网络与信息安全课内实验一对DDos攻击的理解

一．实验目的

1. 熟悉 Linux 系统,Wireshark 软件基本操作。

2. 熟悉影响网络数据包中影响网络通讯质量的关键参数。

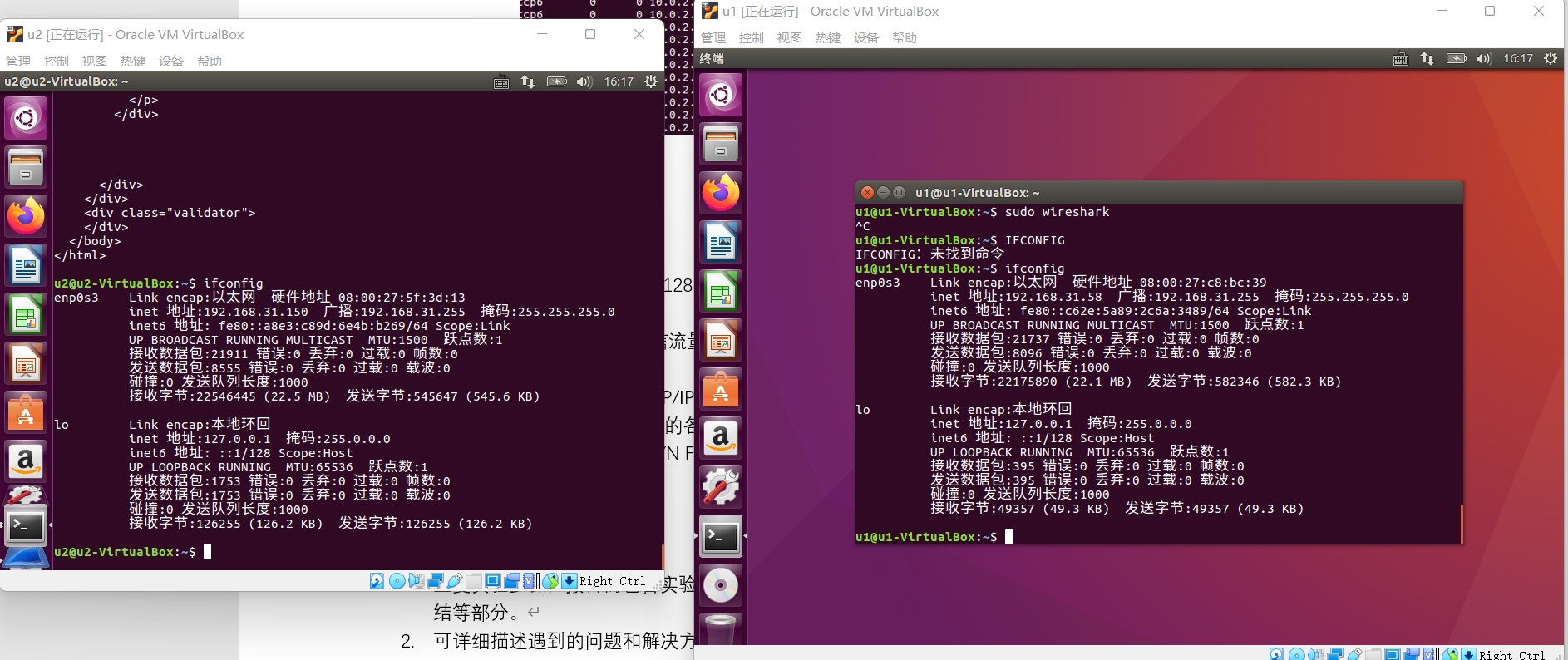
3. SYN 洪泛攻击的实现与观察。

二．实验原理

DoS（拒绝服务）：造成DoS的攻击行为被称为DoS攻击，它的目的是使得计算机或者网络我无法提供正常服务。SYN泛洪攻击是Dos攻击的一种，主要利用的就是TCP三次握手机制的缺陷。

