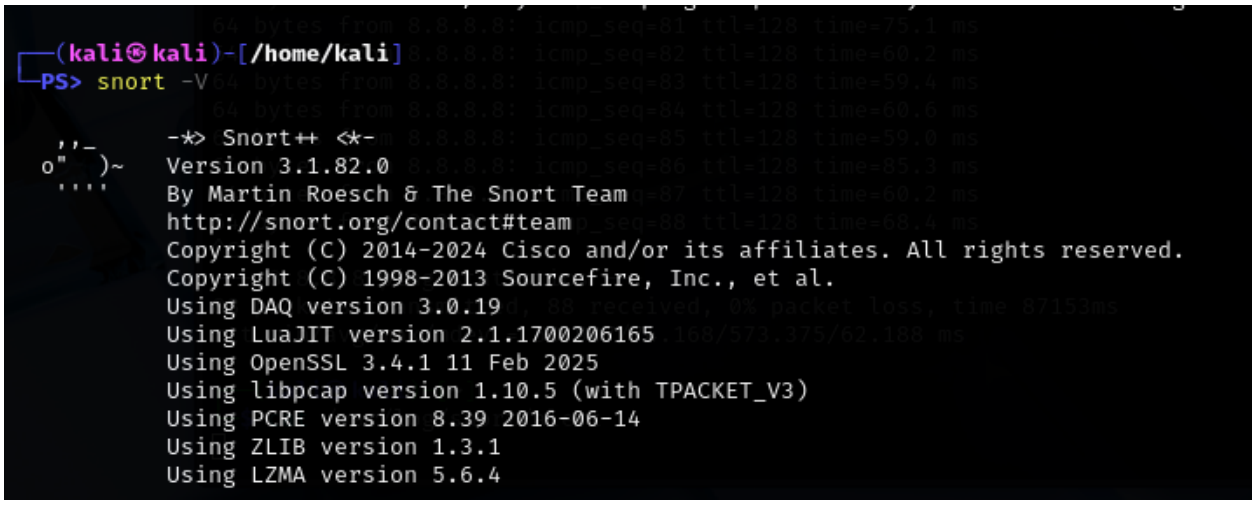
ASSIGNMENT 10

Name: Yogita Mundankar
Rollno. B22CS068

Task 1: Snort Installation

- I**nstall Snort on Kali Linux:** Snort was installed using the following commands:
  sudo apt update
  sudo apt install snort

- **Verify the installation and check the version:** After installation, the version was verified using:



Task 2: Configuring Snort and write Snort Rules

- Configuring snort from the /etc/snort/snort.lua:

- Inside the configuration file made the following changes:

  The `HOME_NET` variable was set to reflect the IP range of the local network:
  IP of VM:192.168.235.128
  Subnet: 192.168.235.0/24

```
┌──(kali㉿kali)-[/home/kali]
└─PS> 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 noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:23:d8:98 brd ff:ff:ff:ff:ff:ff
    inet 192.168.235.128/24 brd 192.168.235.255 scope global dynamic noprefixroute eth0
       valid_lft 1068sec preferred_lft 1068sec
    inet6 fe80::f27e:d33a:db51:f34a/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
-- setup the network addresses you are protecting
HOME_NET = '192.168.235.0/24'
EXTERNAL_NET = 'any'


--HOME_NET = '192.168.235.0/24'

-- set up the external network addresses.
-- (leave as "any" in most situations)
--EXTERNAL_NET = 'any'
include 'snort_defaults.lua'
```

- Ensure local rules are included:

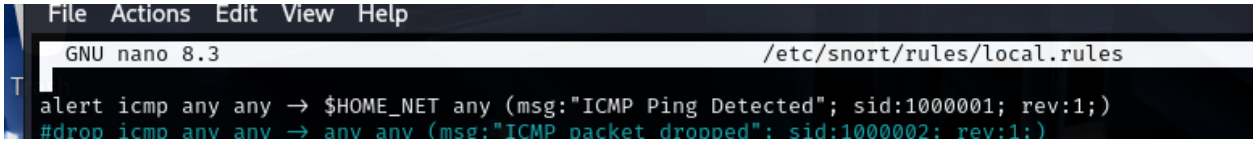
```
references = default_references
classifications = default_classifications
--RULE_PATH = '/etc/snort/rules'-
--include 'snort_defaults.lua'

ips =
{
    -- use this to enable decoder and inspector alerts
    --  enable_builtin_rules = true,

    -- use include for rules files; be sure to set your path
    -- note that rules files can include other rules files
    -- (see also related path vars at the top of snort_defaults.lua)
    include = '/etc/snort/rules/local.rules',
    variables = default_variables
}
```

```
alert_fast = {file = true, }
--alert_full = { }
--alert_sfsocket = { }
```

