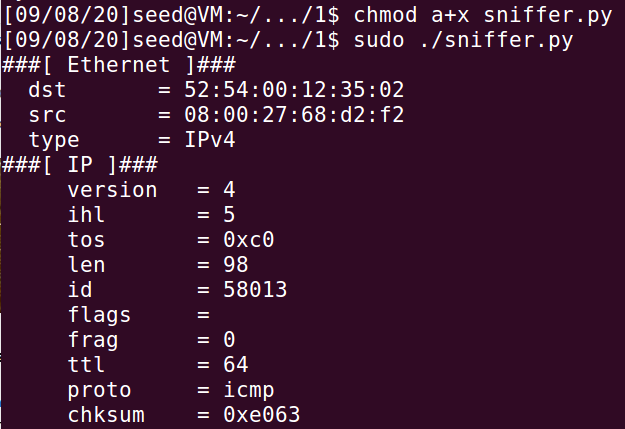
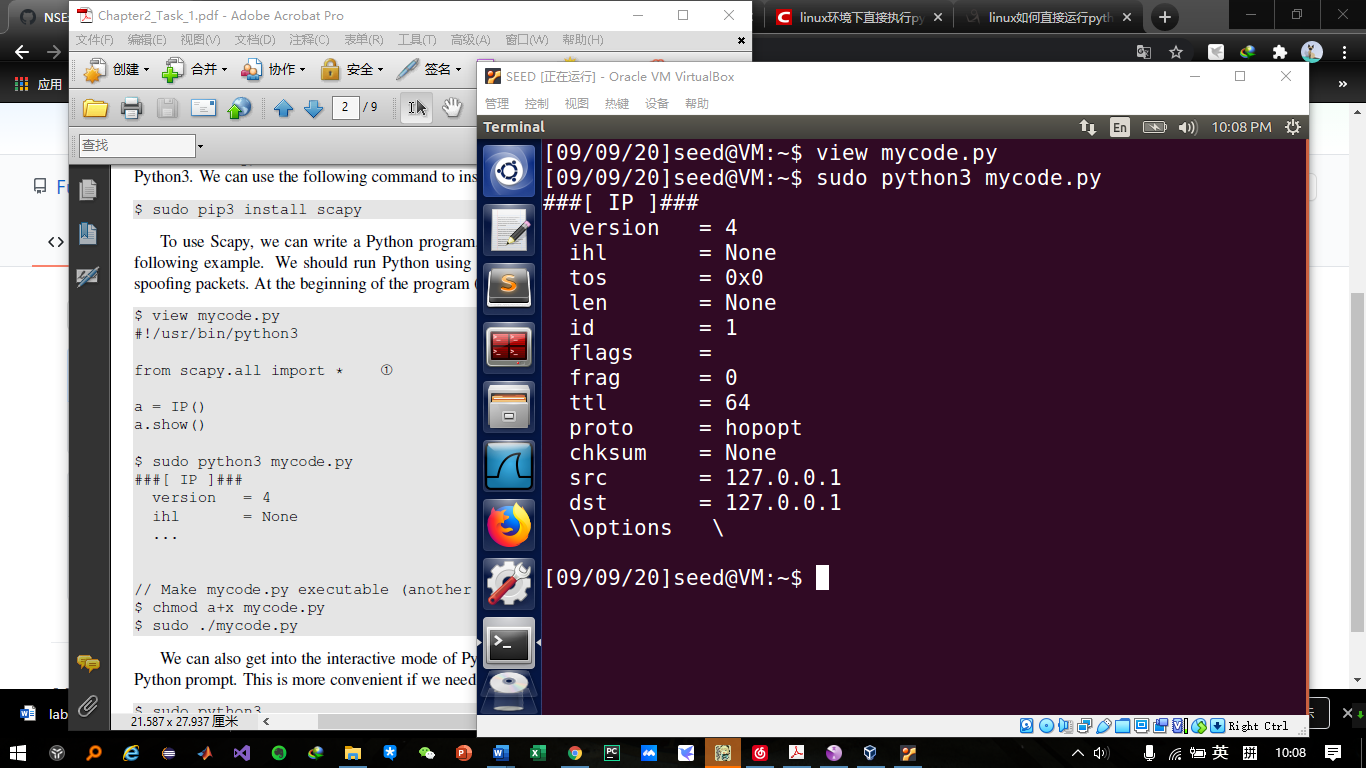
LAB 3

