DEMONSTRATE INTRUSION DETECTION SYSTEM (SNORT)

EXP.NO: 8   **DATE:25-03-2025**

SNORT is an open-source, rule-based Network Intrusion Detection and Prevention System (NIDS/NIPS). Snort is the foremost Open Source Intrusion Prevention System (IPS) in the world. 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 generate alerts for users

AIM:

To start working with Snort to analyse live and captured traffic.

REQUIREMENTS:

- To know basic Linux command-line functionalities like general system navigation and Network fundamentals (ports, protocols and traffic data)

- To have general knowledge of network basics and Linux fundamentals

- Must complete the Network Fundamentals module and Linux Fundamentals rooms (1 2 3) in try hack me platform.

ALGORITHM:

1.   Setup Interactive material and exercise for snort instance setup. Use the folder

"Task-Exercises" on the Desktop.

2.   to generate traffic to our snort interface using the script traffic-generator.sh to trigger traffic to the

snort interface.

3.   Run the "traffic generator.sh" file by executing it as sudos

4.   Choose the exercise type and then automatically open another terminal to show you the output of

the selected action

5.   Once you choose an action, the menu disappears and opens a terminal instance to show you the

output of the action.

6.   Navigate to the Task-Exercises folder and run the command "./.easy.sh" and write the output.

the

```
ubuntu@ip-10-10-138-56:~$ cd Desktop/Task-Exercises/
ubuntu@ip-10-10-138-56:~/Desktop/Task-Exercises$ ./.easy.sh
Too Easy!
ubuntu@ip-10-10-138-56:~/Desktop/Task-Exercises$
```

7. Read details about the Introduction about the IDS and IPS and answer the following questions and

    answer it

    a. Which snort mode can help you stop the threats on a local machine? **Answer:** HIPS

    b. Which snort mode can help you detect threats on a local network? **Answer:** NIDS

    c. Which snort mode can help you detect the threats on a local machine? **Answer:** HIDS

    d. Which snort mode can help you stop the threats on a local network? **Answer:** NIPS

    e. Which snort mode works similar to NIPS mode? **Answer:** NBA

    f. According to the official description of the snort, what kind of NIPS is it? **Answer:** full-blown

    g. NBA training period is also known as … **Answer:** baselining 8. Read the Task 4 content to make

    first interaction with snort instance Run the Snort instance and check the build number. Command: snort -V

```
ubuntu@ip-          :~/Desktop/Task-Exercises$ snort -V
          -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 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: 8.39 2016-06-14
           Using ZLIB version: 1.2.11
```

9. Test the current instance with "/etc/snort/snort.conf" file and check how many rules are loaded

    with the current build. snort -T -c /etc/snort/snort.conf

```
ubuntu@ip-          :~/Desktop/Task-Exercises$ snort -T -c /etc/snort/snort.conf
Running in Test mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
```

the

```
4151 Snort rules read
    3477 detection rules
    0 decoder rules
    0 preprocessor rules
3477 Option Chains linked into 271 Chain Headers
0 Dynamic rules
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

10. Test current instance with "/etc/snort/snortv2.conf" file and check how many rules are loaded

   with the current build. snort -T -c /etc/snort/snortv2.conf

```
ubuntu@ip-           :~/Desktop/Task-Exercises$ snort -T -c /etc/snort/snortv2.conf
Running in Test mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snortv2.conf"
```

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

11. Read to know Sniffer Mode operation and their parameters

Snort has various flags capable of viewing various data about the packet it is ingesting.Sniffer mode parameters:

● -v -Verbose. Display the TCP/IP output in the console.

● -d -Display the packet data (payload).

● -e -Display the link-layer (TCP/IP/UDP/ICMP) headers.

● -X -Display the full packet details in HEX.

● -i -This parameter helps to define a specific network interface to listen/sniff.

the

sudo snort -v-i eth0

sudo snort -v sudo

snort -d sudo snort

-de sudo snort -X

snort -vd snort -de

snort -v -d -e

12. Read given content to know Packet Logger Mode operation and their parameters Packet

logger parameters:

- -l -Logger mode, target log and alert output directory. Default output folder is

/var/log/snort. The default action is to dump as tcpdump format in /var/log/snort ●

-K ASCII- Log packets in ASCII format.

- -r -Reading option, read the dumped logs in Snort.

- -n -Specify the number of packets that will process/read. Snort will stop after reading the specified number of

   packets.

1. Investigate the traffic with the default configuration file with ASCII mode. sudo snort -dev -K

   ASCII -l

2. Execute the traffic generator script and choose "TASK-6 Exercise". Wait until the traffic ends,

   then stop the Snort instance. Now analyse the output summary and answer the question.

   sudo ./traffic-generator.sh

Now, you should have the logs in the current directory. Navigate to folder "145.254.160.237".

3. What is the source port used to connect port 53?
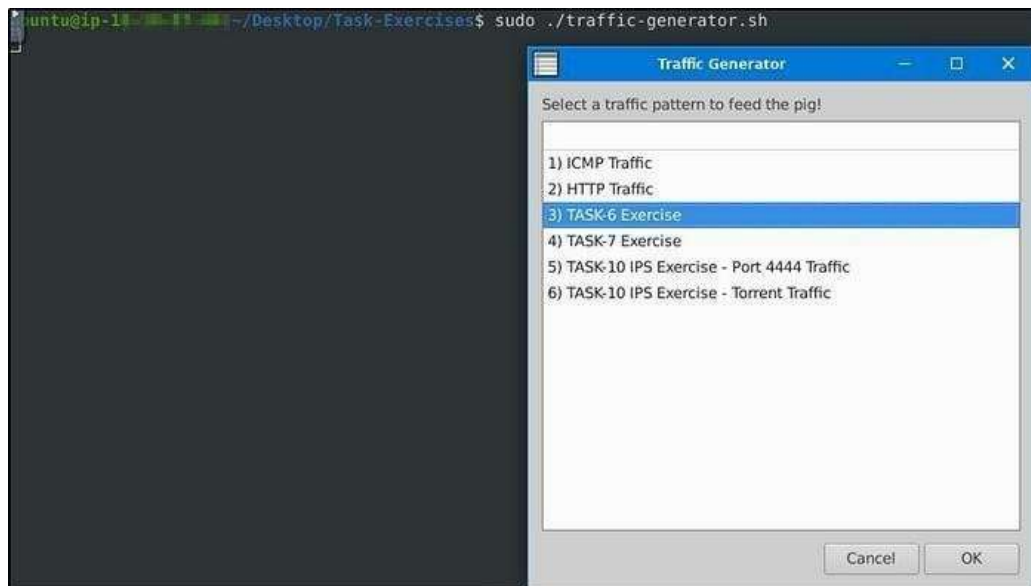
Answer: 3009 Run first snort in logger mode.

   sudo snort -dev -K ASCII -l .

Run the traffic generator script.

   sudo ./traffic-generator.sh

13. We are going to select task #3. As Task #6- Exercise

the

Let's cd to the folder created. We see 3 log files were created, which also denotes the port numbers the machine used in the traffic generated

