Mechanics of running snort - Answers

Exercise 1:

The version can be displayed with the –V command line switch:

snort –V

By running this command, it we find out that the installed version is:

```
student@student:~/Desktop/data/2-running-snort/run-snort$ snort -v
--------------------------------------------------
o")~   Snort++ 3.13.2.0
```

The table with basic snort command options is:

Snort option	Description
-c <configuration_file>	Run snort in IDS mode and use configuration file < configuration_file> which often has extension ".lua"
-r <input>	Read as input file <input> and process it
-R <rulefile>	Use rule file <rulefile> which often uses extension ".rules". You can also put your rules directly in your configuration file.
-l <folder>	Log alerts and logs to directory <folder>
-i <interf>	Receive as input traffic from network interface <interf>
-n <num>	Process the first <num> packets and exit
-d	application-layer payload
-e	link-layer headers
-L dump	Outputs decoded packet headers and ASCII/hex payload data. (Note: -L dump writes to stdout by default; redirect it to a file inside your folder using > /path/to/output_folder/packets.txt if needed)
-A cmp	the same as -A fast -d -e and will show information about the alert along with packet headers and payload.
-q	quite mode: suppresses the Snort startup banner and summary statistics, leaving only the clean packet dump
--bpf <filter>	Filters incoming traffic before Snort processes it using BPF (Berkeley Packet Filter)

Read more at https://snort-org-site.s3.amazonaws.com/production/release_files/files/000/038/976/original/snort_user.html?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAU7AK5ITMISWICI6U%2F20261118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20261118T084459Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=54dfc34542d6572af90f47899a70532dab66683302fdeab47aa44b3257e4c4e9

Exercise 2:

- What command option did you use?

snort -T -c snort.lua

- How many rules did Snort report were in the snort.conf file?

```
--------------------------------------------------
rule counts
       total rules loaded: 1
```

1 snort rule

Exercise 3:

The extracted packet is the following:

03/26-23:49:47.104231 10.10.10.101:1872 -> 10.10.10.100:**21**
TCP TTL:128 TOS:0x0 ID:43430 IpLen:20 DgmLen:48 DF
******S* Seq: 0x30D5AFFD Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => **MSS: 1460 NOP NOP SackOK**
=+

- What is the destination host's port number for this record? 21
- What TCP flags are set?
  SYN flag is set.
  Flags that are not set have the "*" in the TCP flag bit (12UAPRSF)
- What TCP options are set? MSS: 1460, NOP, NOP, SackOK
- What is the source MAC address of the first record? Use the –e command option to display the link layer (in this case Ethernet header).
  0:10:B5:39:C6:93
- What is the USER name supplied for this FTP connection?
  USER anonymous

03/26-23:49:50.699094 10.10.10.101:1872 -> 10.10.10.100:21
TCP TTL:128 TOS:0x0 ID:44454 IpLen:20 DgmLen:56 DF
AP Seq: 0x30D5AFFE Ack: 0xD2A6F989 Win: 0x21E8 TcpLen: 20 55
53 45 52 20 61 6E 6F 6E 79 6D 6F 75 73 0D 0A **USER anonymous**..

=+

```
pkt:8
eth(DLT):  00:10:B5:39:C6:93 -> 00:10:B5:39:C6:9A  type:0x0800
ipv4(0x0800):  10.10.10.101 -> 10.10.10.100
        Next:0x06 TTL:128 TOS:0x0 ID:44454 IpLen:20 DgmLen:56 DF
tcp(0x06):  ***AP***  SrcPort:1872  DstPort:21
        Seq: 0x30D5AFFE  Ack: 0xD2A6F989  Win: 0x21E8  TcpLen: 20

snort.raw[16]:
- - - - - - - - - - - - - -  - - - - - - - - - - - - - -  - - - - - - - - - -
55 53 45 52 20 61 6E 6F  6E 79 6D 6F 75 73 0D 0A  USER ano nymous..
- - - - - - - - - - - - - -  - - - - - - - - - - - - - -  - - - - - - - - - -
```

Exercise 4:

To view the packet mech, you run the command:

snort –r mech –v

- How many packets from the input file were TCP? UDP? ICMP?