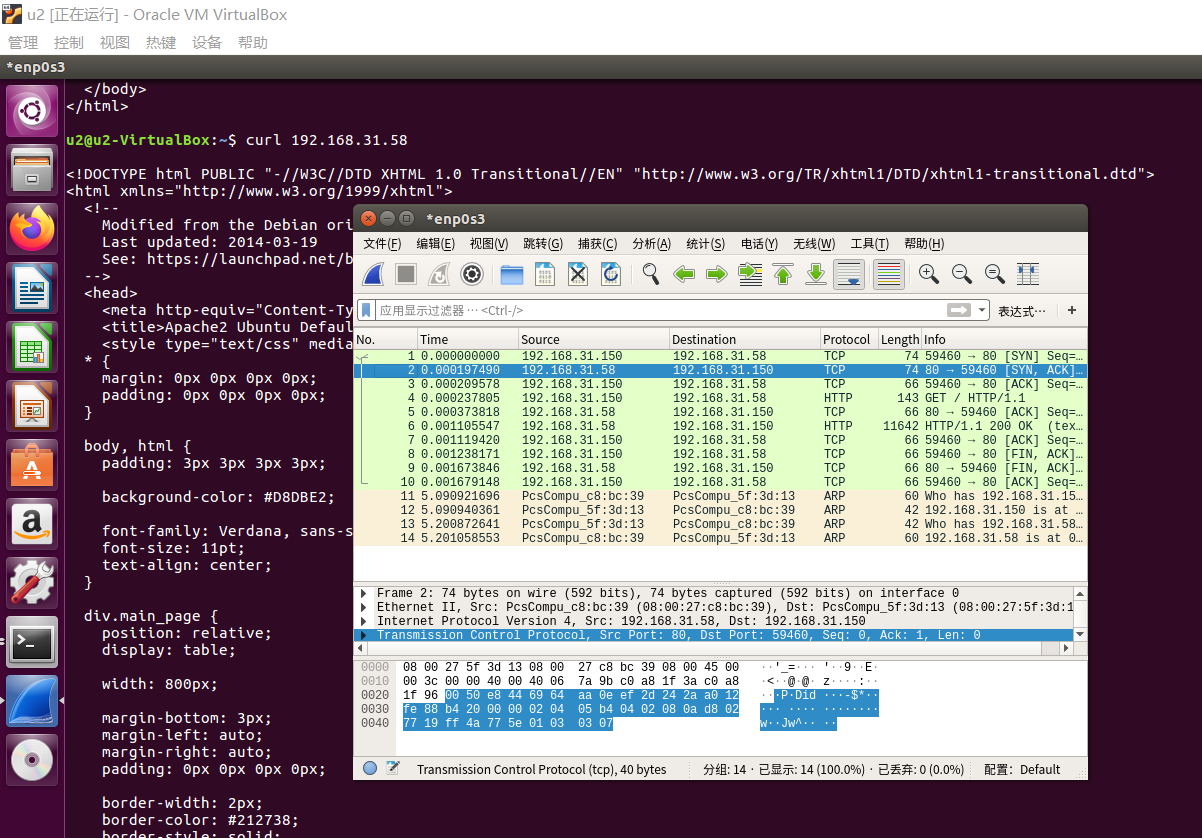
在SYN泛洪攻击中，Attacker发送TCP数据包至Server，SYN是TCP三次握手中的第一个数据包，而当这个Server返回ACK以后，Attacker不再进行确认，那这个连接就处在了一个挂起的半连接状态。由于服务器在处理TCP请求时，会在协议栈留一块缓冲区来存储握手的过程，当然如果超过一定的时间内没有接收到客户端的报文，本次连接在协议栈中存储的数据将会被丢弃。 Attacker如果利用这段时间发送大量的连接请求，全部挂起在半连接状态。这样将不断消耗服务器资源，直到拒绝服务，就会导致服务端较长时间内丧失服务功能——这样就形成了DoS攻击。这种攻击方式就称为SYN泛洪攻击。