```
root@ip-██ ███ ██:/home/ubuntu/Desktop/Task-Exercises# cd 145.254.160.237/
root@ip-██ ███ ██:/home/ubuntu/Desktop/Task-Exercises/145.254.160.237# ls
TCP:3371-80  TCP:3372-80  UDP:3009-53
root@ip-█ ███ ██:/home/ubuntu/Desktop/Task-Exercises/145.254.160.237# cat UDP\:3009-53
07/02-10:07:34.985487 00:00:01:00:00:00 -> FE:FF:20:00:01:00 type:0x800 len:0x59
145.254.160.237:3009 -> 145.253.2.203:53 UDP TTL:128 TOS:0x0 ID:3913 IpLen:20 DgmLen:75
Len: 47
00 23 01 00 00 01 00 00 00 00 00 00 07 70 61 67   .#...........pag
65 61 64 32 11 67 6F 6F 67 6C 65 73 79 6E 64 69   ead2.googlesyndi
63 61 74 69 6F 6E 03 63 6F 6D 00 00 01 00 01      cation.com.....
```

Use snort.log.1640048004

14. Read the snort.log file with Snort;

What is the IP ID of the 10th packet?

Answer: 49313

The log file created should be in the current directory.

```
ubuntu@ip-1██ ██ ███:~/Desktop/Task-Exercises/Exercise-Files$ cd TASK-6
ubuntu@ip-1██ ██ ███:~/Desktop/Task-Exercises/Exercise-Files/TASK-6$ ls
snort.log.1640048004
```

snort -r snort.log.1640048004 -n 10

```
WARNING: No preprocessors configured for policy 0.
05/13-10:17:09.754737 65.208.228.223:80 -> 145.254.160.237:3372
TCP TTL:47 TOS:0x0 ID:49313 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x114C6C54  Ack: 0x38AFFFF3  Win: 0x1920  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Read the "snort.log.1640048004" file with Snort; what is the referer of the 4th packet?

Answer: http://www.ethereal.com/development.html
Add "-X" to display results in ASCII format.