20, 4, 6

Snort processed 30 packets.
===
Breakdown by protocol (includes rebuilt packets):

TCP: 20 (66.667%)
UDP: 4 (13.333%)
ICMP: 6 (20.000%)

```
--------------------------------------------------
codec
                    total: 30          (100.000%)
                      eth: 30          (100.000%)
                    icmp4: 6           ( 20.000%)
                     ipv4: 30          (100.000%)
                      tcp: 20          ( 66.667%)
                      udp: 4           ( 13.333%)
--------------------------------------------------
```

Exercise 5:

- Display only the ICMP packets from the input file mech. What command did you use?

  snort –r mech –v -q -L dump --bpf 'icmp'

  ```
  student@student:~/Desktop/data/2-running-snort/run-snort$ sudo snort -r mech -v -q -L dump --bpf 'icmp'
  ```

- How many echo request(s) and reply(s) are there?

  3 Echo requests, 3 Echo replies

  03/26-23:50:04.936104 10.10.10.101 -> 10.10.10.100
  ICMP TTL:1 TOS:0x0 ID:46502 IpLen:20 DgmLen:92
  Type:8 Code:0 ID:512 Seq:2304 **ECHO**
  =+

  03/26-23:50:04.936124 10.10.10.100 -> 10.10.10.101
  ICMP TTL:255 TOS:0x0 ID:5750 IpLen:20 DgmLen:92
  Type:0 Code:0 ID:512 Seq:2304 **ECHO REPLY**
  =+

  03/26-23:50:04.938241 10.10.10.101 -> 10.10.10.100
  ICMP TTL:1 TOS:0x0 ID:46758 IpLen:20 DgmLen:92
  Type:8 Code:0 ID:512 Seq:2560 **ECHO**
  =+

  03/26-23:50:04.938251 10.10.10.100 -> 10.10.10.101
  ICMP TTL:255 TOS:0x0 ID:5751 IpLen:20 DgmLen:92
  Type:0 Code:0 ID:512 Seq:2560 **ECHO REPLY**
  =+

  03/26-23:50:04.941180 10.10.10.101 -> 10.10.10.100
  ICMP TTL:1 TOS:0x0 ID:47014 IpLen:20 DgmLen:92
  Type:8 Code:0 ID:512 Seq:2816 **ECHO**

=+

03/26-23:50:04.941190 10.10.10.100 -> 10.10.10.101
ICMP TTL:255 TOS:0x0 ID:5752 IpLen:20 DgmLen:92
Type:0 Code:0 ID:512 Seq:2816 **ECHO REPLY**
=+

```
pkt:1
eth(DLT):  00:10:B5:39:C6:93 -> 00:10:B5:39:C6:9A  type:0x0800
ipv4(0x0800):  10.10.10.101 -> 10.10.10.100
        Next:0x01 TTL:1 TOS:0x0 ID:46502 IpLen:20 DgmLen:92
icmp4(0x01):  Type:8  Code:0
        ID:512   Seq:2304  ECHO

pkt:2
eth(DLT):  00:10:B5:39:C6:9A -> 00:10:B5:39:C6:93  type:0x0800
ipv4(0x0800):  10.10.10.100 -> 10.10.10.101
        Next:0x01 TTL:255 TOS:0x0 ID:5750 IpLen:20 DgmLen:92
icmp4(0x01):  Type:0  Code:0
        ID:512  Seq:2304  ECHO REPLY

pkt:3
eth(DLT):  00:10:B5:39:C6:93 -> 00:10:B5:39:C6:9A  type:0x0800
ipv4(0x0800):  10.10.10.101 -> 10.10.10.100
        Next:0x01 TTL:1 TOS:0x0 ID:46758 IpLen:20 DgmLen:92
icmp4(0x01):  Type:8  Code:0
        ID:512   Seq:2560  ECHO
```

```
pkt:4
eth(DLT):  00:10:B5:39:C6:9A -> 00:10:B5:39:C6:93  type:0x0800
ipv4(0x0800):  10.10.10.100 -> 10.10.10.101
        Next:0x01 TTL:255 TOS:0x0 ID:5751 IpLen:20 DgmLen:92
icmp4(0x01):  Type:0  Code:0
        ID:512  Seq:2560  ECHO REPLY

pkt:5
eth(DLT):  00:10:B5:39:C6:93 -> 00:10:B5:39:C6:9A  type:0x0800
ipv4(0x0800):  10.10.10.101 -> 10.10.10.100
        Next:0x01 TTL:1 TOS:0x0 ID:47014 IpLen:20 DgmLen:92
icmp4(0x01):  Type:8  Code:0
        ID:512   Seq:2816  ECHO

pkt:6
eth(DLT):  00:10:B5:39:C6:9A -> 00:10:B5:39:C6:93  type:0x0800
ipv4(0x0800):  10.10.10.100 -> 10.10.10.101
        Next:0x01 TTL:255 TOS:0x0 ID:5752 IpLen:20 DgmLen:92
icmp4(0x01):  Type:0  Code:0
        ID:512  Seq:2816  ECHO REPLY
```