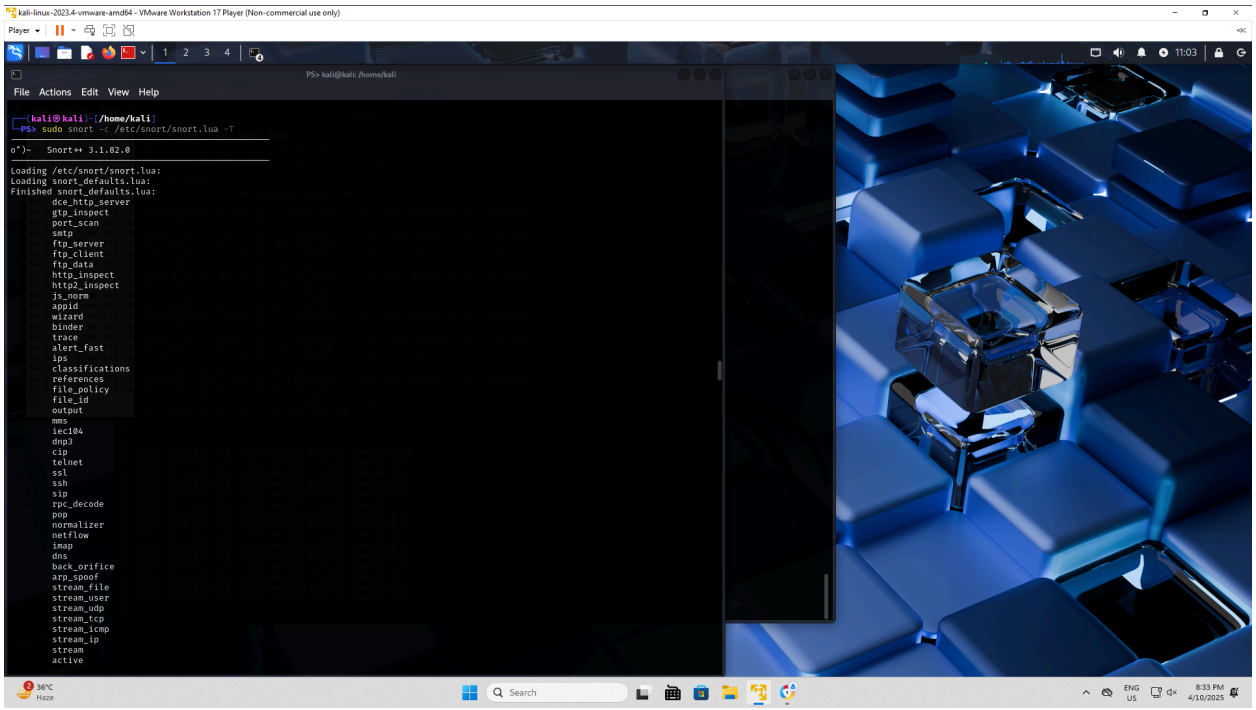
- Writing Snort Rules:

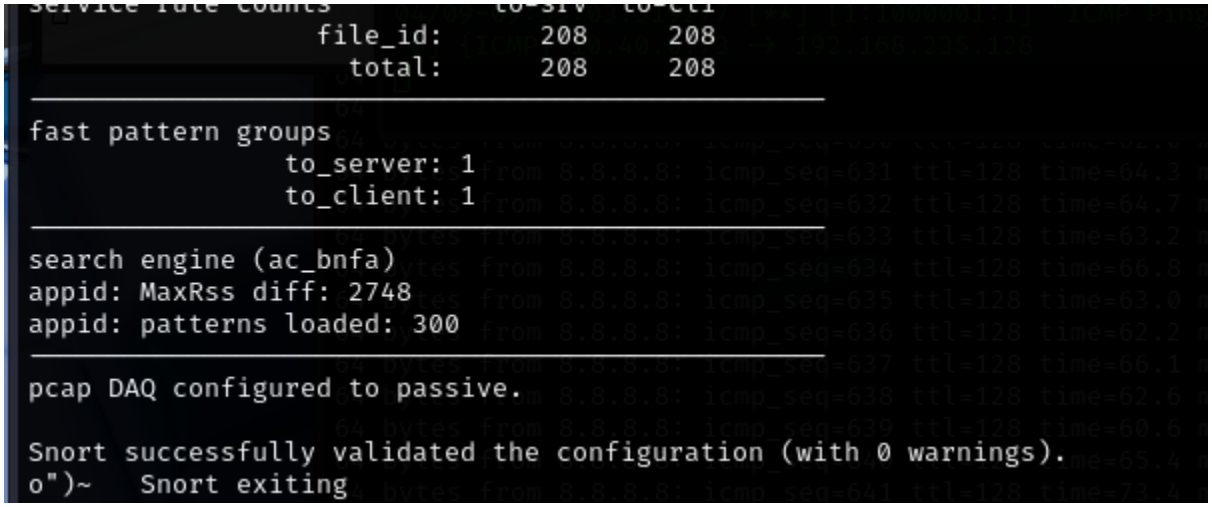
```
┌──(kali㉿kali)-[/home/kali]
└─PS> sudo nano /etc/snort/rules/local.rules
```