sudo snort -Xr snort.log.1640048004 -n 4

```
ubuntu@ip.██ ████.██:~/Desktop/Task-Exercises/Exercise-Files/TASK-6$ sudo snort -Xr snort.log.1640048004  -n 4
Exiting after 4 packets
Running in packet dump mode
```

```
WARNING: No preprocessors configured for policy 0.
05/13-10:17:08.222534 145.254.160.237:3372 -> 65.208.228.223:80
TCP TTL:128 TOS:0x0 ID:3909 IpLen:20 DgmLen:519 DF
***AP*** Seq: 0x38AFFE14  Ack: 0x114C618C  Win: 0x25BC  TcpLen: 20
0x0000: FE FF 20 00 01 00 00 00 01 00 00 00 08 00 45 00  .. ...........E.
0x0010: 02 07 0F 45 40 00 80 06 90 10 91 FE A0 ED 41 D0  ...E@.........A.
0x0020: E4 DF 0D 2C 00 50 38 AF FE 14 11 4C 61 8C 50 18  ...,.P8....La.P.
0x0030: 25 BC A9 58 00 00 47 45 54 20 2F 64 6F 77 6E 6C  %..X..GET /downl
0x0040: 6F 61 64 2E 68 74 6D 6C 20 48 54 54 50 2F 31 2E  oad.html HTTP/1.
0x0050: 31 0D 0A 48 6F 73 74 3A 20 77 77 77 2E 65 74 68  1..Host: www.eth
0x0060: 65 72 65 61 6C 2E 63 6F 6D 0D 0A 55 73 65 72 2D  ereal.com..User-
0x0070: 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 35  Agent: Mozilla/5
0x0080: 2E 30 20 28 57 69 6E 64 6F 77 73 3B 20 55 3B 20  .0 (Windows; U;
0x0090: 57 69 6E 64 6F 77 73 20 4E 54 20 35 2E 31 3B 20  Windows NT 5.1;
0x00A0: 65 6E 2D 55 53 3B 20 72 76 3A 31 2E 36 29 20 47  en-US; rv:1.6) G
0x00B0: 65 63 6B 6F 2F 32 30 30 34 30 31 31 33 0D 0A 41  ecko/20040113..A
0x00C0: 63 63 65 70 74 3A 20 74 65 78 74 2F 78 6D 6C 2C  ccept: text/xml,
0x00D0: 61 70 70 6C 69 63 61 74 69 6F 6E 2F 78 6D 6C 2C  application/xml,
0x00E0: 61 70 70 6C 69 63 61 74 69 6F 6E 2F 78 68 74 6D  application/xhtm
0x00F0: 6C 2B 78 6D 6C 2C 74 65 78 74 2F 68 74 6D 6C 3B  l+xml,text/html;
0x0100: 71 3D 30 2E 39 2C 74 65 78 74 2F 70 6C 61 69 6E  q=0.9,text/plain
0x0110: 3B 71 3D 30 2E 38 2C 69 6D 61 67 65 2F 70 6E 67  ;q=0.8,image/png
0x0120: 2C 69 6D 61 67 65 2F 6A 70 65 67 2C 69 6D 61 67  ,image/jpeg,imag
0x0130: 65 2F 67 69 66 3B 71 3D 30 2E 32 2C 2A 2F 2A 3B  e/gif;q=0.2,*/*;
0x0140: 71 3D 30 2E 31 0D 0A 41 63 63 65 70 74 2D 4C 61  q=0.1..Accept-La
0x0150: 6E 67 75 61 67 65 3A 20 65 6E 2D 75 73 2C 65 6E  nguage: en-us,en
0x0160: 3B 71 3D 30 2E 35 0D 0A 41 63 63 65 70 74 2D 45  ;q=0.5..Accept-E
0x0170: 6E 63 6F 64 69 6E 67 3A 20 67 7A 69 70 2C 64 65  ncoding: gzip,de
0x0180: 66 6C 61 74 65 0D 0A 41 63 63 65 70 74 2D 43 68  flate..Accept-Ch
0x0190: 61 72 73 65 74 3A 20 49 53 4F 2D 38 38 35 39 2D  arset: ISO-8859-
0x01A0: 31 2C 75 74 66 2D 38 3B 71 3D 30 2E 37 2C 2A 3B  1,utf-8;q=0.7,*;
0x01B0: 71 3D 30 2E 37 0D 0A 4B 65 65 70 2D 41 6C 69 76  q=0.7..Keep-Aliv
0x01C0: 65 3A 20 33 30 30 0D 0A 43 6F 6E 6E 65 63 74 69  e: 300..Connecti
0x01D0: 6F 6E 3A 20 6B 65 65 70 2D 61 6C 69 76 65 0D 0A  on: keep-alive..
0x01E0: 52 65 66 65 72 65 72 3A 20 68 74 74 70 3A 2F 2F  Referer: http://
0x01F0: 77 77 77 2E 65 74 68 65 72 65 61 6C 2E 63 6F 6D  www.ethereal.com
0x0200: 2F 64 65 76 65 6C 6F 70 6D 65 6E 74 2E 68 74 6D  /development.htm
0x0210: 6C 0D 0A 0D 0A                                   l....
```

Read the "snort.log.1640048004" file with Snort; what is the Ack number of the 8th packet?

Answer: 0x38AFFFF3

sudo snort -r snort.log.1640048004 -n 8

Note to read the 8th packet of the results.

```
WARNING: No preprocessors configured for policy 0.
05/13-10:17:09.123830 65.208.228.223:80 -> 145.254.160.237:3372
TCP TTL:47 TOS:0x0 ID:49312 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x114C66F0  Ack: 0x38AFFFF3  Win: 0x1920  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Read the "snort.log.1640048004" file with Snort; what is the number of the "TCP port 80" packets?

Answer: 41

For this task, we will be utilizing "BPF". According to Wikipedia, "The Berkeley Packet Filter (BPF) is a technology used in certain computer operating systems for programs that need to, among other things, analyze network traffic. It provides a raw interface to data link layers, permitting raw link-layer packets to be sent and received."

Check out the syntax for BPF here: https://biot.com/capstats/bpf.html sudo

snort -r snort.log.1640048004 'tcp port 80'

The result will only display traffic captured from port 80.