三．实验步骤



两个虚拟机，u1为server，u2为attacker，u1的ip为192.168.31.58，u2的ip为192.168.31.150

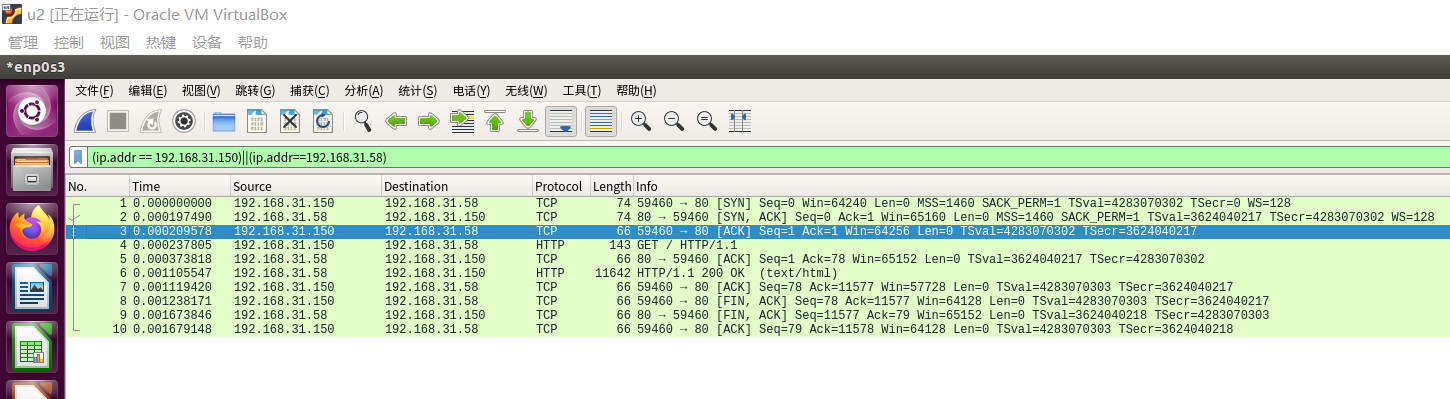
在u1上安装了apache2，并且u2能通过u1的ip进行访问



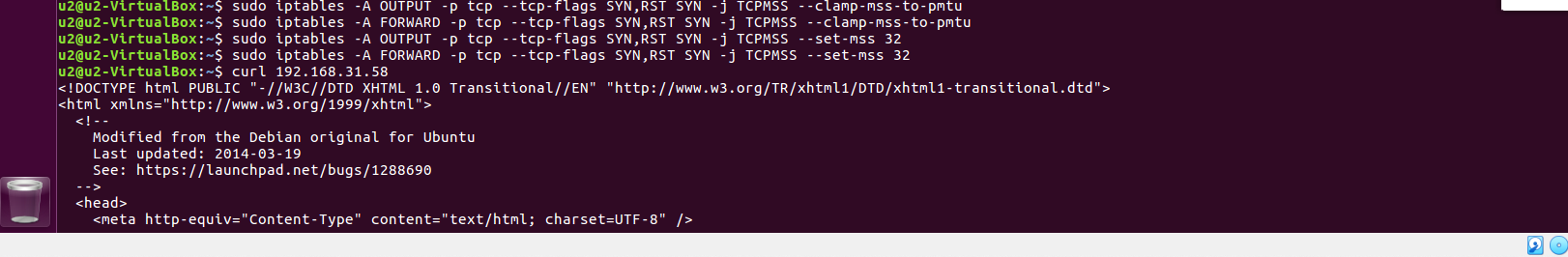
在不修改MSS值的情况下使用命令行工具curl访问server端的Web服务，并使用wireshark进行抓包