- A custom rule was created to detect ICMP (ping) traffic:



- Checked if the file is configured successfully:



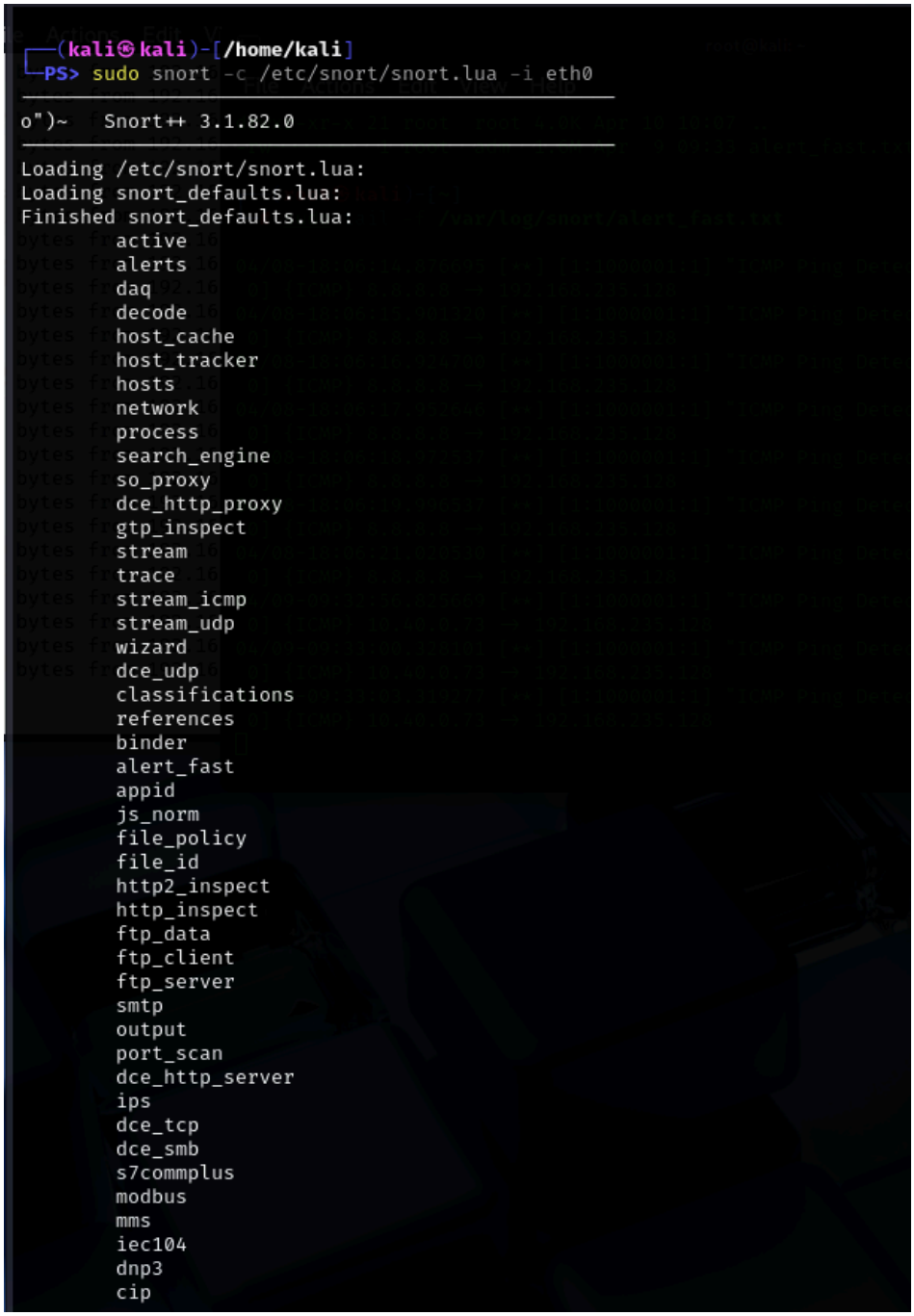