```
Packet I/O Totals:
   Received:              41
   Analyzed:              41 (100.000%)
    Dropped:               0 (  0.000%)
   Filtered:               0 (  0.000%)
Outstanding:               0 (  0.000%)
   Injected:               0
```

TASK 7: OPERATION MODE 3: IDS/IPS

IDS/IPS mode depends on the rules and configuration. TASK-10 summarises the essential paths, files and variables. Also, TASK-3 covers configuration testing. Here, we need to understand the operating logic first, and then we will be going into rules in TASK-9

NIDS mode parameters:

- -c :Defining the configuration file.

- -T :Testing the configuration file.

- -N :Disable logging.

- -D :Background mode.

- -A: Alert modes;

- full: Full alert mode, providing all possible information about the alert. This one also is the default mode; once you use -A and don't specify any mode, snort uses this mode.

- fast: Fast mode shows the alert message, timestamp, source and destination IP, along with port numbers.

- console: Provides fast style alerts on the console screen.

- cmg: CMG style, basic header details with payload in hex and text format.

- none: Disabling alerting

Once you start running IDS/IPS mode, you need to use rules. We will use a pre-defined ICMP rule as an example. The defined rule will only generate alerts in any direction of ICMP packet activity.

alert icmp any any <> any any (msg: 'ICMP Packet Found'; sid: 100001; rev:1;)

IDS/IPS mode with the different parameters:

sudo snort -c /etc/snort/snort.conf -T sudo

snort -c /etc/snort/snort.conf -N sudo snort -c

/etc/snort/snort.conf -D sudo snort -c

/etc/snort/snort.conf -D -X -l . sudo snort -c

/etc/snort/snort.conf -A console sudo snort -c

/etc/snort/snort.conf -A cmg sudo snort -c

/etc/snort/snort.conf -A fast sudo snort -c

/etc/snort/snort.conf -A full sudo snort -c

/etc/snort/snort.conf -A none With parameter

"-D", we can activate verbosity (-v) or full

packet dump (-X) with packet logger mode (-

l) and we will still have the logs in the logs

folder, but there will be no output in the

console.

Once you start the background mode and want to check the corresponding process, you can easily use the "ps" command as shown below:

    ps -ef | grep snort

If you want to stop the daemon, you can easily use the "kill" command to stop the process.

    sudo kill -9 <pid>

Using rule file without configuration file

    sudo snort -c /etc/snort/rules/local.rules -A console

IPS mode and dropping packets

Snort IPS mode activated with -Q — daq afpacket parameters. You can also activate this mode by editing snort.conf file.

Activate the Data Acquisition (DAQ) modules and use the afpacket module to use snort as an IPS: -i eth0:eth1

    sudo snort -c /etc/snort/snort.conf -q -Q --daq afpacket -i eth0:eth1 -A console Investigate

the traffic with the default configuration file.

    sudo snort -c /etc/snort/snort.conf -A full -l .

Execute the traffic generator script and choose "TASK-7 Exercise". Wait until the traffic stops, then stop the

Snort instance. Now analyse the output summary and answer the question.

sudo ./traffic-generator.sh

What is the number of the detected HTTP GET methods? **Answer:** 2

```
HTTP Inspect - encodings (Note: stream-reassembled packets included):
    POST methods:                        0
    GET methods:                         2
    HTTP Request Headers extracted:      2
    HTTP Request Cookies extracted:      0
    Post parameters extracted:           0
    HTTP response Headers extracted:     3
    HTTP Response Cookies extracted:     0
    Unicode:                             0
    Double unicode:                      0
    Non-ASCII representable:             0
    Directory traversals:                0
    Extra slashes ("//"):                1
    Self-referencing paths ("./"):       0
    HTTP Response Gzip packets extracted: 1
    Gzip Compressed Data Processed:      1272.00
    Gzip Decompressed Data Processed:    3608.00
    Total packets processed:             2142
```

TASK 8: OPERATION MODE 4: PCAP INVESTIGATION

Capabilities of Snort are not limited to sniffing, logging and detecting/preventing the threats. PCAP

read/investigate mode helps us work with pcap files. Once we have a pcap file and process it with Snort, we

will receive default traffic statistics with alerts depending on our rule set.

PCAP mode parameters:

● - r / — pcap-single= :Read a single pcap

● — pcap-list="" :Read pcaps provided in command (space separated).

● — pcap-show :Show pcap name on console during processing.

Investigating single pcap file with a configuration file.

> sudo snort -c /etc/snort/snort.conf -q -r icmp-test.pcap -A console -n 10

Investigating multiple PCAPs with parameter " — pcap-list"

> sudo snort -c /etc/snort/snort.conf -q --pcap-list="icmp-test.pcap http2.pcap" -A console -n 10

Investigating multiple PCAPs with parameter " — pcap-show"

Snort will identify the traffic, distinguish each pcap file and prompts the alerts according to our ruleset.

> sudo snort -c /etc/snort/snort.conf -q --pcap-list="icmp-test.pcap http2.pcap" -A console –pcap

show

Answer the questions below

Investigate the mx-1.pcap file with the default configuration file.
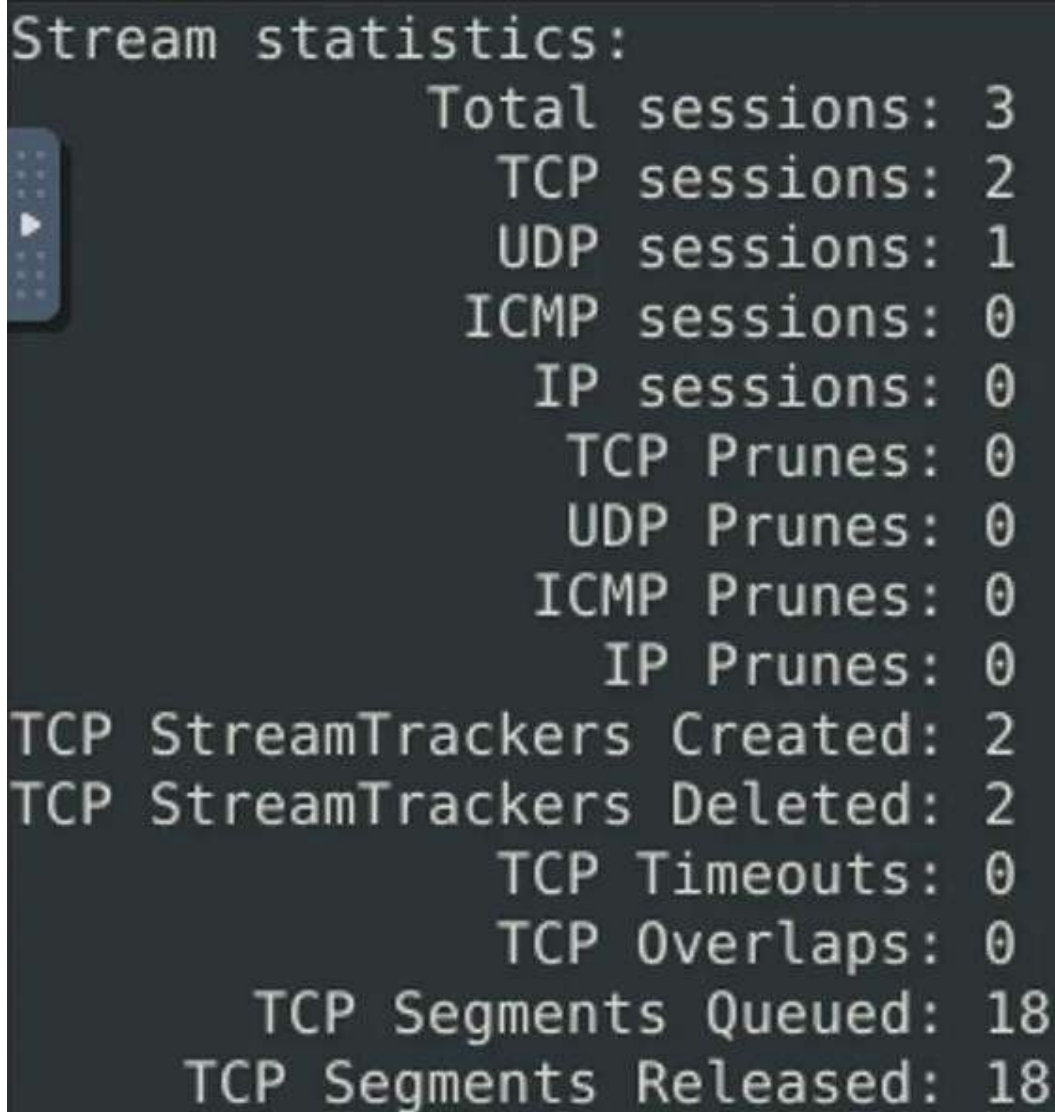
1. What is the number of the generated alerts? **Answer:** 170

sudo snort -c /etc/snort/snort.conf -A full -l . -r mx-1.pcap