57117131 周煦彭

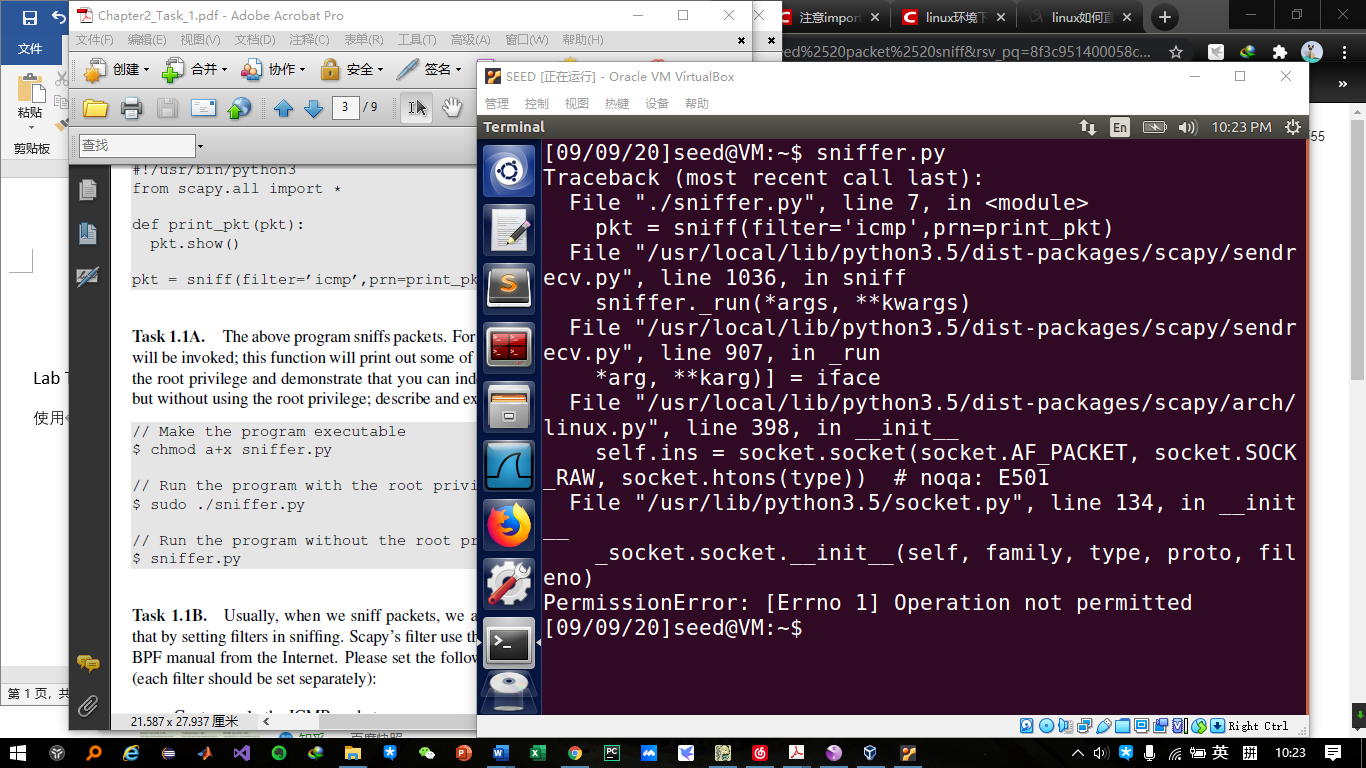
Lab Task Set 1: Using Tools to Sniff and Spoof Packets