过滤器的使用：ip.addr==u1 || ip.addr==u2

或者使用(ip.src==u1&&ip.dst==u2)||(ip.src==u2&&ip.dst==u1)

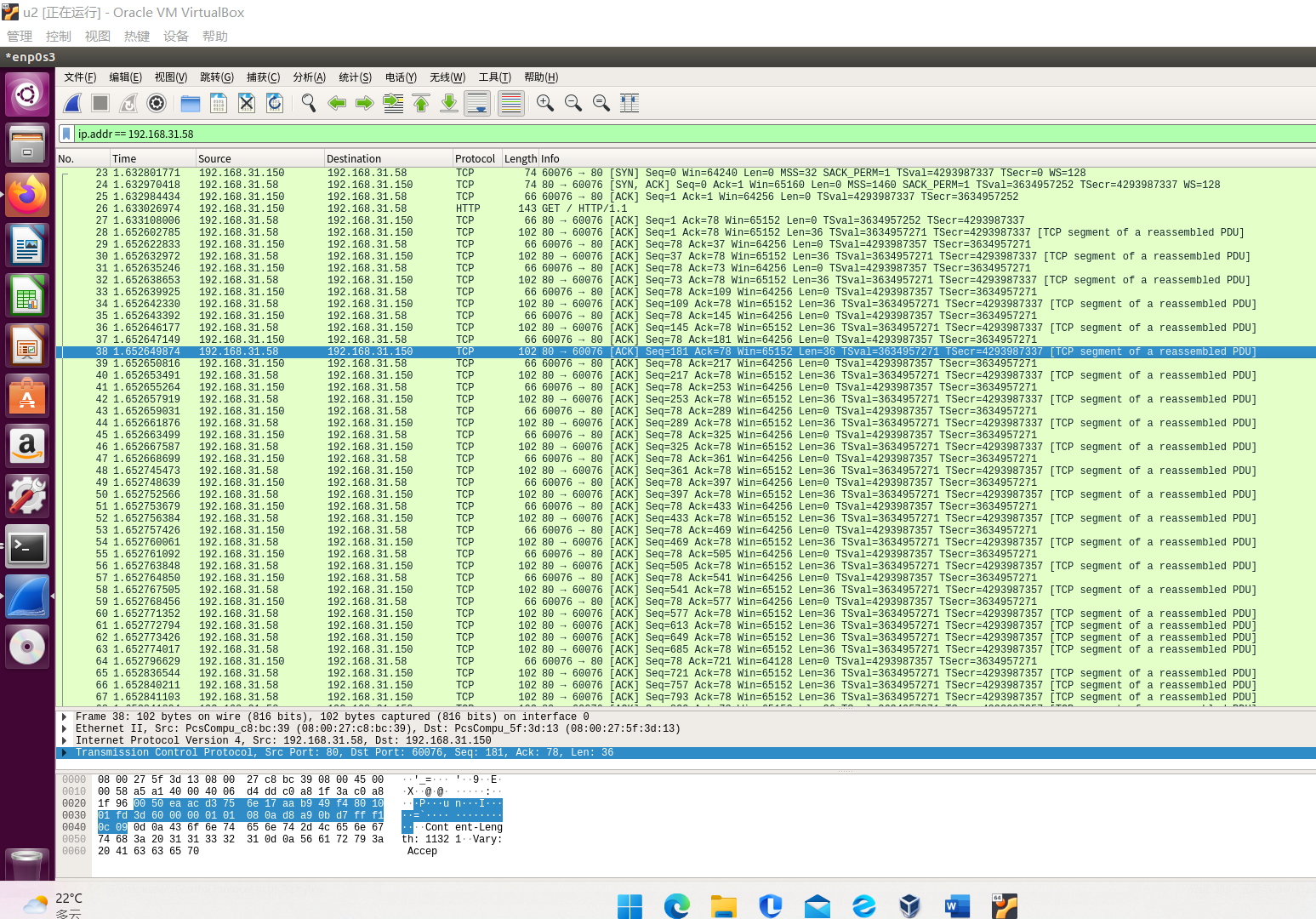


此时MSS的值为1460



使用iptables命令修改TCP通讯中的MSS值

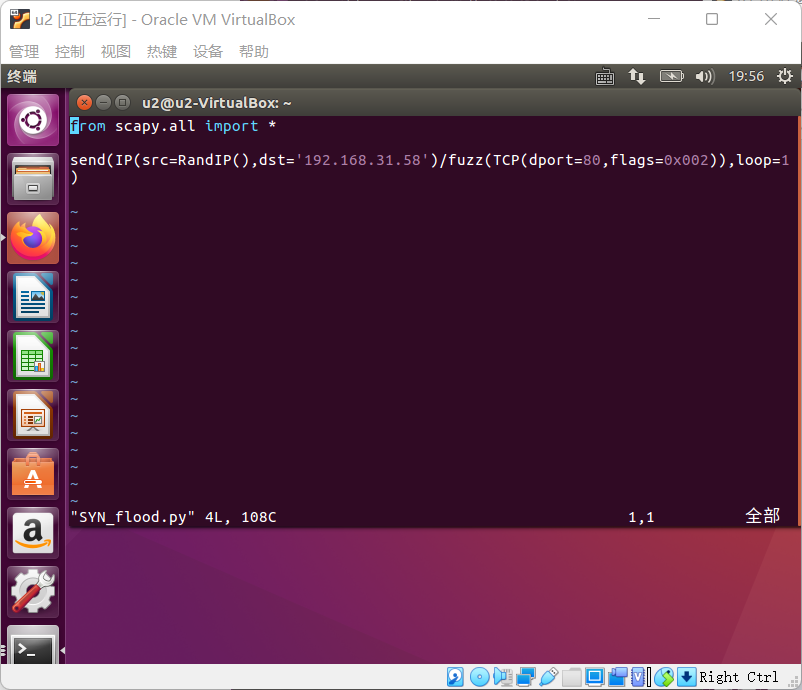
再次访问server端的Web服务



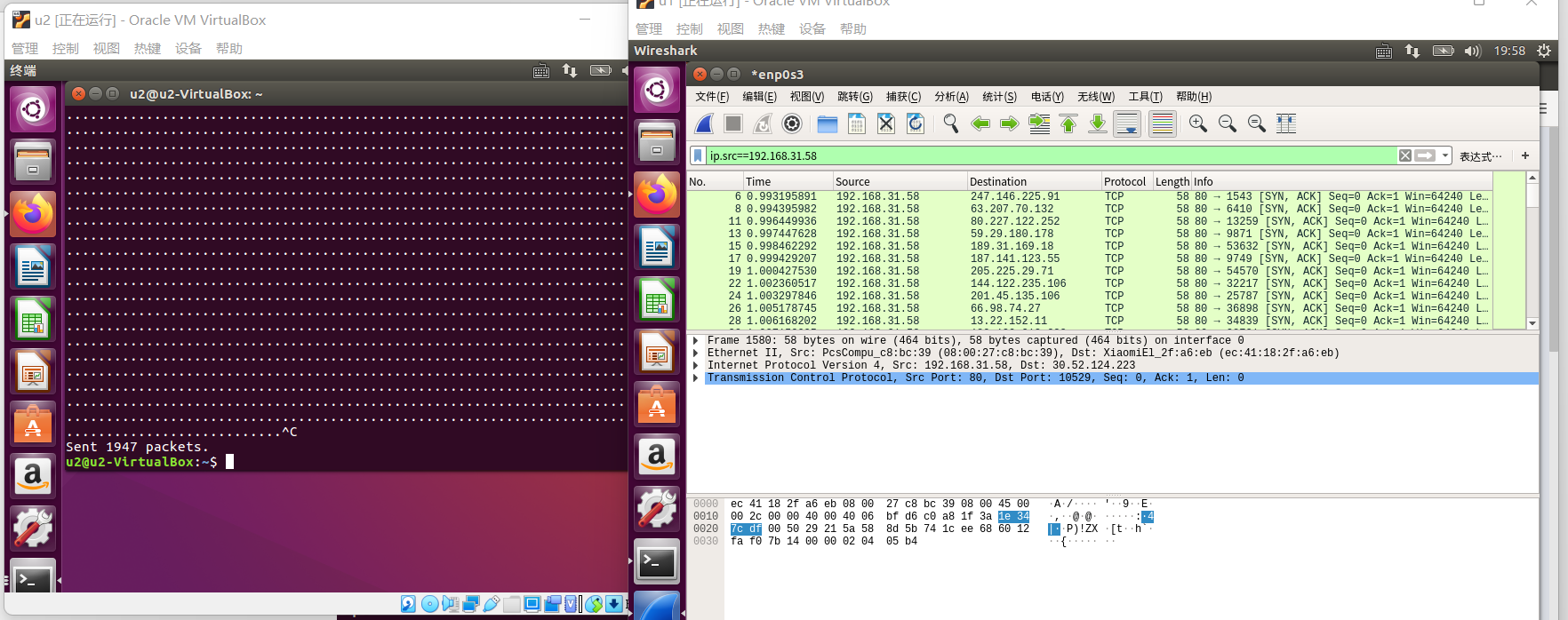
通过观察发现，MSS已经被修改为32

SYN-Flood 攻击：

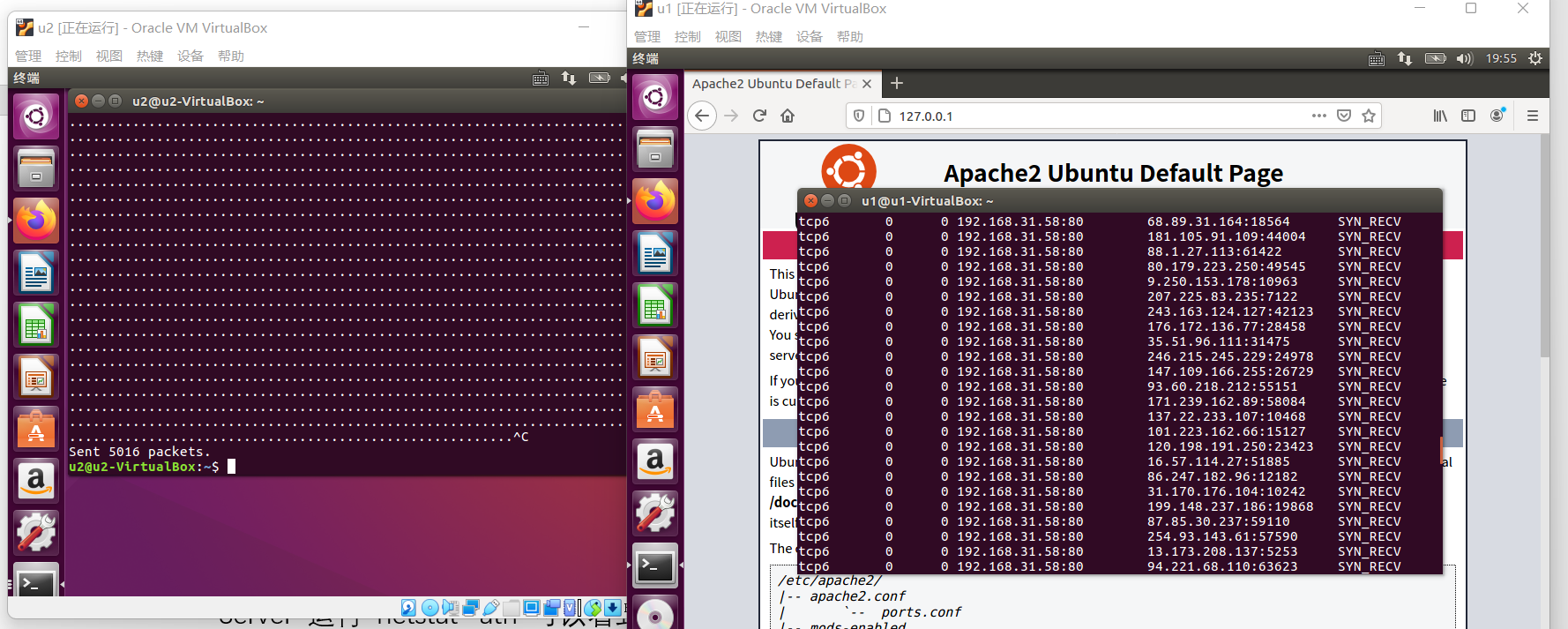
编写python程序‘



运行python程序并且在u1上通过wireshark抓包，可以看到server发出了很多的SYN请求