```
          Alerts:         170 (147.826%)
          Logged:         170 (147.826%)
          Passed:           0 (  0.000%)
Limits:
           Match:           0
           Queue:           0
             Log:           0
           Event:           0
           Alert:           0
Verdicts:
           Allow:         115 (100.000%)
           Block:           0 (  0.000%)
         Replace:           0 (  0.000%)
       Whitelist:           0 (  0.000%)
       Blacklist:           0 (  0.000%)
          Ignore:           0 (  0.000%)
           Retry:           0 (  0.000%)
```

2. Keep reading the output. How many TCP Segments are Queued?

Answer: 18

```
Stream statistics:
              Total sessions: 3
                TCP sessions: 2
                UDP sessions: 1
               ICMP sessions: 0
                 IP sessions: 0
                  TCP Prunes: 0
                  UDP Prunes: 0
                 ICMP Prunes: 0
                   IP Prunes: 0
TCP StreamTrackers Created: 2
TCP StreamTrackers Deleted: 2
                TCP Timeouts: 0
                TCP Overlaps: 0
          TCP Segments Queued: 18
        TCP Segments Released: 18
```

Keep reading the output.
3. How many "HTTP response headers" were extracted?

Answer: 3

```
HTTP Inspect - encodings (Note: stream-reassembled packets included):
    POST methods:                                0
    GET methods:                                 2
    HTTP Request Headers extracted:              2
    HTTP Request Cookies extracted:              0
    Post parameters extracted:                   0
    HTTP response Headers extracted:             3
    HTTP Response Cookies extracted:             0
    Unicode:                                     0
    Double unicode:                              0
    Non-ASCII representable:                      0
    Directory traversals:                        0
    Extra slashes ("//"):                        1
    Self-referencing paths ("./"):               0
    HTTP Response Gzip packets extracted: 1
    Gzip Compressed Data Processed:         1272.00
    Gzip Decompressed Data Processed:       3608.00
    Total packets processed:                24
```

Investigate the mx-1.pcap file with the second configuration file.

> sudo snort -c /etc/snort/snortv2.conf -A full -l . -r mx-1.pcap

4. What is the number of the generated alerts? **Answer:**

68

```
Action Stats:
     Alerts:                  68 ( 59.130%)
     Logged:                  68 ( 59.130%)
     Passed:                   0 (  0.000%)
Limits:
        Match:                 0
        Queue:                 0
          Log:                 0
        Event:                 0
        Alert:                 0
Verdicts:
        Allow:              115 (100.000%)
        Block:                 0 (  0.000%)
      Replace:                 0 (  0.000%)
    Whitelist:                 0 (  0.000%)
    Blacklist:                 0 (  0.000%)
       Ignore:                 0 (  0.000%)
        Retry:                 0 (  0.000%)
```

Investigate the mx-2.pcap file with the default configuration file.

sudo snort -c /etc/snort/snort.conf -A full -l . -r mx-2.pcap

What is the number of the generated alerts?

Answer: 340