Exercise 6:

To write all the UDP records to a binary file, and log traffic in the logme directory, you can use the following command:

mkdir logme
snort –r mech -v -q -L pcap --bpf 'udp' –l logme

```
student@student:~/Desktop/data/2-running-snort/run-snort$ mkdir logme
student@student:~/Desktop/data/2-running-snort/run-snort$ snort -r mech -v -q -L pcap --bpf 'udp' -l logme
```

- Try reading the newly created binary file as input to Snort in verbose mode to verify that you have collected only UDP records.

snort –r logme/<newly_created_file> -v -L dump

- How many UDP records were collected? Use either the statistics or count the packets to answer this.

4

```
-------------------------------------------------
codec
                      total: 4          (100.000%)
                        eth: 4          (100.000%)
                       ipv4: 4          (100.000%)
                        udp: 4          (100.000%)
-------------------------------------------------
```

- What are the IP identification numbers (ID) in the first two records that are displayed?

46246, 5749

```
pkt:1
eth(DLT):  00:10:B5:39:C6:93 -> 00:10:B5:39:C6:9A  type:0x0800
ipv4(0x0800):  10.10.10.101 -> 10.10.10.100
        Next:0x11 TTL:128 TOS:0x0 ID:46246 IpLen:20 DgmLen:78
udp(0x11):  SrcPort:137 DstPort:137 Len:50

pkt:2
eth(DLT):  00:10:B5:39:C6:9A -> 00:10:B5:39:C6:93  type:0x0800
ipv4(0x0800):  10.10.10.100 -> 10.10.10.101
        Next:0x11 TTL:64 TOS:0x0 ID:5749 IpLen:20 DgmLen:293
udp(0x11):  SrcPort:137 DstPort:137 Len:265
```

Exercise 7:

- Look at the alert file in lognids subdirectory. What is the message associated with the alert?

Attempted anonymous ftp connection

[**] [1:1111111:0] Anonymous FTP attempt [**]
[Priority: 0]
03/26-23:49:50.699094 10.10.10.101:1872 -> 10.10.10.100:21
TCP TTL:128 TOS:0x0 ID:44454 IpLen:20 DgmLen:56 DF
AP Seq: 0x30D5AFFE Ack: 0xD2A6F989 Win: 0x21E8 TcpLen: 20

```
student@student:~/Desktop/data/2-running-snort/run-snort$ ls lognids
alert_full.txt
student@student:~/Desktop/data/2-running-snort/run-snort$ cat lognids/alert_full.txt
[**] [1:1111111:0] "Anonymous FTP attempt" [**]
[Priority: 0]
03/26-23:49:50.699094 10.10.10.101:1872 -> 10.10.10.100:21
TCP TTL:128 TOS:0x0 ID:44454 IpLen:20 DgmLen:56 DF
***AP*** Seq: 0x30D5AFFE  Ack: 0xD2A6F989  Win: 0x21E8  TcpLen: 20
```