1.1A 使用sudo ./sniffer.py命令运行程序，由于使用特权执行命令，故显示结果如下



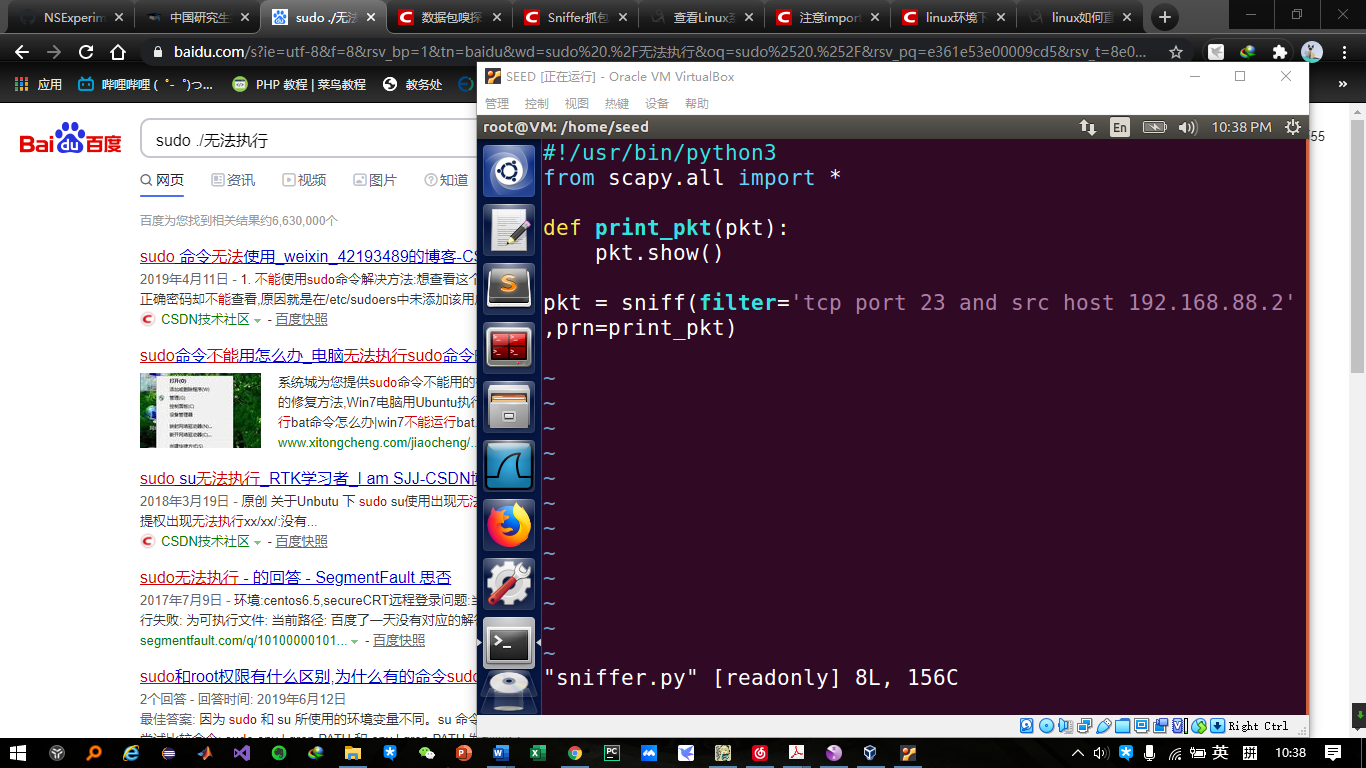


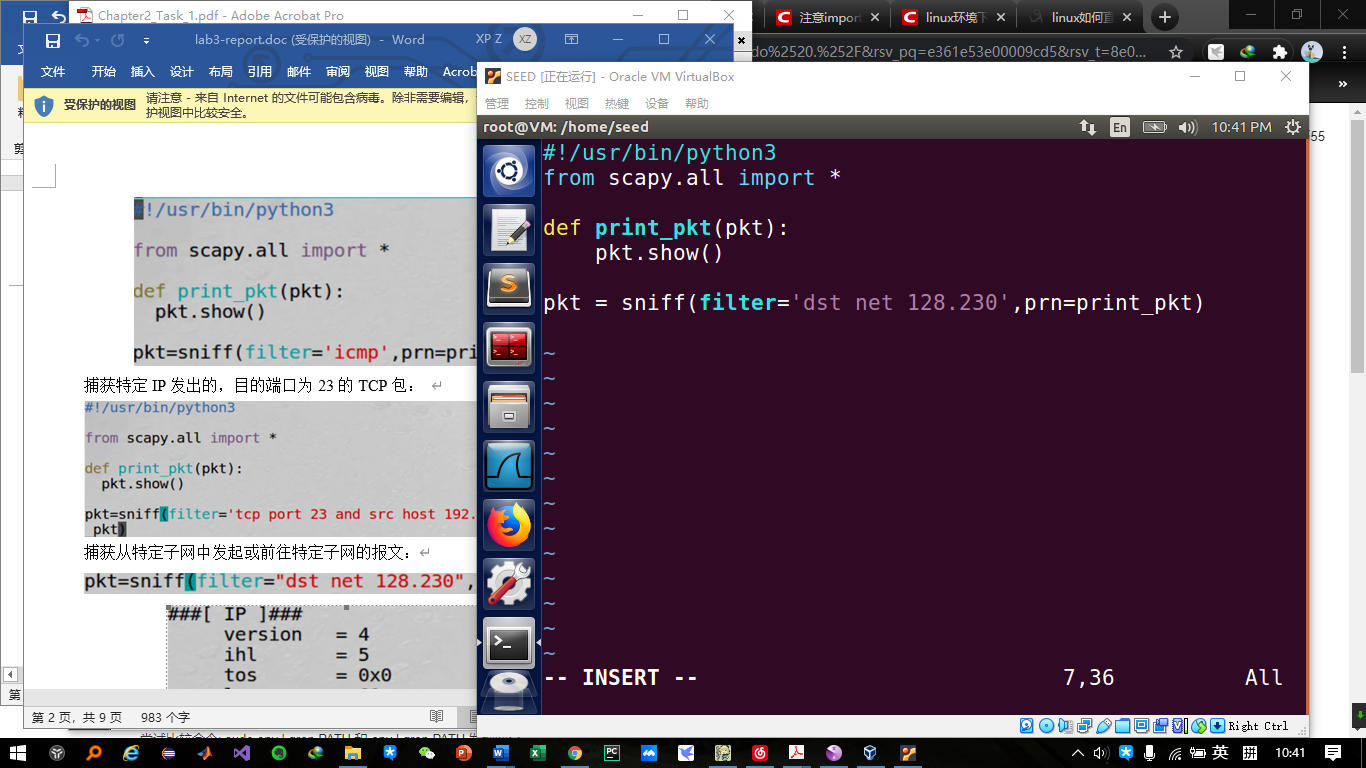