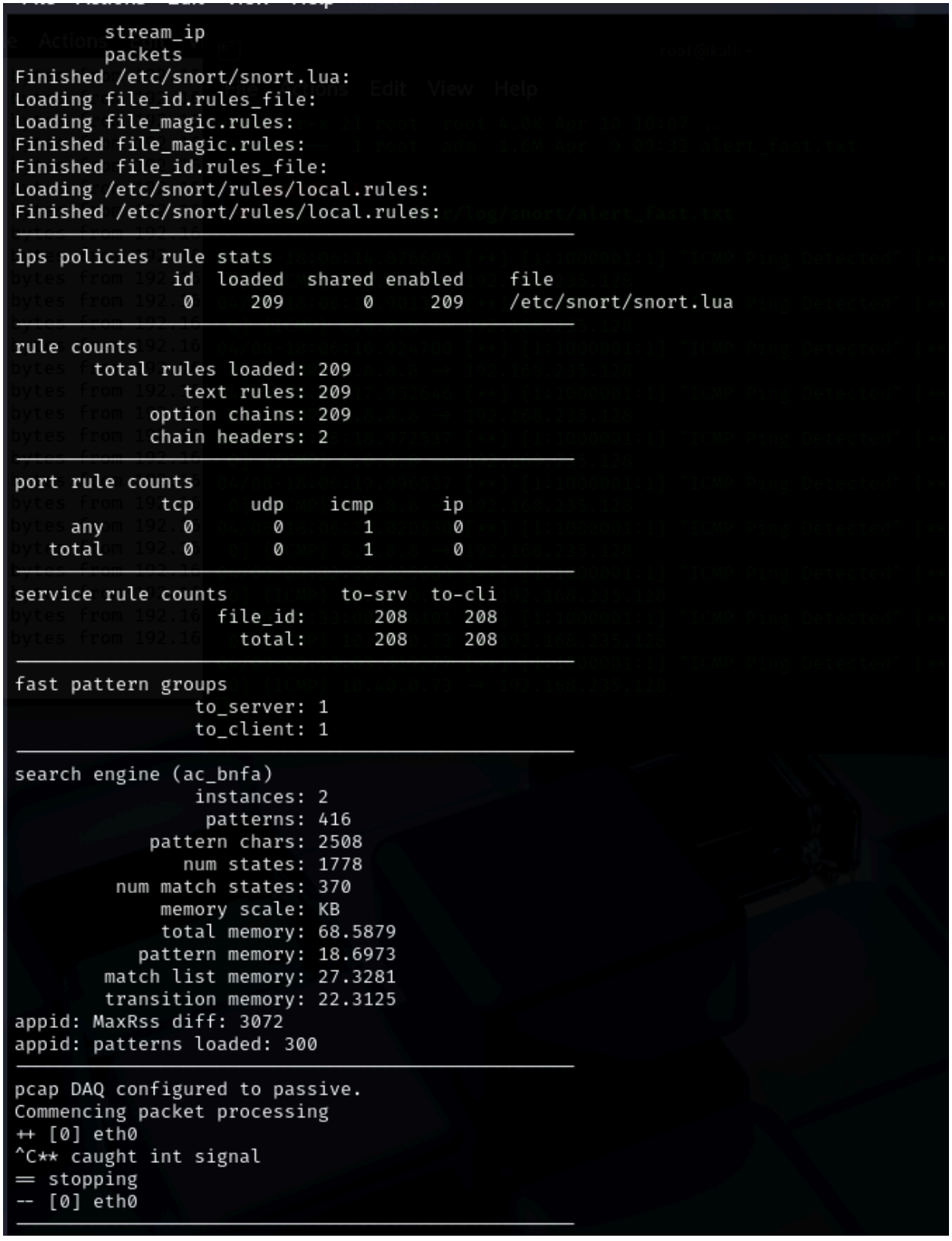
Snort is successfully configured as can be seen from the above message.

