**网络安全综合课程设计实验4**

57117204 姚晓

2020/9/12

**Task1: SYN Flooding Attack**

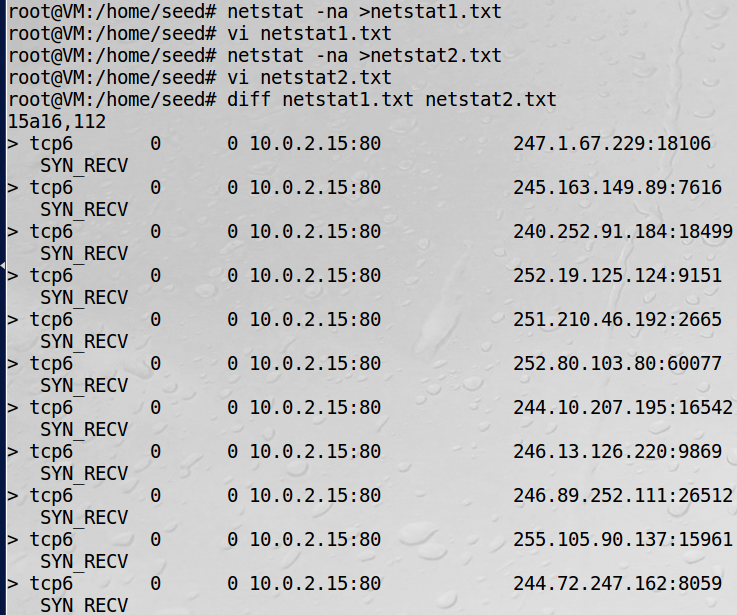
本次实验用两台主机Ubuntu01（IP地址为10.0.2.4）和Ubuntu02（IP地址为10.0.2.15）首先输入命令netstat –na并把结果保存在netstat1.txt（忘记了截图），然后关闭SYNcookie：



在Ubuntu01当中输入命令攻击Ubuntu02：



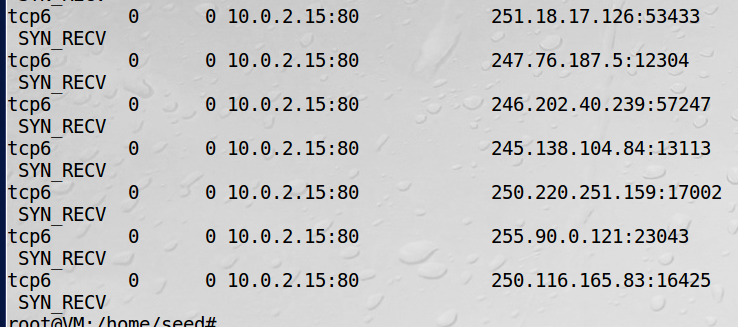
再次输入命令netstat –na并把结果保存在netstat2.txt，对比两个文件发现，多了很多SYN帧，但没有建立连接，并且攻击时Ubuntu02变得十分缓慢，内存占用大：



打开Ubuntu02的SYNcookie：



然后用netstat-pna查看，发现还是有很多SYN帧：

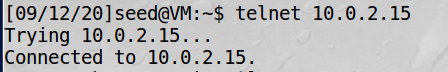


这是因为SYNcookie的工作机制为：当服务器受到第一次握手SYN信息是，将部分信息利用自己的密钥进行哈希闭关返回给客户端。当再次收到客户端的信息时，利用自己的密钥校验哈希值的准确性，即可判断这个客户端是之前发来第一次握手的客户端。服务器不会在SYN等待队列满了之后拒绝服务，而是通过cookie达到继续工作的效果。

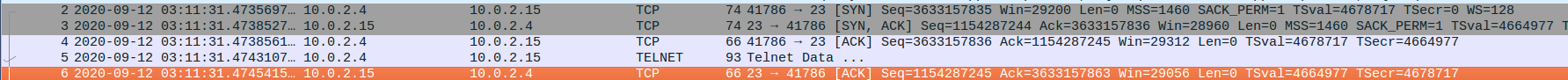
**Task2: TCP RST Attackson telnet and ssh Connections**

本次实验用Ubuntu01（IP地址10.0.2.4）作为客户端发起连接，Ubuntu02（IP地址10.0.15）作为服务端，Ubuntu03（IP地址10.0.2.5）作为攻击者。

首先用Ubuntu01发起telnet连接：

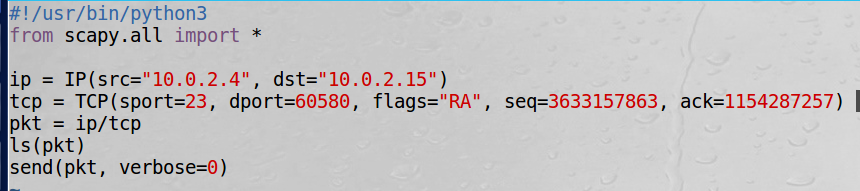


用Ubuntu03的wireshark抓包：



可以看到端口为41786，下一个seq应该为3633157863，ack应该为1154287257。

在Ubuntu03中构造攻击代码（成功的代码忘了截图，这是第一次失败版本的截图，当时忘记了改端口号，其他部分代码，没有变）：



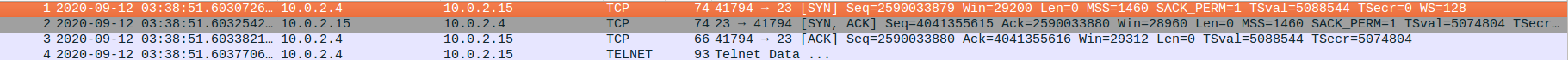
运行攻击代码，可以看到连接被断开：



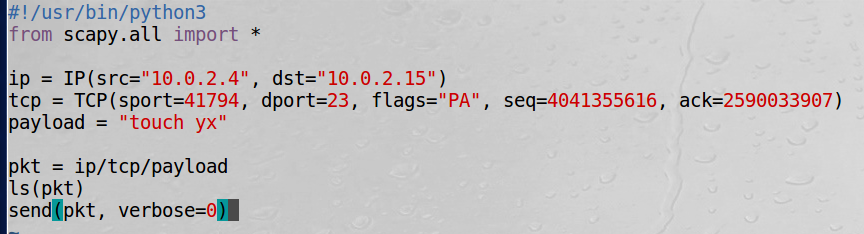
ssh连接也是同样的操作，但是我忘记了截图。

**Task4: TCP Session Hijacking**

本次实验用Ubuntu01（IP地址10.0.2.4）作为客户端发起连接，Ubuntu02（IP地址10.0.15）作为服务端，Ubuntu03（IP地址10.0.2.5）作为攻击者。

首先用Ubuntu01发起telnet连接，在Ubuntu03用wireshark抓包：

可以看到01的端口为41794，02的下一个seq应该为4041355616，ack应该为2590033907，然后再Ubuntu03构造攻击代码：



运行之后在Ubuntu02的目录下看到yx文件：