使用去特权执行 sniffer.py,执行失败



1.1B

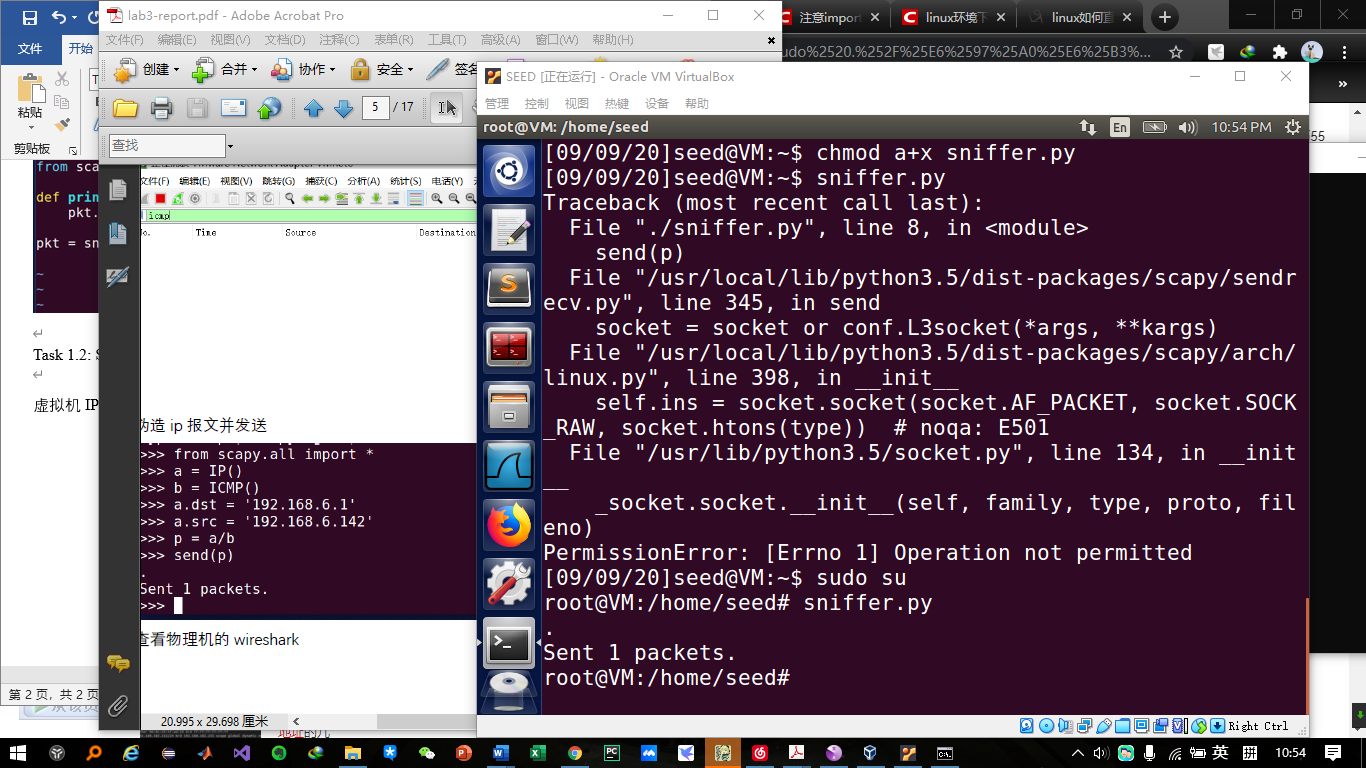
1）ICMP抓取过程与上述一致。

2）抓取TCP报文。修改程序如下：

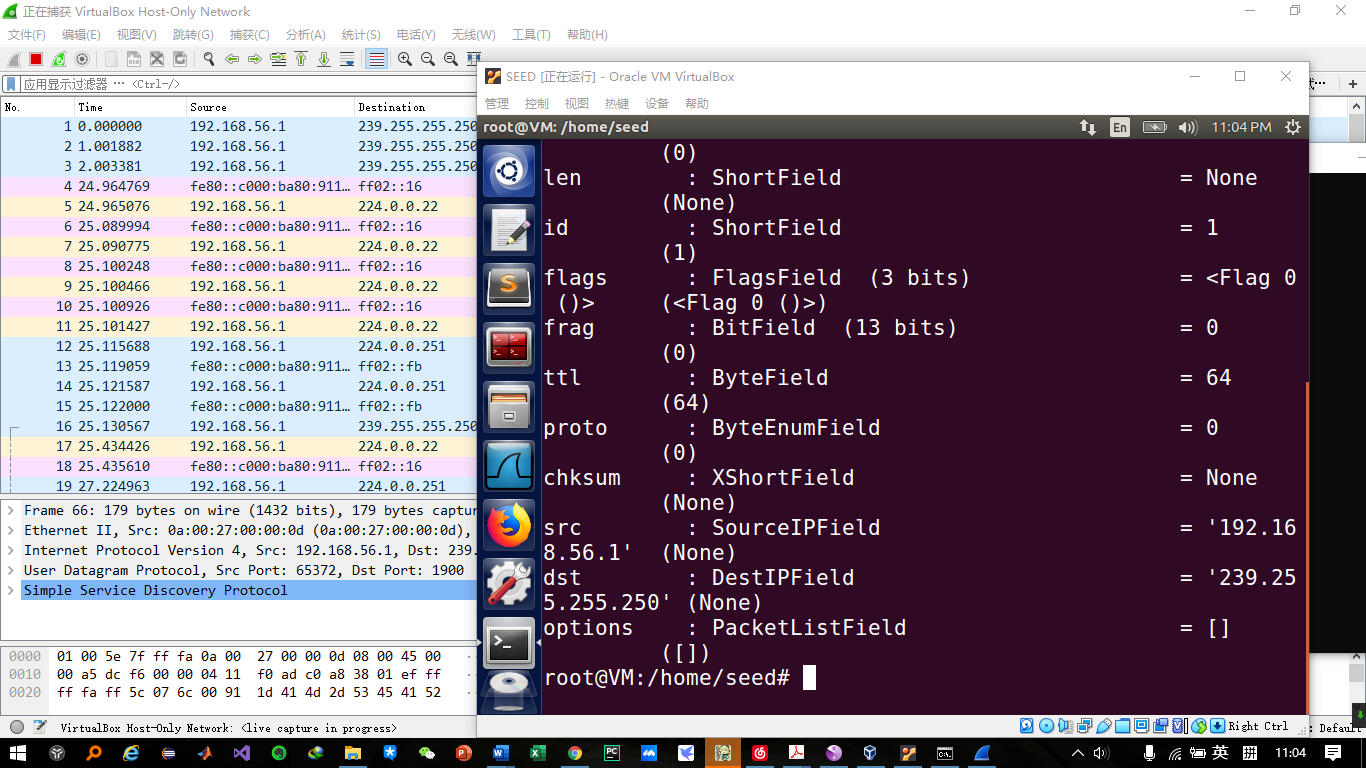
