**Project 7 – Snort IDS/IPS, CSE 467/567**
**Department of Computer Science and Engineering, Miami University**
**100 points**

## Objectives

This lab provides hands-on experience using Snort to detect and prevent network traffic.

## Background:

Snort is an open souce Intrusion Detection/Prevention System (IDS/IPS). An IDS uses rules to passively monitor traffic and send alerts to the user or administrator when anomalies are detected, while an IPS is able to perform actions against anomalies such as dropping those packets.

Snort uses a rule-based language. The rule structure is as follows:

<Rule Actions> <Protocol> <Source IP Address> <Source Port><Direction Operator> <Destination IP Address> <Destination Port> (rule options: message, identification number, revision number)

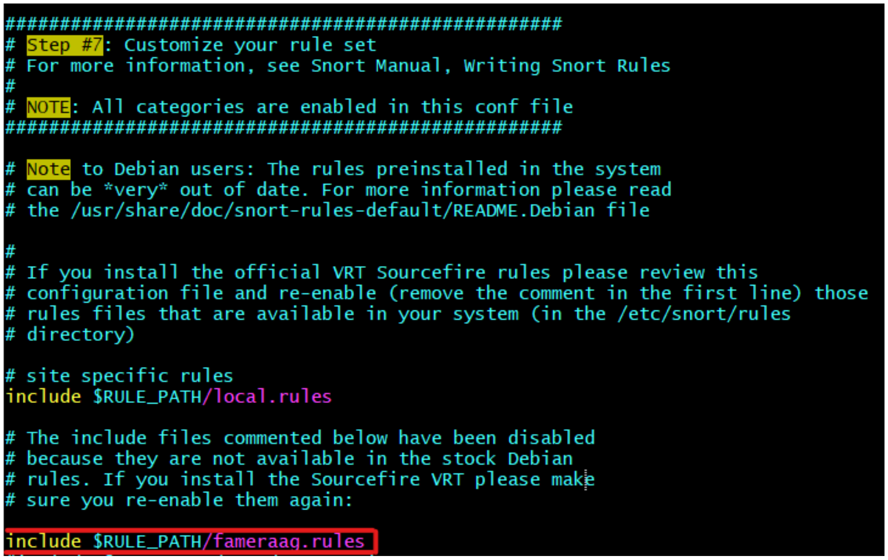
Most of the material in this lab is credited to Aversa Prentosito, a current graduate student at Miami that wrote a report for the Fall 2021 semester offering of CSE 460/560 (Ethical Hacking).

## Task 1 - Set Up Snort:

1. Log into your Kali VM
2. Type snort --help to ensure snort is installed in your machine
3. (5 points) List the files in /etc/snort. What are the names of the files that exist in this directory? **Copy/paste the names or submit a screenshot**.
    a. attribute_table.dtd file_magic.conf **rules** threshold.conf
    b. classification.config gen-msg.map snort.conf unicode.map
    c. community-sid-msg.map reference.config snort.debian.conf
4. (5 points) View the snort rules using ls /etc/snort/rules. Choose one of the rules in this directory and use cat to see its description and examples (i.e. cat /etc/snort/rules/backdoor.rules). **List the rule you chose and paste on of the examples from the output** (i.e. the line that starts with "alert").
    a. ICMP rule
    b. alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:3;)
    c.
5. Modify the $RULE_PATH in snort.conf. This is where you will set a path to the rules you are going to create in this lab.
    a. Use the command sudo vim /etc/snort/snort.conf to edit the conf file. Then in the file, scroll down to Step #7, then type include

All activities practiced in this course are done in a controlled laboratory environment. Doing any of these activities on computers for which you do not have permission **CAN BE CONSIDERED A CYBERCRIME** and you would face legal consequences