在u1上运行 netstat –atn，同样可以看到出现了很多SYN请求



四．思考题

1. 给出过滤ip地址为10.0.2.128的攻击机和10.0.2.129的服务器进行通信的数据包的wireshark过滤器语句。

答：(ip.addr==10.0.2.128) || (ip.addr==10.0.2.129) 或者使用

((ip.src==10.0.2.128)&&(ip.dst==10.0.2.129))||((ip.src==10.0.2.129)&&(ip.dst==10.0.2.128))

2. 和默认的未修改MSS的通信流量相比，修改MSS值过后的数据包数量是变多了还是变少了？

答：修改后MSS为32，修改前为1460，修改后的数据包数量比修改前变多了。

3. MTU和MSS分别工作在TCP/IP协议栈的哪一层，它们之间有什么关系？

答：MTU作用于数据链路层，MSS作用域传输层，表明传输层能够传输的数据大小。MUT往往会大于MSS, 可以简单理解为MTU= MSS+TCP层头部长度+IP层头部长度.

4. 对SYN\_flood.py那一行命令的各个参数进行注释。

答：send(IP(src=RandIP(),dst='192.168.31.58')/fuzz(TCP(dport=80,flags=0x002)),loop=1)

send()函数为Scapy库基于网络层的数据包发送服务，代码中该函数发送了IP协议和TCP协议的组合数据包。 IP包头下src 设置了发送端IP地址，RandIP()即随机IP地址；dst 设置了对端IP地址。TCP包头下dport 设置了对端目的端口，flags 为标志位，SYN信号该标志位即为2，也可直接设置为flags='S' 。 fuzz()函数可以在保证在对部分参数有要求的条件下（如本行的dport和flags），快速构造正确的随机模板，本行便快速构造了TCP数据包。而loop参数默认为0，如果值不为0则数据包将一直循环发送。

5. 查阅资料，给出几个防御SYN Flood攻击的手段。

答：1.首包丢弃：首包丢弃指的是FW在接收到SYN报文的时候直接将SYN报文丢弃，如果是正常的客户端那么它就会进行TCP的重传，FW会对其进行记录，如果进行了正常重传，那么就说明这个客户端时一个正常的客户端。2.SYN Cookie: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。3. 无效连接监视释放: 这种方法不停监视系统的半开连接和不活动连接，当达到一定阈值时拆除这些连接，从而释放系统资源。这种方法对于所有的连接一视同仁，而且由于SYN Flood造成的半开连接数量很大，正常连接请求也被淹没在其中被这种方式误释放掉，因此这种方法属于入门级的SYN Flood方法。

五．实验小结

本次实验仅实现了最简单的SYN洪泛攻击、且仅用wireshark进行了抓包分析。但感觉完成该实验已经让我对TCP协议三次握手原则以及协议参数有个较深刻的认识。过程中也逐步熟悉了linux系统的操作。