3）捕获特定子网的报文

Task 1.2: Spoofing ICMP Packets

虚拟机IP地址192.168.56.1 主机地址10.203.202.167,发送报文实现如下：

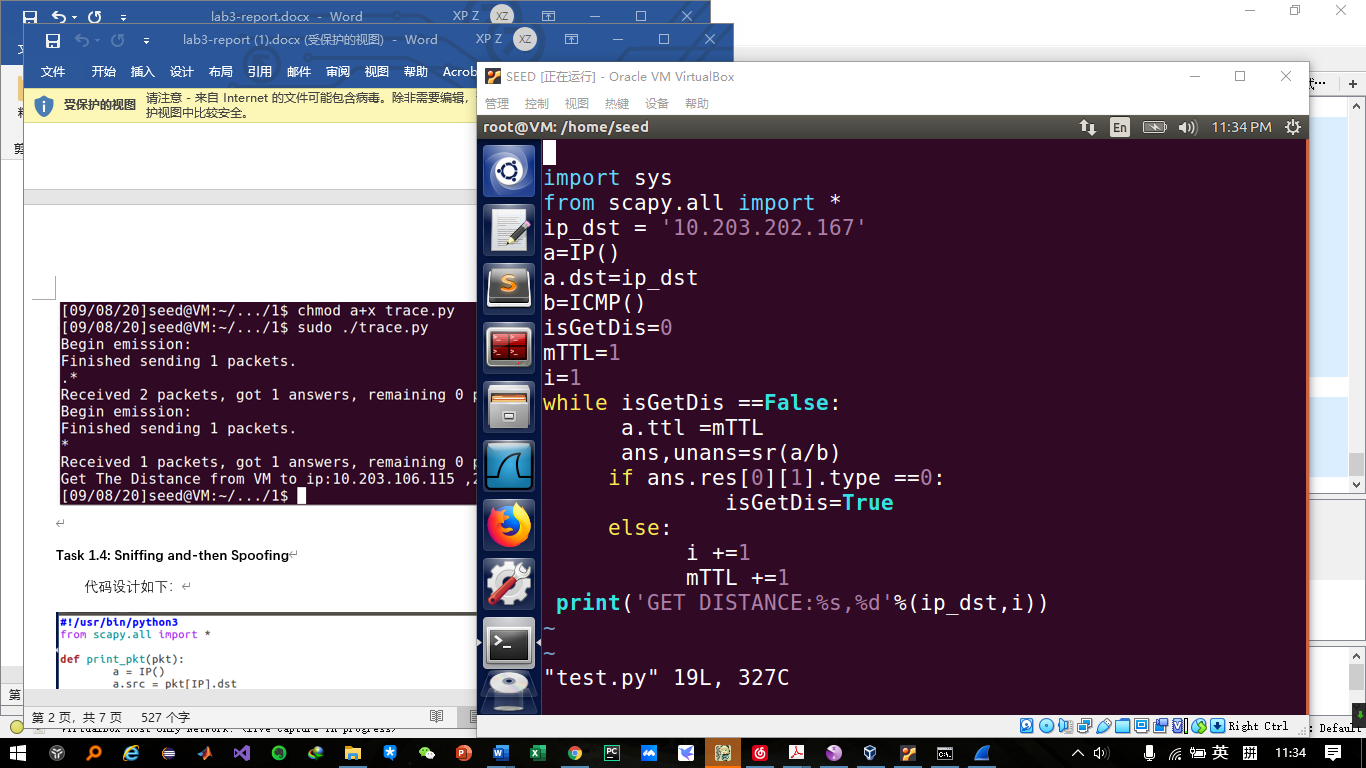


Ls(a)显示如下



所以发送成功。

Task 1.3: Traceroute

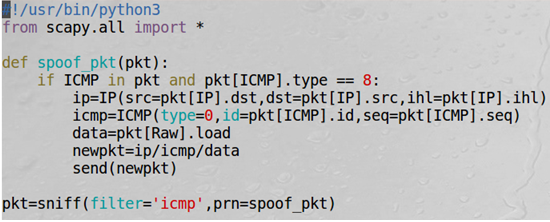
使用scapy得出虚拟机与主机间的路由距离