$RULE_PATH/<yourUID>.rules at the top of the list of "include" statements. <yourUID> is your miami unique ID. See example below

b. 

```
####################################################
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
####################################################

# Note to Debian users: The rules preinstalled in the system
# can be *very* out of date. For more information please read
# the /usr/share/doc/snort-rules-default/README.Debian file

#
# If you install the official VRT Sourcefire rules please review this
# configuration file and re-enable (remove the comment in the first line) those
# rules files that are available in your system (in the /etc/snort/rules
# directory)

# site specific rules
include $RULE_PATH/local.rules

# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:

include $RULE_PATH/fameraag.rules
```

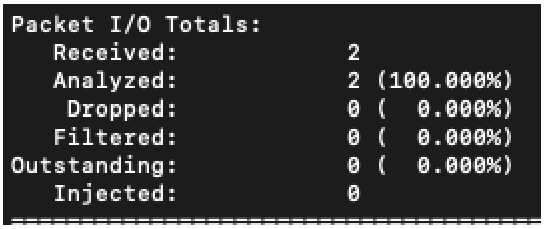
6. Modify the HOME_NET variable in snort.conf
   a. (5 points) Find out your Kali machine's IP address using ifconfig. **Copy your IP address here**.
      i. 172.17.13.47
   b. sudo vim /etc/snort/snort.conf
   c. Go to Step #1, and find the line that says ipvar HOME_NET any. Replace the "any" with your IP address.

## Task 2 - Alert for Ping (Partner Needed):

1. Create a rule to log ping packets
   a. vim into /etc/snort/rules/<yourUID>.rules
   b. In your .rules file, write log icmp any any -> $HOME_NET any (msg: "Ping detected"; sid:1000001; rev:1; classtype:icmp-event;). This rule will log ping packets. Here is a breakdown of what each part of the rule means:
      i. log = log the packet if criteria is met
      ii. icmp = protocol that SNORT analyzes
      iii. first any = source IP address
      iv. second any = source port number
      v. $HOME_NET = destination IP address (prentoam1 Kali machine)
      vi. third any = destination port
      vii. msg = message that you are alerted with
      viii. sid > 1000000 because up to 10000000 rules are registered
      ix. rev = revision of SNORT rule

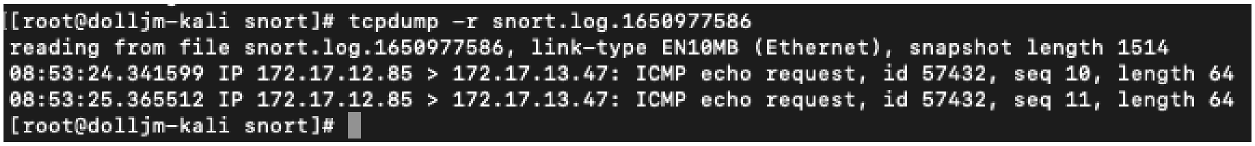
x. classtype = categorizes the rule into one of SNORT's categories

2. Restart Snort using sudo /etc/init.d/snort restart
3. Have your friend begin pinging your machine. Make sure he/she doesn't stop pinging you until you have completed Steps 4 and 5.
4. Start Snort using sudo snort -dev -c /etc/snort/snort.conf -l /var/log/snort/ -i eth0 -A full -k none. Below is a breakdown of the flags
   a. -c /etc/snort/snort.conf = specify config file with rules enabled
   b. -l /var/log/snort/ = specify the logging directory where alerts and logs are placed
   c. -i eth0 = specify interface SNORT listens on
   d. -A full = generate alerts using full alert mode
   e. -k none = disable SNORT's checksum verification, so packets are still inspected if the checksum hasn't been computed yet
5. Use CTRL + C to exit snort. Your friend can now stop pinging you.
6. View the snort logs in /var/log/snort
   a. sudo su to use root privileges
   b. cd /var/log/snort
   c. ls
   d. snort -r snort.log.<long number here>. For example, my log is "snort.log.1650574752"
7. (10) **How many ICMP packet did you capture? What IP address are the packets coming from (list the address, don't just say "my friend's IP")? Take a screenshot of the "Packet I/O Totals" block from the output.**
   **a. 2, 172.17.12.85**
   **b.**



```
Packet I/O Totals:
   Received:            2
   Analyzed:            2 (100.000%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:            0
===================================
```

8. (15 points) In the same /var/log/snort directory, run tcpdump -r snort.log.<long number here>. **Take a screenshot of the command and output**.