Task 3: Running Snort in IDS Mode

- Running Snort in IDS:

Start Snort in IDS mode and monitor network traffic:




stream_ip
packets
Finished /etc/snort/snort.lua:
Loading file_id.rules_file:
Loading file_magic.rules:
Finished file_magic.rules:
Finished file_id.rules_file:
Loading /etc/snort/rules/local.rules:
Finished /etc/snort/rules/local.rules:
ips policies rule stats
id loaded shared enabled file
0 209 0 209 /etc/snort/snort.lua
rule counts
total rules loaded: 209
text rules: 209
option chains: 209
chain headers: 2
port rule counts
tcp udp icmp ip
any 0 0 1 0
total 0 0 1 0
service rule counts to-srv to-cli
file_id: 208 208
total: 208 208
fast pattern groups
to_server: 1
to_client: 1
search engine (ac_bnfa)
instances: 2
patterns: 416
pattern chars: 2508
num states: 1778
num match states: 370
memory scale: KB
total memory: 68.5879
pattern memory: 18.6973
match list memory: 27.3281
transition memory: 22.3125
appid: MaxRss diff: 3072
appid: patterns loaded: 300
pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
^C** caught int signal
== stopping
-- [0] eth0

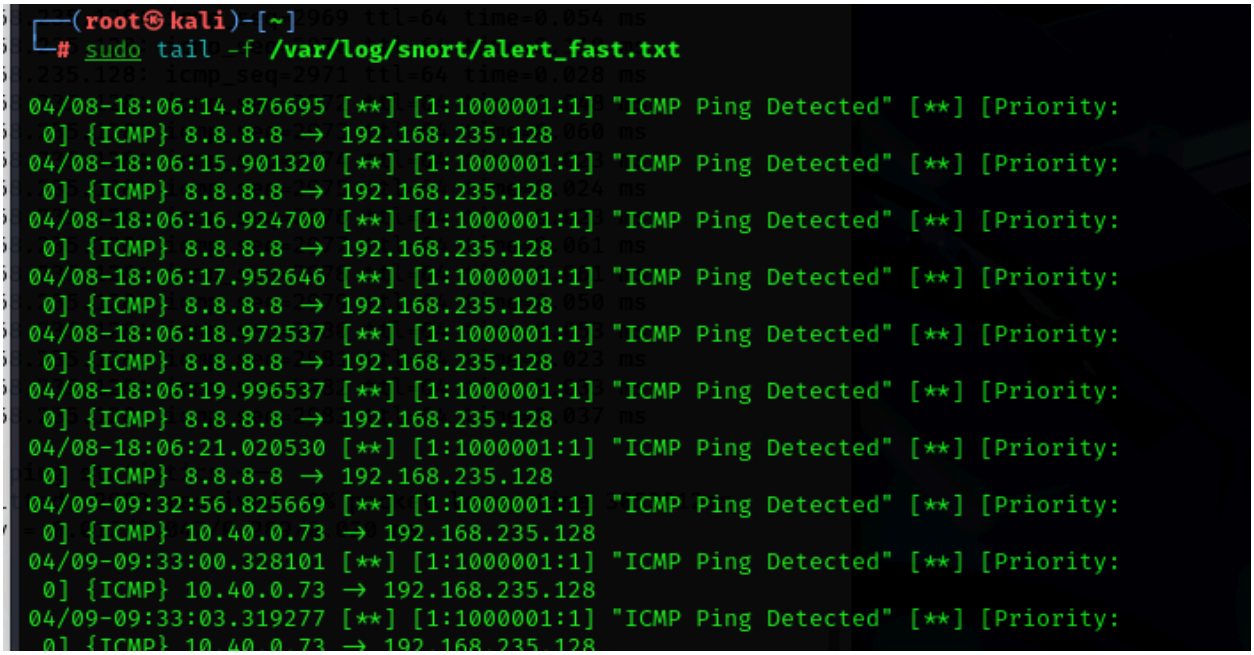
```
Packet Statistics
-------------------------------------------------------------------
daq
                received: 748
                analyzed: 748
                   allow: 748
                rx_bytes: 70866
-------------------------------------------------------------------
codec
                   total: 748          (100.000%)
                     arp: 68           (  9.091%)
                     eth: 748          (100.000%)
                   icmp4: 655          ( 87.567%)
                   icmp6: 5            (  0.668%)
                    igmp: 5            (  0.668%)
                    ipv4: 669          ( 89.439%)
                    ipv6: 11           (  1.471%)
           ipv6_hop_opts: 5            (  0.668%)
                     udp: 15           (  2.005%)
-------------------------------------------------------------------
Module Statistics
-------------------------------------------------------------------
appid
                 packets: 680
       processed_packets: 680
          total_sessions: 8
      service_cache_adds: 3
           bytes_in_use: 456
            items_in_use: 3
-------------------------------------------------------------------
arp_spoof
                 packets: 68
-------------------------------------------------------------------
back_orifice
                 packets: 15
-------------------------------------------------------------------
binder
             raw_packets: 68
               new_flows: 8
                inspects: 76
-------------------------------------------------------------------
detection
                analyzed: 748
              hard_evals: 660
                  alerts: 327
            total_alerts: 327
                 logged: 327
-------------------------------------------------------------------
port_scan
                 packets: 680
                trackers: 11
-------------------------------------------------------------------
search_engine
    non_qualified_events: 333
```