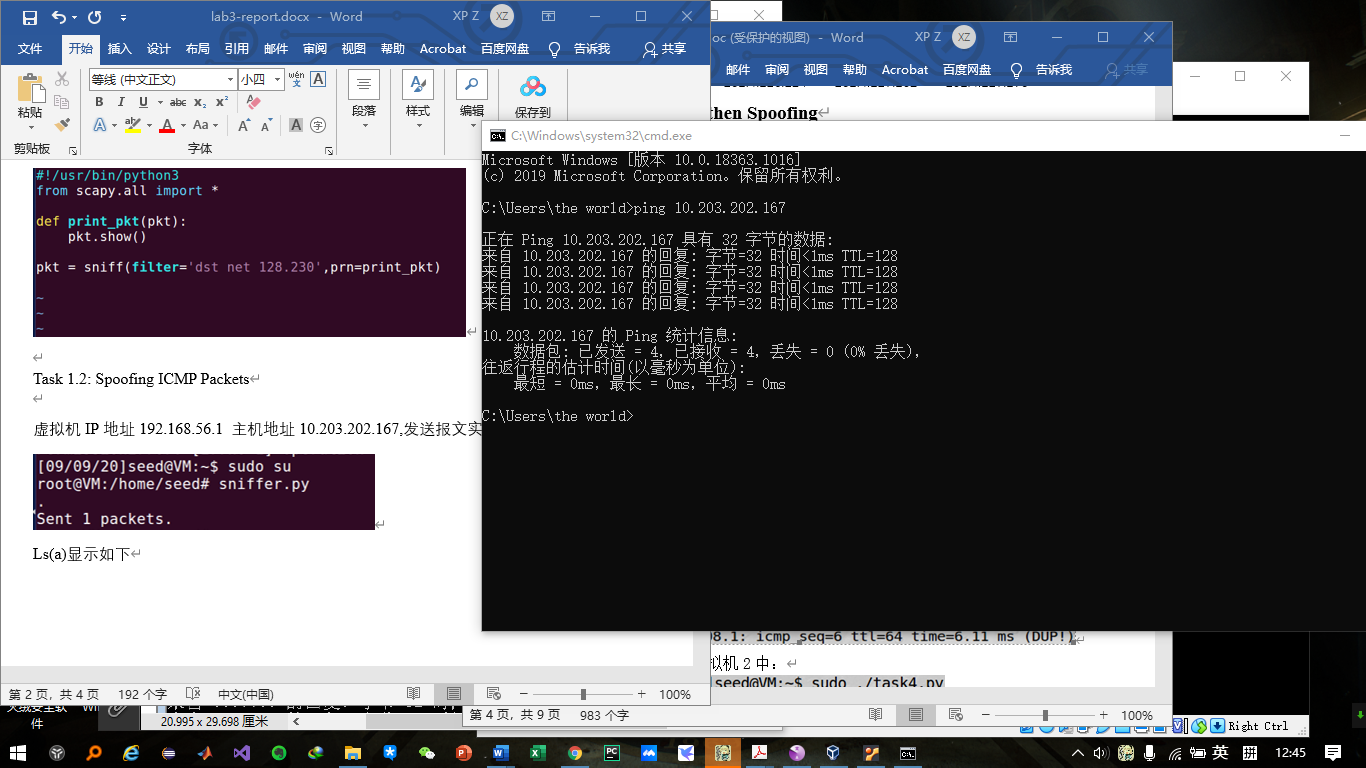
运行结果如下，获得距离为2



Task 1.4: Sniffing and-then Spoofing

设计如下程序



在虚拟机上ping到物理主机