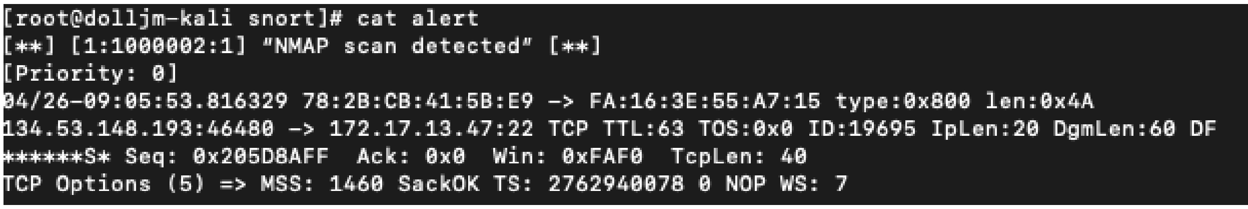
```
[root@dolljm-kali snort]# tcpdump -r snort.log.1650977586
reading from file snort.log.1650977586, link-type EN10MB (Ethernet), snapshot length 1514
08:53:24.341599 IP 172.17.12.85 > 172.17.13.47: ICMP echo request, id 57432, seq 10, length 64
08:53:25.365512 IP 172.17.12.85 > 172.17.13.47: ICMP echo request, id 57432, seq 11, length 64
[root@dolljm-kali snort]# 
```

# Task 3 - Alert for Nmap:

All activities practiced in this course are done in a controlled laboratory environment. Doing any of these activities on computers for which you do not have permission **CAN BE CONSIDERED A CYBERCRIME** and you would face legal consequences

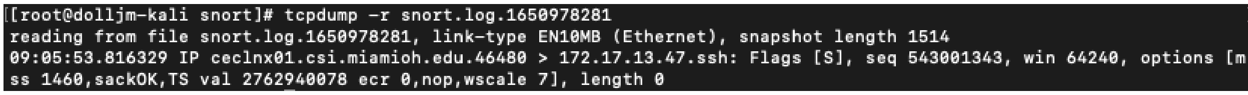
**Project 7 – Snort IDS/IPS, CSE 467/567**
**Department of Computer Science and Engineering, Miami University**
**100 points**

1. Create a rule to alet for NMAP SYN scan traffic
    a. vim into /etc/snort/rules/<yourUID>.rules
    b. Add the rule alert tcp any any -> $HOME_NET 22 (msg: “NMAP scan detected”; sid:1000002; rev:1; flags:S;)
2. Restart Snort using sudo /etc/init.d/snort restart
3. Start Snort using sudo snort -dev -c /etc/snort/snort.conf -l /var/log/snort/ -i eth0 -A full -k none.
4. Have your friend perform a SYN scan on your machine using the -sS and -p 22 flags (e.g. sudo nmap -sS -p 22 <ip address>)
5. Use CTRL + C to exit snort
6. Use sudo su and navigate to the /var/log/snort directory like in Task 2.
7. (15 points) Run cat alert. **Take a screenshot of the output.**
    a.



```
[root@dolljm-kali snort]# cat alert
[**] [1:1000002:1] "NMAP scan detected" [**]
[Priority: 0]
04/26-09:05:53.816329 78:2B:CB:41:5B:E9 -> FA:16:3E:55:A7:15 type:0x800 len:0x4A
134.53.148.193:46480 -> 172.17.13.47:22 TCP TTL:63 TOS:0x0 ID:19695 IpLen:20 DgmLen:60 DF
******S* Seq: 0x205D8AFF  Ack: 0x0  Win: 0xFAF0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 2762940078 0 NOP WS: 7
```