stream
flows: 8
total_prunes: 2
idle_prunes_proto_timeout: 2
udp_timeout_prunes: 2
stream_icmp
sessions: 2
max: 2
created: 2
released: 2
stream_ip
sessions: 1
max: 1
created: 1
released: 1
total_bytes: 80
stream_udp
sessions: 5
max: 5
created: 5
released: 5
total_bytes: 1258
wizard
udp_scans: 5
udp_misses: 5
Appid Statistics
detected apps and services
Application: Services Clients Users Payloads Misc Referred
unknown: 5 0 0 0 0 0
Summary Statistics
process
signals: 1
timing
runtime: 00:05:28
seconds: 328.093612
pkts/sec: 2
o")~ Snort exiting

- Generate ICMP traffic:

```
┌──(root㉿kali)-[~]
└─# ping 192.168.235.128

PING 192.168.235.128 (192.168.235.128) 56(84) bytes of data.
64 bytes from 192.168.235.128: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 192.168.235.128: icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from 192.168.235.128: icmp_seq=3 ttl=64 time=0.025 ms
64 bytes from 192.168.235.128: icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from 192.168.235.128: icmp_seq=5 ttl=64 time=0.037 ms
64 bytes from 192.168.235.128: icmp_seq=6 ttl=64 time=0.042 ms
64 bytes from 192.168.235.128: icmp_seq=7 ttl=64 time=0.046 ms
64 bytes from 192.168.235.128: icmp_seq=8 ttl=64 time=0.041 ms
64 bytes from 192.168.235.128: icmp_seq=9 ttl=64 time=0.032 ms
64 bytes from 192.168.235.128: icmp_seq=10 ttl=64 time=0.040 ms
64 bytes from 192.168.235.128: icmp_seq=11 ttl=64 time=0.047 ms
64 bytes from 192.168.235.128: icmp_seq=12 ttl=64 time=0.025 ms
64 bytes from 192.168.235.128: icmp_seq=13 ttl=64 time=0.054 ms
64 bytes from 192.168.235.128: icmp_seq=14 ttl=64 time=0.024 ms
64 bytes from 192.168.235.128: icmp_seq=15 ttl=64 time=0.024 ms
64 bytes from 192.168.235.128: icmp_seq=16 ttl=64 time=0.023 ms
64 bytes from 192.168.235.128: icmp_seq=17 ttl=64 time=0.025 ms
64 bytes from 192.168.235.128: icmp_seq=18 ttl=64 time=0.022 ms
64 bytes from 192.168.235.128: icmp_seq=19 ttl=64 time=0.063 ms
64 bytes from 192.168.235.128: icmp_seq=20 ttl=64 time=0.023 ms
64 bytes from 192.168.235.128: icmp_seq=2977 ttl=64 time=0.061 ms
64 bytes from 192.168.235.128: icmp_seq=2978 ttl=64 time=0.061 ms
64 bytes from 192.168.235.128: icmp_seq=2979 ttl=64 time=0.050 ms
64 bytes from 192.168.235.128: icmp_seq=2980 ttl=64 time=0.053 ms
64 bytes from 192.168.235.128: icmp_seq=2981 ttl=64 time=0.023 ms
64 bytes from 192.168.235.128: icmp_seq=2982 ttl=64 time=0.033 ms
64 bytes from 192.168.235.128: icmp_seq=2983 ttl=64 time=0.037 ms
^C
── 192.168.235.128 ping statistics ──
2983 packets transmitted, 2983 received, 0% packet loss, time 3053612ms
rtt min/avg/max/mdev = 0.017/0.047/0.289/0.020 ms
```

- Capture alerts in the Snort logs.