```
Action Stats:
      Alerts:              340 (147.826%)
      Logged:              340 (147.826%)
      Passed:                0 (  0.000%)
Limits:
       Match:                0
       Queue:                0
         Log:                0
       Event:                0
       Alert:                0
Verdicts:
       Allow:              230 (100.000%)
       Block:                0 (  0.000%)
     Replace:                0 (  0.000%)
   Whitelist:                0 (  0.000%)
   Blacklist:                0 (  0.000%)
      Ignore:                0 (  0.000%)
       Retry:                0 (  0.000%)
```

Keep reading the output.
5. What is the number of the detected TCP packets?

Answer: 82

```
Breakdown by protocol (includes rebuilt packets):
         Eth:          230 (100.000%)
        VLAN:            0 (  0.000%)
         IP4:          222 ( 96.522%)
        Frag:            0 (  0.000%)
        ICMP:          136 ( 59.130%)
         UDP:            4 (  1.739%)
         TCP:           82 ( 35.652%)
         IP6:            0 (  0.000%)
```

Investigate the mx-2.pcap and mx-3.pcap files with the default configuration file.

sudo snort -c /etc/snort/snort.conf -A full -l . --pcap-list="mx-2.pcap mx-3.pcap"

6. What is the number of the generated alerts?

Answer: 1020