8. (15 points) Run tcpdump -r  snort.log.<long number here>. Keep in mind that this is a different log file than the one used in the previous task. You can use ls -l to see the dates and times of your log files so you can choose the most recent one. **Take a screenshot of the command and output.**
    a.



```
[[root@dolljm-kali snort]# tcpdump -r snort.log.1650978281
reading from file snort.log.1650978281, link-type EN10MB (Ethernet), snapshot length 1514
09:05:53.816329 IP ceclnx01.csi.miamioh.edu.46480 > 172.17.13.47.ssh: Flags [S], seq 543001343, win 64240, options [m
ss 1460,sackOK,TS val 2762940078 ecr 0,nop,wscale 7], length 0
```

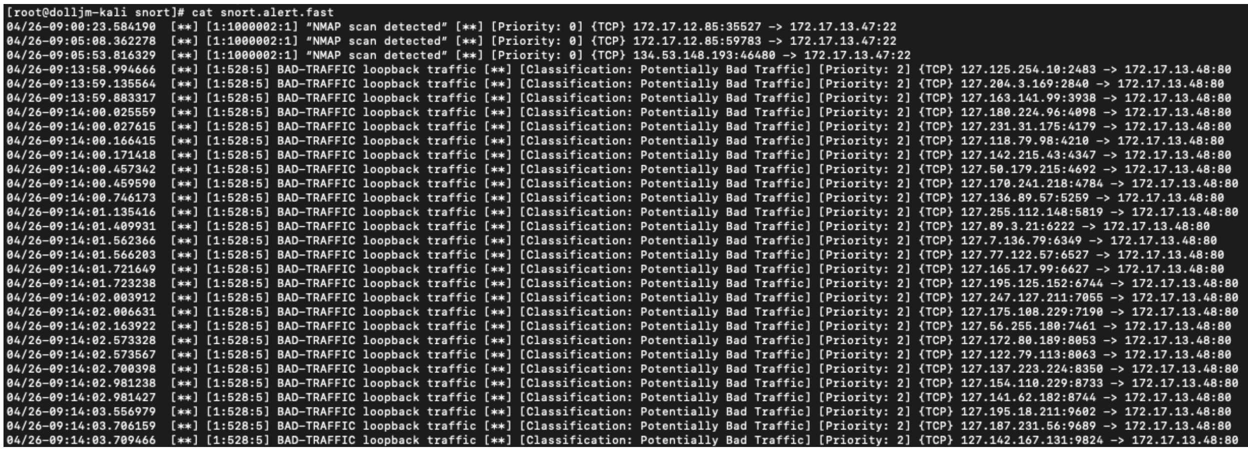
# Task 4 - Stop DoS Activity:

1. Create a rule to alert for packets targeting port 80
    a. vim into /etc/snort/rules/<yourUID>.rules
    b. Add the rule alert tcp any any -> $HOME_NET 80 (msg: “Possible DoS detected, block traffic”; sid:1000003; rev:1; detection_filter: track by_dst, count 25, seconds 5;)
2. Restart Snort using sudo /etc/init.d/snort restart
3. Start Snort using sudo snort -dev -c /etc/snort/snort.conf -l /var/log/snort/ -i eth0 -A full -k none
4. Have your friend perform a DoS attack on your machine. The attack will be a SYN flood attack, where the attacker quickly initiates connection without finalizing it.  The server then must wait for connections that are half-open, taking up resourcesand rendering the system unresponsive to traffic that is legitimate.