TASK 4: Running Snort in IPS Mode

- Modify the Snort rule to drop ICMP traffic:

```
┌──(kali㉿kali)-[/home/kali]
└─PS> sudo nano /etc/snort/rules/local.rules
```

```
File  Actions  Edit  View  Help
  GNU nano 8.3                    /etc/snort/rules/local.rules
drop icmp any any → any any (msg:"ICMP Packet Dropped"; sid:10000002; rev:1;)
```

- Configure iptables to redirect traffic to Snort:

```
┌──(kali㉿kali)-[/home/kali]
└─PS> sudo nano /etc/snort/rules/local.rules

┌──(kali㉿kali)-[/home/kali]
└─PS> sudo iptables -F

┌──(kali㉿kali)-[/home/kali]
└─PS> sudo iptables -t nat -F

┌──(kali㉿kali)-[/home/kali]
└─PS> sudo iptables -t mangle -F

┌──(kali㉿kali)-[/home/kali]
└─PS> sudo sysctl1 -w net.ipv4.ip_forward=1
sudo: sysctl1: command not found

┌──(kali㉿kali)-[/home/kali]
└─PS> sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1

┌──(kali㉿kali)-[/home/kali]
└─PS> sudo iptables -A FORWARD -j NFQUEUE --queue-num 0
```

- Run Snort in Inline IPS Mode:

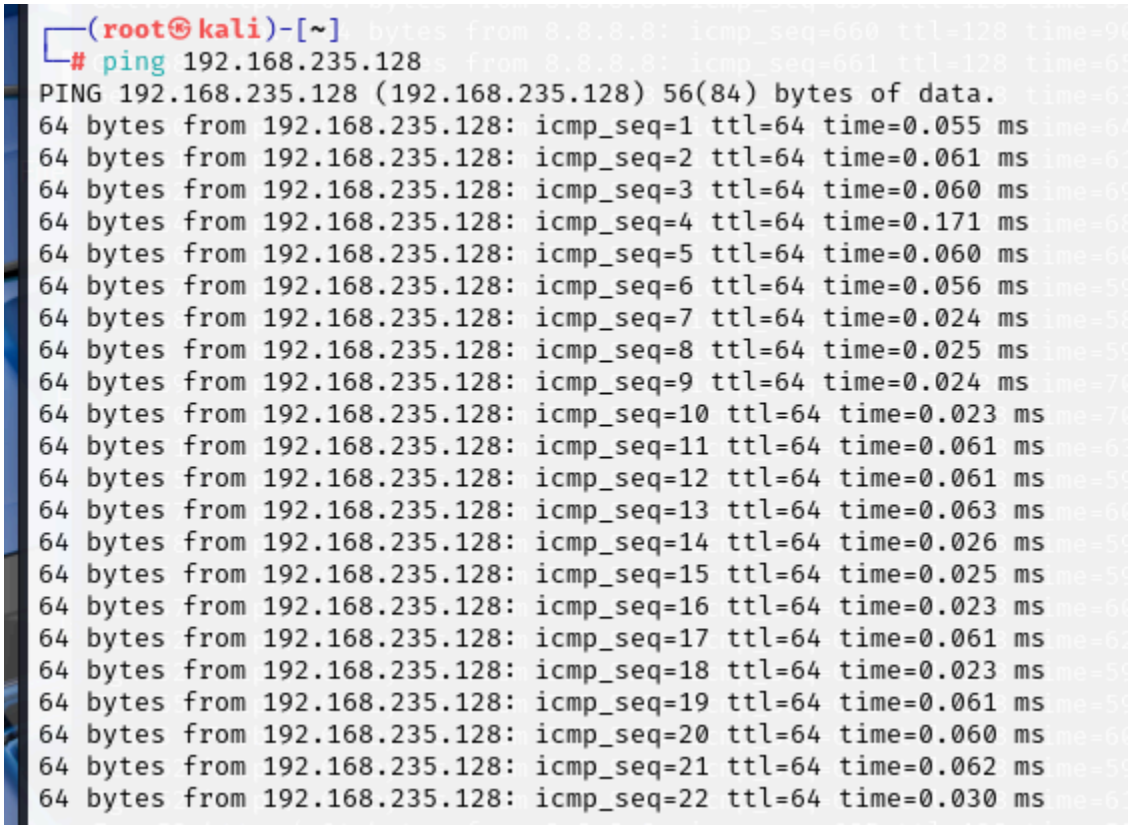
```
┌──(root㉿kali)-[~]
└─# sudo snort -c /etc/snort/snort.lua -i eth0 --daq afpacket --daq-mode inline
```

```
o")~   Snort++ 3.1.82.0
-------------------------------------------------------------------------------
Loading /etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
        smtp
        ftp_server
        ftp_client
        ftp_data
        output
        http2_inspect
        js_norm
        appid
        wizard
        binder
        ips
        classifications
        http_inspect
        file_policy
        file_id
        references
```