```
Action Stats:
       Alerts:         1020 (147.826%)
       Logged:         1020 (147.826%)
       Passed:            0 (  0.000%)
Limits:
        Match:            0
        Queue:            0
          Log:            0
        Event:            0
        Alert:            0
```

TASK 9: SNORT RULE STRUCTURE

Understanding the Snort rule format is essential for any blue and purple teams. The primary structure of the snort rule is shown below

The following rule will generate an alert for each ICMP packet processed by snort;
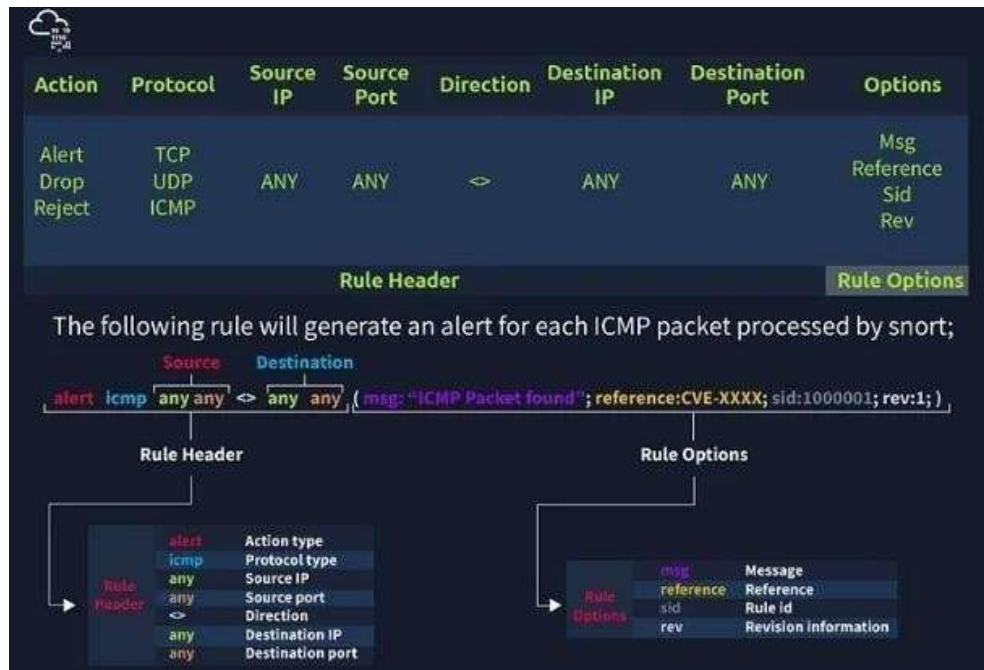
Remember, once you create a rule, it is a local rule and should be in your "local.rules" file. This file is located under "**/etc/snort/rules/local.rules**". A quick reminder on how to edit your local rules is shown below.

> sudo gedit /etc/snort/rules/local.rules

In this task, the default Snort rules have been deactivated and the location of rule to be applied is in the current working directory.

Use the attached VM and navigate to the Task-Exercises/Exercise-Files/TASK-9 folder to answer the questions! Note that you can use the following command to create the logs in the current directory: -l .

Use **"task9.pcap"**

1.   Write a rule to filter IP ID "35369" and run it against the given pcap file. What is the request name of the detected packet?

> sudo snort -c local.rules -A full -l . -r task9.pcap

Answer: TIMESTAMP REQUEST

Before we run the command, we need to edit the rule to filter IP ID "35369". Refer to the section above for Non-Payload Detection Rule Options. We will create only one rule.

sudo nano local.rules

- *alert tcp any any <> any any (msg:"ID Test";id:35369;sid:10000000001; rev:1;)*

Let's now run Snort. Observe that it read only the rule we have applied.

Clear the previous log and alarm files and deactivate/comment out the old rule

2.   Create a rule to filter **packets with Syn flag** and run it against the given pcap file. What is the number of detected packets?

Answer: 1

Again, refer to the Non-Payload Detection Rule Options. We will include the Option "flags" with a value of "S" to detect SYN flags.

        alert tcp any any <> any any (msg:"FLAG TEST";flags:S;sid:10000000002; rev:1)

Let's run Snort.

        sudo snort -c local.rules -A full -l . -r task9.pcap

The alert file would confirm that only one packet was detected.

Clear the previous log and alarm files and deactivate/comment out the old rule.

3. Write a rule to filter **packets with Push-Ack flags** and run it against the given pcap file. What is the number of detected packets?

Answer: 216

We just need to change the value of the option "flags" to "PA" to detect Push-Ack flags.

>   alert tcp any any <> any any (msg:"Push-Ack FLAG TEST";flags:PA;sid:10000000003; rev:1)

- Run Snort. Modified a bit with "-q" so it won't display

  results in the screen. sudo snort -c local.rules -A full -l -q . -

  r task9.pcap

Again, there are ways on how to determine the detected flags. One, is by reading from the log file created.

>   sudo snort -r snort.log.1689840434

Or from the alert file that was created. We will concatenate the file, then grep some of the keywords we used in the option, and then count the results by line.

<p align="center">cat alert | grep "Push-Ack" | wc -l</p>

Clear the previous log and alarm files and deactivate/comment out the old rule.

4. Create a rule to filter **packets with the same source and destination IP** and run it against the given pcap file. What is the number of detected packets?

Answer: 10

Refer on the Non-Payload Rule Options on SameIP. We will be using the option "sameip". alert

ip any any <> any any (msg:"SAME IP TEST";sameip;sid:10000000004; rev:1)

Run the command as above to start Snort detecting. Then look for the result. Initially I got 13, but he hint says we need to filter TCP and UDP.

Run again Snort then read the alert or log file.

5. Case Example — An analyst modified an existing rule successfully. Which rule option must the analyst change after the implementation?

Answer: rev

As the rules are modified for performance and efficiency issues, "rev" number will change too.

TASK 10: SNORT2 OPERATION LOGIC: POINTS TO REMEMBER

Let's start with overviewing the main configuration file (snort.conf)

      sudo gedit /etc/snort/snort.conf

Navigate to the "Step #1: Set the network variables." section.

This section manages the scope of the detection and rule paths.

TAG NAME	INFO	EXAMPLE
HOME_NET	That is where we are protecting.	'any' OR '192.168.1.1/24'
EXTERNAL_NET	This field is the external network, so we need to keep it as 'any' or '!$HOME_NET'.	'any' OR '!$HOME_NET'
RULE_PATH	Hardcoded rule path.	/etc/snort/rules
SO_RULE_PATH	These rules come with registered and subscriber rules.	$RULE_PATH/so_rules
PREPROC_RULE_PATH	These rules come with registered and subscriber rules.	$RULE_PATH/plugin_rules

Navigate to the "Step #2: Configure the decoder." section.

In this section, you manage the IPS mode of snort. The single-node installation model IPS model works best with "afpacket" mode. You can enable this mode and run Snort in IPS

TAG NAME	INFO	EXAMPLE
#config daq:	IPS mode selection.	afpacket
#config daq_mode:	Activating the inline mode	inline
#config logdir:	Hardcoded default log path.	/var/logs/snort

Task 11: Conclusion

In this room, we covered Snort, what it is, how it operates, and how to create and use the rules to investigate threats.

Answer the questions below

Which IDS or IPS type can help you stop the threats on a local machine?

> HIPS   ✓ Correct Answer

Which IDS or IPS type can help you detect threats on a local network?

> NIDS   ✓ Correct Answer

Which IDS or IPS type can help you detect the threats on a local machine?

> HIDS   ✓ Correct Answer

Which IDS or IPS type can help you stop the threats on a local network?

> NIPS   ✓ Correct Answer

Which described solution works by detecting anomalies in the network?

> NBA   ✓ Correct Answer

According to the official description of the snort, what kind of NIPS is it?

> full-blown   ✓ Correct Answer

NBA training period is also known as ...

> baselining   ✓ Correct Answer

Run the Snort instance and check the build number.

> 149   ✓ Correct Answer   💡 Hint

Test the current instance with "/etc/snort/snort.conf" file and check how many rules are loaded with the current build.

> 4151   ✓ Correct Answer   💡 Hint

Test the current instance with "/etc/snort/snortv2.conf" file and check how many rules are loaded with the current build.

> 1   ✓ Correct Answer   💡 Hint

Investigate the traffic with the default configuration file **with ASCII mode.**

```
sudo snort -dev -K ASCII -l .
```
Execute the traffic generator script and choose **"TASK-6 Exercise"**. Wait until the traffic ends, then stop the Snort instance. Now analyse the output summary and answer the question.

```
sudo ./traffic-generator.sh
```
Now, you should have the logs in the current directory. Navigate to folder "**145.254.160.237**". What is the source port used to connect port 53?

3009	✓ Correct Answer	♀ Hint

Use **snort.log.1640048004**

Read the snort.log file with Snort; what is the IP ID of the 10th packet?