All activities practiced in this course are done in a controlled laboratory environment. Doing any of these activities on computers for which you do not have permission **CAN BE CONSIDERED A CYBERCRIME** and you would face legal consequences

   a. Have your friend run `sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source <your IP address>`. Run the command for 5-ish seconds
      i. Sending 15000 packets at size 120 bytes each
      ii. Enable SYN flag (-S)
      iii. TCP window size of 64
      iv. Attack is directed at defending machine's HTTP web server (port 80)
      v. Send packets as fast as possible (--flood)
      vi. Generate spoofed IP addresses to disguise the source (--rand-source)
   b. Use CTRL+C to end the attack

5. Use CTRL+C to stop Snort
6. cd to /var/log/snort as root
7. (30 points) Type cat snort.alert.fast. **Take a screenshot of the command and all the output.** The output should include both alerts from the nmap scan and the DoS attack.
   a.



```
[root@dolljm-kali snort]# cat snort.alert.fast
04/26-09:00:23.584190  [**] [1:1000002:1] "NMAP scan detected" [**] [Priority: 0] {TCP} 172.17.12.85:35527 -> 172.17.13.47:22
04/26-09:05:08.362278  [**] [1:1000002:1] "NMAP scan detected" [**] [Priority: 0] {TCP} 172.17.12.85:59783 -> 172.17.13.47:22
04/26-09:05:53.816329  [**] [1:1000002:1] "NMAP scan detected" [**] [Priority: 0] {TCP} 134.53.148.193:46480 -> 172.17.13.47:22
04/26-09:13:58.994666  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.125.254.10:2483 -> 172.17.13.48:80
04/26-09:13:59.135564  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.204.3.169:2840 -> 172.17.13.48:80
04/26-09:13:59.883317  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.163.141.99:3938 -> 172.17.13.48:80
04/26-09:14:00.025559  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.180.224.96:4098 -> 172.17.13.48:80
04/26-09:14:00.027615  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.231.31.175:4179 -> 172.17.13.48:80
04/26-09:14:00.166415  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.118.79.98:4210 -> 172.17.13.48:80
04/26-09:14:00.171418  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.142.215.43:4347 -> 172.17.13.48:80
04/26-09:14:00.457342  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.50.179.215:4692 -> 172.17.13.48:80
04/26-09:14:00.459590  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.170.241.218:4784 -> 172.17.13.48:80
04/26-09:14:00.746173  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.136.89.57:5259 -> 172.17.13.48:80
04/26-09:14:01.135416  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.255.112.148:5819 -> 172.17.13.48:80
04/26-09:14:01.409931  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.89.3.21:6222 -> 172.17.13.48:80
04/26-09:14:01.562366  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.7.136.79:6349 -> 172.17.13.48:80
04/26-09:14:01.566203  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.77.122.57:6527 -> 172.17.13.48:80
04/26-09:14:01.721649  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.165.17.99:6627 -> 172.17.13.48:80
04/26-09:14:01.723238  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.195.125.152:6744 -> 172.17.13.48:80
04/26-09:14:02.003912  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.247.127.211:7055 -> 172.17.13.48:80
04/26-09:14:02.006631  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.175.108.229:7190 -> 172.17.13.48:80
04/26-09:14:02.163922  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.56.255.180:7461 -> 172.17.13.48:80
04/26-09:14:02.573328  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.172.80.189:8053 -> 172.17.13.48:80
04/26-09:14:02.573567  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.122.79.113:8063 -> 172.17.13.48:80
04/26-09:14:02.700398  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.137.223.224:8350 -> 172.17.13.48:80
04/26-09:14:02.981238  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.154.110.229:8733 -> 172.17.13.48:80
04/26-09:14:02.981427  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.141.62.182:8744 -> 172.17.13.48:80
04/26-09:14:03.556979  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.195.18.211:9602 -> 172.17.13.48:80
04/26-09:14:03.706159  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.187.231.56:9689 -> 172.17.13.48:80
04/26-09:14:03.709466  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.142.167.131:9824 -> 172.17.13.48:80
```