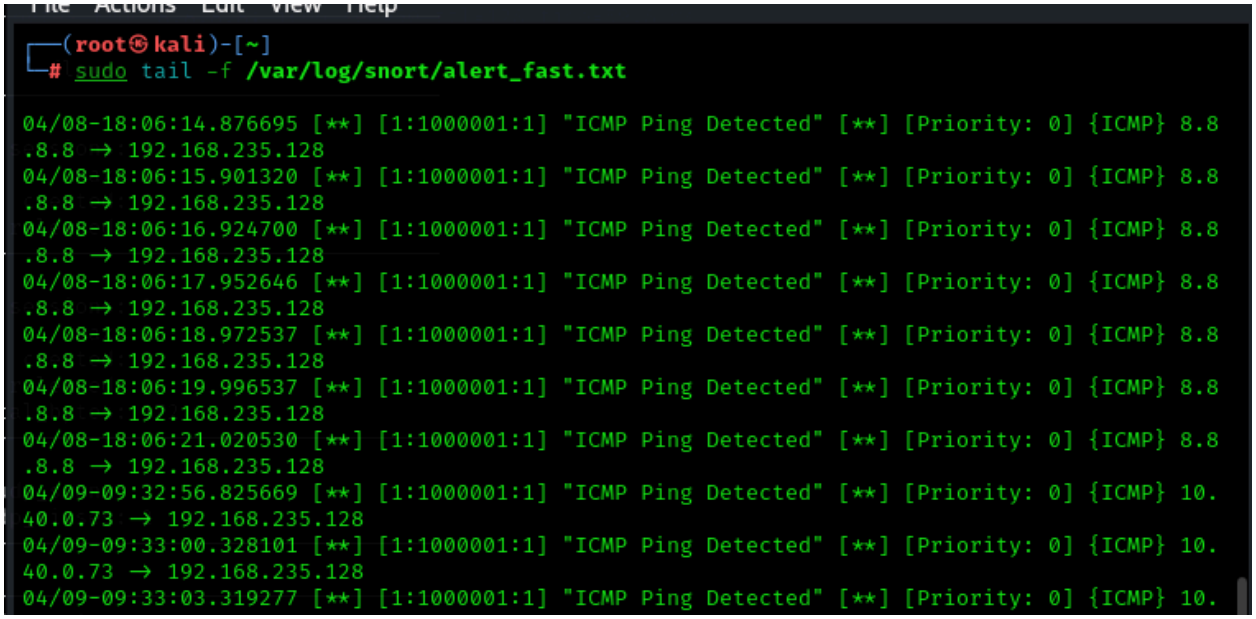
```
== stopping
04/08-11:32:47.265318 [**] [1:1000001:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.235.1 → 192.168.235.128
04/08-11:32:47.265361 [**] [1:1000001:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.235.128 → 192.168.235.1
04/08-11:32:48.271169 [**] [1:1000001:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.235.1 → 192.168.235.128
04/08-11:32:48.271217 [**] [1:1000001:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.235.128 → 192.168.235.1
04/08-11:32:49.290829 [**] [1:1000001:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.235.1 → 192.168.235.128
04/08-11:32:49.290871 [**] [1:1000001:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.235.128 → 192.168.235.1
04/08-11:32:50.296148 [**] [1:1000001:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.235.1 → 192.168.235.128
04/08-11:32:50.296189 [**] [1:1000001:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.235.128 → 192.168.235.1
-- [0] eth0
--------------------------------------------------
Packet Statistics
--------------------------------------------------
daq
                received: 17
                analyzed: 17
                   allow: 17
                rx_bytes: 1096
--------------------------------------------------
codec
                   total: 17          (100.000%)
                     arp: 9           ( 52.941%)
                     eth: 17          (100.000%)
                   icmp4: 8           ( 47.059%)
                    ipv4: 8           ( 47.059%)
--------------------------------------------------
Module Statistics
--------------------------------------------------
appid
                 packets: 8
        processed_packets: 8
           total_sessions: 1
--------------------------------------------------
arp_spoof
                 packets: 9
--------------------------------------------------
binder
              raw_packets: 9
                new_flows: 1
                 inspects: 10
--------------------------------------------------
detection
                analyzed: 17
```

TASK 5: Testing and logging

- Generate ICMP Traffic:



- Verify ICMP Packets are Being Dropped:



Conclusion:

Snort was successfully installed and configured to operate in both IDS and IPS modes:

- In **IDS mode**, it generated alerts upon detecting ICMP packets.
- In **IPS mode**, it effectively **blocked ICMP traffic**, as verified through ping failure and corresponding log entries.