```
snort -r snort.log.1640048004 -n 10
```

49313	✓ Correct Answer	♀ Hint

Read the "**snort.log.1640048004**" file with Snort; what is the referer of the 4th packet?

http://www.ethereal.com/development.html	✓ Correct Answer	♀ Hint

Read the "**snort.log.1640048004**" file with Snort; what is the Ack number of the 8th packet?

0x38AFFFF3	✓ Correct Answer

Read the "**snort.log.1640048004**" file with Snort; what is the number of the **"TCP port 80"** packets?

41	✓ Correct Answer	♀ Hint

Investigate the traffic with the default configuration file.

```
sudo snort -c /etc/snort/snort.conf -A full -l .
```
Execute the traffic generator script and choose **"TASK-7 Exercise"**. Wait until the traffic stops, then stop the Snort instance. Now analyse the output summary and answer the question.

```
sudo ./traffic-generator.sh
```
What is the number of the detected HTTP GET methods?

2	✓ Correct Answer	♀ Hint

You can practice the rest of the parameters by using the traffic-generator script.

No answer needed	✓ Correct Answer

Investigate the **mx-1.pcap** file with the default configuration file.

```
sudo snort -c /etc/snort/snort.conf -A full -l . -r mx-1.pcap
```
What is the number of the generated alerts?

170	✓ Correct Answer

Keep reading the output. How many TCP Segments are Queued?

18	✓ Correct Answer

Keep reading the output.How many "HTTP response headers" were extracted?

3	✓ Correct Answer

Investigate the **mx-1.pcap** file **with the second** configuration file.

```
sudo snort -c /etc/snort/snortv2.conf -A full -l . -r mx-1.pcap
```
What is the number of the generated alerts?

68	✓ Correct Answer

Investigate the **mx-2.pcap** file with the default configuration file.

```
sudo snort -c /etc/snort/snort.conf -A full -l . -r mx-2.pcap
```
What is the number of the generated alerts?

340	✓ Correct Answer	⚡ Hint

Keep reading the output. What is the number of the detected TCP packets?

82	✓ Correct Answer

Investigate the **mx-2.pcap and mx-3.pcap** files with the default configuration file.

```
sudo snort -c /etc/snort/snort.conf -A full -l . --pcap-list="mx-2.pcap mx-3.pcap"
```
What is the number of the generated alerts?

1020	✓ Correct Answer

Use "task9.pcap". Write a rule to filter IP ID "35369" and run it against the given pcap file. What is the request name of the detected packet? You may use this command: "snort -c local.rules -A full -l . -r task9.pcap"

TIMESTAMP REQUEST	✓ Correct Answer	⚡ Hint

Clear the previous alert file and comment out the old rules. Create a rule to filter packets with **Syn** flag and run it against the given pcap file. What is the number of detected packets?

1	✓ Correct Answer

Clear the previous alert file and comment out the old rules. Write a rule to filter packets with **Push-Ack** flags and run it against the given pcap file. What is the number of detected packets?

216	✓ Correct Answer

Clear the previous alert file and comment out the old rules. Create a rule to filter **UDP** packets with the same source and destination IP and run it against the given pcap file. What is the number of packets that show the same source and destination address?

7	✓ Correct Answer

Case Example - An analyst modified an existing rule successfully. Which rule option must the analyst change after the implementation?

rev	✓ Correct Answer

RESULT:

        In this room, we covered snort, hat it is, how it operates and how to correct and use the rules to investigate threats

LOG ANALYSIS FOR DETECTION AND RESPONSE

EXP.NO: 9   **DATE:25-03-2025**

AIM:

        The primary aim of the Log Analysis for Detection and Response is to equip learners with the knowledge and practical skills required to analyze system and network logs effectively. This is to identify potential security incidents, respond to threats, and enhance the overall security posture of an organization.

OBJECTIVE:

1. Introduction to Logs: A log is a stream of time-sequenced messages that record occurring events. Log analysis is the process of making sense of the events captured in the logs to paint a clear picture of what has happened across the infrastructure.

2. Importance of Logs:

    System Troubleshooting: Analyzing system errors and warning logs helps IT teams understand and quickly respond to system failures, minimizing downtime, and improving overall system reliability.

    Cyber Security Incidents: In the security context, logs are crucial in detecting and responding to security incidents. Firewall logs, intrusion detection system (IDS) logs, and system authentication logs, for

example, contain vital information about potential threats and suspicious activities. Performing log analysis helps SOC teams and Security Analysts identify and quickly respond to unauthorized access attempts, malware, data breaches, and other malicious activities.

Threat Hunting: On the proactive side, cyber security teams can use collected logs to actively search for advanced threats that may have evaded traditional security measures. Security Analysts and Threat Hunters can analyze logs to look for unusual patterns, anomalies, and indicators of compromise (IOCs) that might indicate the presence of a threat actor.

Compliance: Organizations must often maintain detailed records of their system's activities for regulatory and compliance purposes. Regular log analysis ensures that organizations can provide accurate reports and demonstrate compliance with regulations such as GDPR, HIPAA, or PCI DSS.

3. Different Types of Logs

TASK 1: INVESTIGATION THEORY

Understand the concepts of timelines, data visualisation and threat intelligence.

TASK 2: DETECTION ENGINEERING

This task encompasses common log file locations on Linux systems, common patterns for identifying suspicious behaviour, and common attack signatures.

TASK 3: AUTOMATED VS. MANUAL ANALYSIS

This short task explains the pros and cons of automated and manual analysis. Manual analysis is the process of examining data and artifacts without using automation tools, whereas automated analysis involves tools.