Mechanics of running Snort

In order to run this section, you need to open the provided virtual machine (**Ubuntu 26**) with VMWare Workstation. Use the following credentials to login:

Username: student
Password: student

**********Use the tcpdump binary file **mech** as input for this exercise. **********

Open a terminal and type the following command, in order to change directory:

```
cd  /home/student/Desktop/data/2-running-snort/run-snort/
```

Note: this document is tested with snort 3.13.2

Exercise 1:

Run Snort asking for help on the various command options. Use this command option for reference:

```
snort -h or snort -?
```

- What is the command option that you would use to discover the current version of Snort that you are running?
- Now, run Snort again to display the current version. What is your current version?
- Complete the table below from the help screen:

Snort Option	Description
-c <rulesfile>	
-r <input>	
-l <folder>	
-i <interf>	
-n <num>	
-L dump	
-d	
-e	
-T	
-A cmp	
--bpf	
-q	

Exercise 2:

Run Snort to check the configuration file "snort.conf" (test and report on the current Snort configuration). This will help you discover any errors that you have in your Snort configuration file before trying to run Snort in NIDS mode. The command option required to check the configuration file can be found by running Snort and asking it for help.

- What command option did you use?
- How many rules did Snort report were in the snort.conf file?

Exercise 3:

Now, let's read some data by running Snort and reading the input file **mech**. This is done by running the command:

```
snort -r mech –v –n 1 -L dump
```

The –r option filename informs Snort to read from a file, not a network interface. The –v option tells Snort to run in verbose mode (dump the packets). The –n option displays the first packet only (packet count of 1). -L dump (console packet dump).

- What is the destination host's port number for this record?
- What TCP flags are set?
- What TCP options are set?
- 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). Use the –e command option to display Ethernet header:

```
snort –r mech –v –n 1 -e -L dump
```

- Finally, dump the application layer of the eighth record. This will display the payload portion of the data. What option did you use to dump the application layer? It may be easier to examine this record if you "quiet" the statistics and banner using the –q option. What is the USER name supplied for this FTP connection?

```
snort  -r mech –v –n 8 –d -q -L dump
```

Exercise 4:

Description: Read from the input file mech again in verbose mode, but let's look at the Snort statistics displayed on the screen after all the records are displayed. These statistics can be helpful when you are interested in the mix of traffic, how Snort is performing, or for debugging purposes, to name a few of the reason you may examine them. According to the statistics, how many total packets were read from the input file.

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

Exercise 5:

As you will soon discover, Snort rules can look for all kinds of traits of a packet in an attempt to find anomalous network traffic. Sometimes, though, you may have to revert to using tcpdump filters if Snort doesn't have an option to examine a particular packet attribute. A tcpdump filter can be supplied at the end of the command. The example used here is trivial and a Snort rule could be written to do just this, but this example is used for demonstration purposes only. For instance, if you only wanted to see UDP records from the input file, the Snort command would be:

```
snort –r mech –v -q --bpf 'udp'-L dump
```

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

- Display only the ICMP packets from the input file mech. What command did you use?
- How many echo request(s) and reply(s) are there?

Exercise 6:

Instead of spewing data to the screen, Snort can save the data to a binary tcpdump-like file as you find in the file mech. In fact, for sites that have high throughput, Snort is usually run in a mode to write to a binary file and the traffic is then analyzed retrospectively. This makes Snort more efficient because it is just reading packets and writing – not analyzing them in near real-time.

Create a subdirectory to contain log files:

```
mkdir logme
```

Write all udp records to a binary file

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

Look in the logme subdirectory for a new file.

- Try reading the newly created binary file as input to Snort in verbose mode to verify that you have collected only UDP records.
- How many UDP records were collected? Use either the statistics or count the packets to answer this.
- What are the IP identification numbers (ID) in the first two records that are displayed?

Exercise 7:

Now, let's examine how Snort is run in NIDS mode, the mode that will tell you about anomalous traffic. Snort comes with a configuration file, snort.lua that contains configuration options and includes rules files for you to use. You will use an abbreviated snort.lua configuration file for this exercise. It simply has one rule in it:

alert tcp any any -> 10.10.10.100 21 (msg: "Anonymous FTP attempt"; content: "anonymous";)

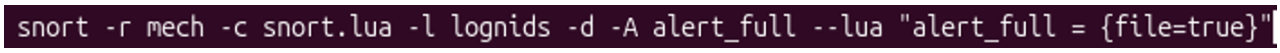
This rule says to "alert" on TCP traffic from any source IP and any source port to destination IP 10.10.10.100 destination port 21 (FTP) that has the word "anonymous" in the payload portion of the packet. If the rule triggers, the output from the alert will have a label of "Anonymous FTP attempt" to describe what the perceived problem is. The action "alert" says to put abbreviated output in the file "alert". It will also "log" the activity in a subdirectory with the name of the offending IP number.

If there is no folder 'lognids', create one

mkdir lognids

Run the command:

```
snort –r mech –c snort.lua –l lognids –d –A alert_full --lua
                    "alert_full = {file=true}"
```



- Open the directory lognids. What directories/files do you see?
- Look at the alert file. What is the message associated with the alert?