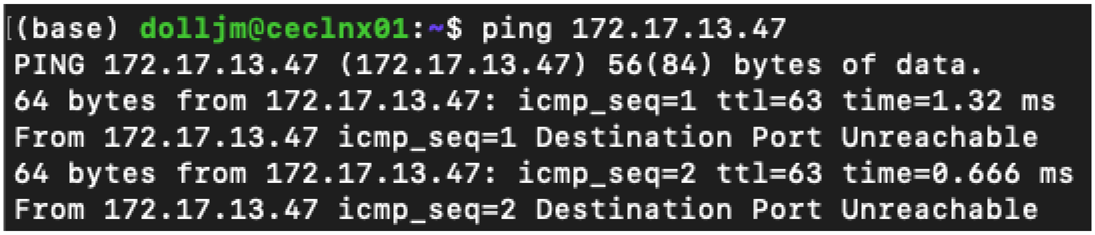
# (20 points) EXTRA CREDIT

Find out how to block a specific IP from reaching your website. For example, one friend should not be able to reach your website, but another friend with a different IP should. Provide proof of your snort rule, and show that one friend is unable to access your site while the other friend

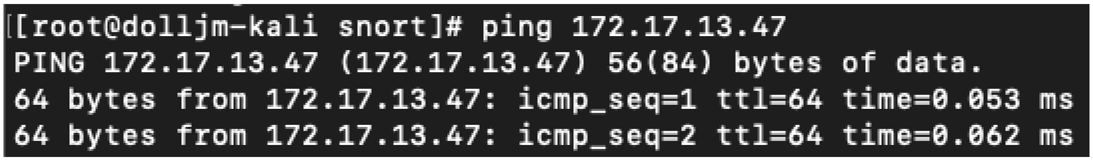
reject icmp 134.53.148.193 any -> $HOME_NET any (msg: "Blacklisted IP"; sid:1000005; rev:1;)

When I tried to ping my ip from my blocked ceclnx ip, the terminal said the "Destination unreachable", yet I was still able to successfully ping myself from my machine.

All activities practiced in this course are done in a controlled laboratory environment. Doing any of these activities on computers for which you do not have permission **CAN BE CONSIDERED A CYBERCRIME** and you would face legal consequences



```
[(base) dolljm@ceclnx01:~$ ping 172.17.13.47
 PING 172.17.13.47 (172.17.13.47) 56(84) bytes of data.
 64 bytes from 172.17.13.47: icmp_seq=1 ttl=63 time=1.32 ms
 From 172.17.13.47 icmp_seq=1 Destination Port Unreachable
 64 bytes from 172.17.13.47: icmp_seq=2 ttl=63 time=0.666 ms
 From 172.17.13.47 icmp_seq=2 Destination Port Unreachable
```



```
[[root@dolljm-kali snort]# ping 172.17.13.47
 PING 172.17.13.47 (172.17.13.47) 56(84) bytes of data.
 64 bytes from 172.17.13.47: icmp_seq=1 ttl=64 time=0.053 ms
 64 bytes from 172.17.13.47: icmp_seq=2 ttl=64 time=0.062 ms
```

## Submission:

- Submit on Canvas the required information in .doc, .docx, or .pdf format
- Please use the same numbering as in this document. Please **highlight, bold, or color your final answer. Make your solution VISIBLE.**

## Rubric

| **Description** | **Points** |
|---|---|
| Task 1 Q 3 | 5 |
| Task 1 Q 4 | 5 |
| Task 1 Q 6.a | 5 |
| Task 2 Q 7 | 10 |
| Task 2 Q 8 | 15 |
| Task 3 Q 7 | 15 |
| Task 3 Q 8 | 15 |
| Task 4 Q 7 | 30 |
| **Total** | **100** |

All activities practiced in this course are done in a controlled laboratory environment. Doing any of these activities on computers for which you do not have permission **CAN BE CONSIDERED A CYBERCRIME** and you would face legal consequences