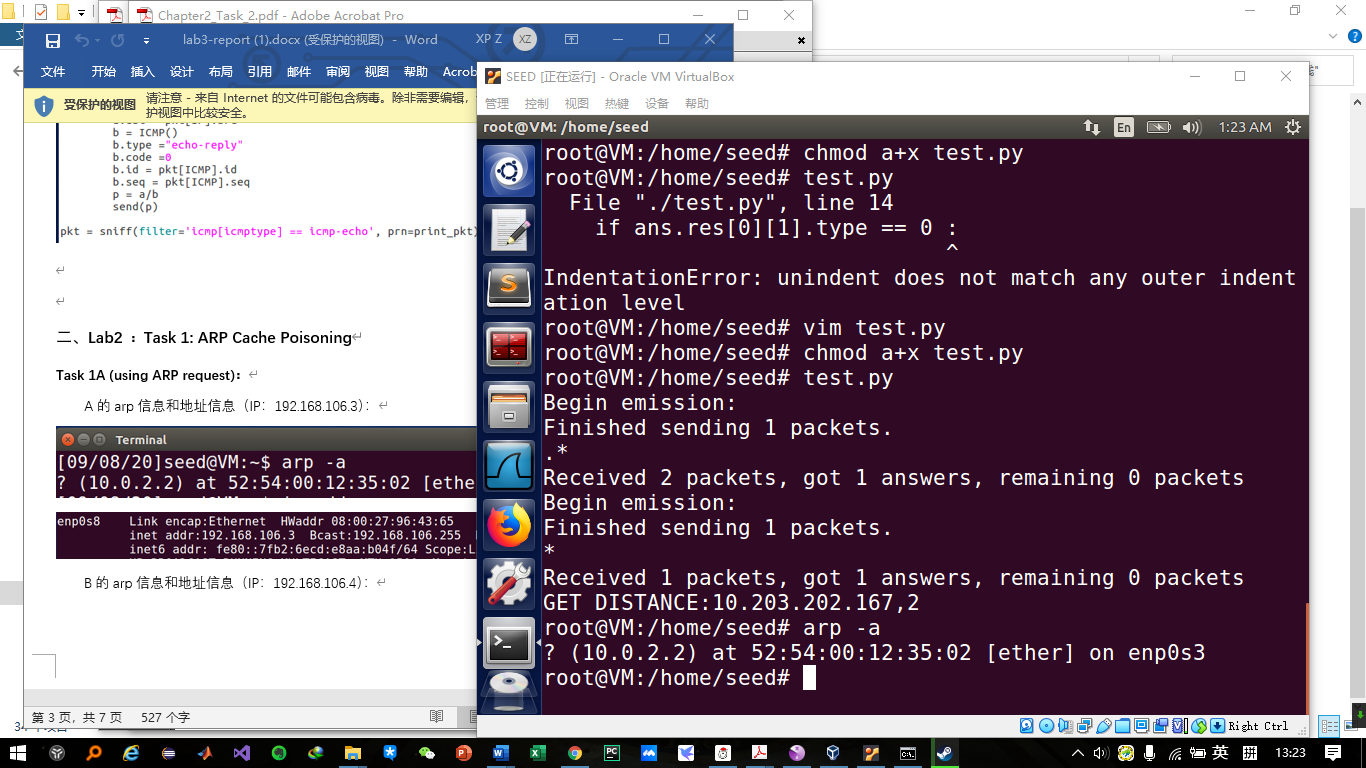
成功ping通

ARP Cache Poisoning Attack Lab

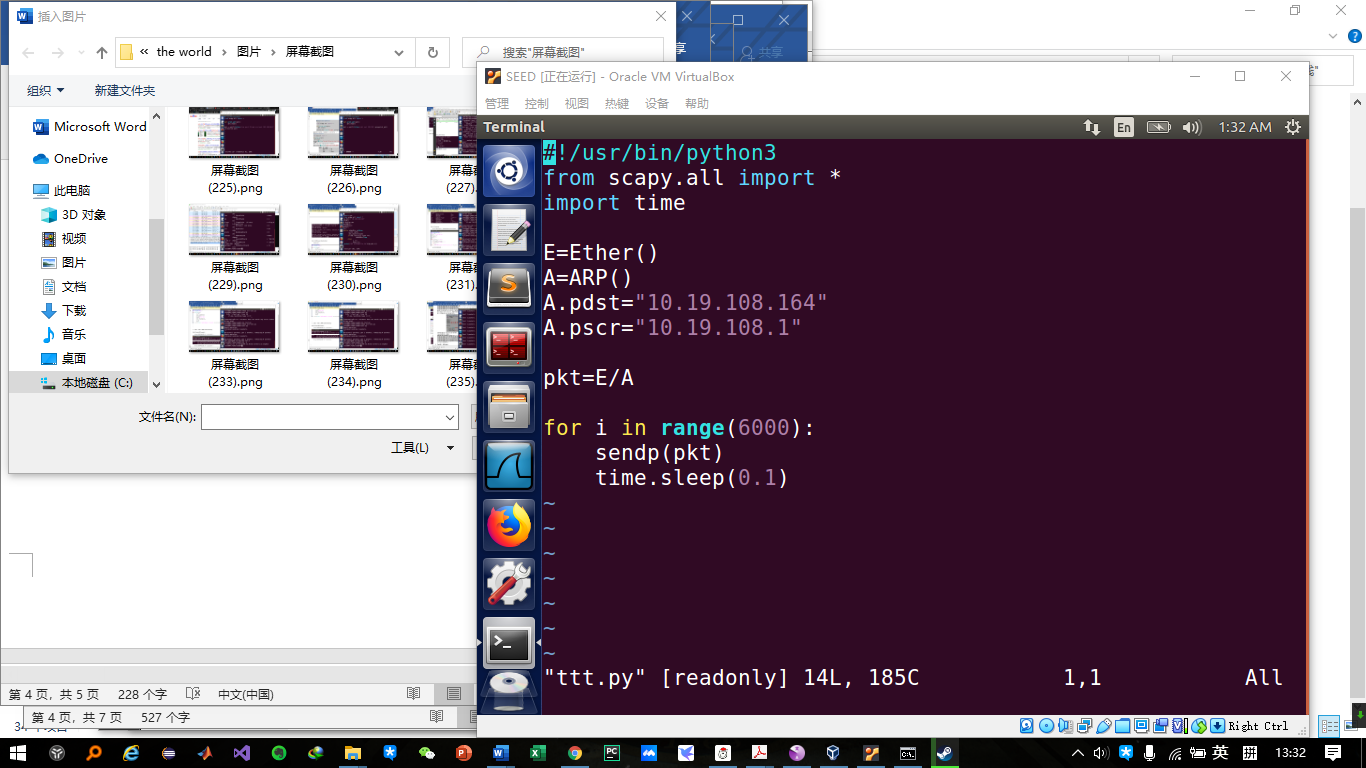
Task 1: ARP Cache Poisoning

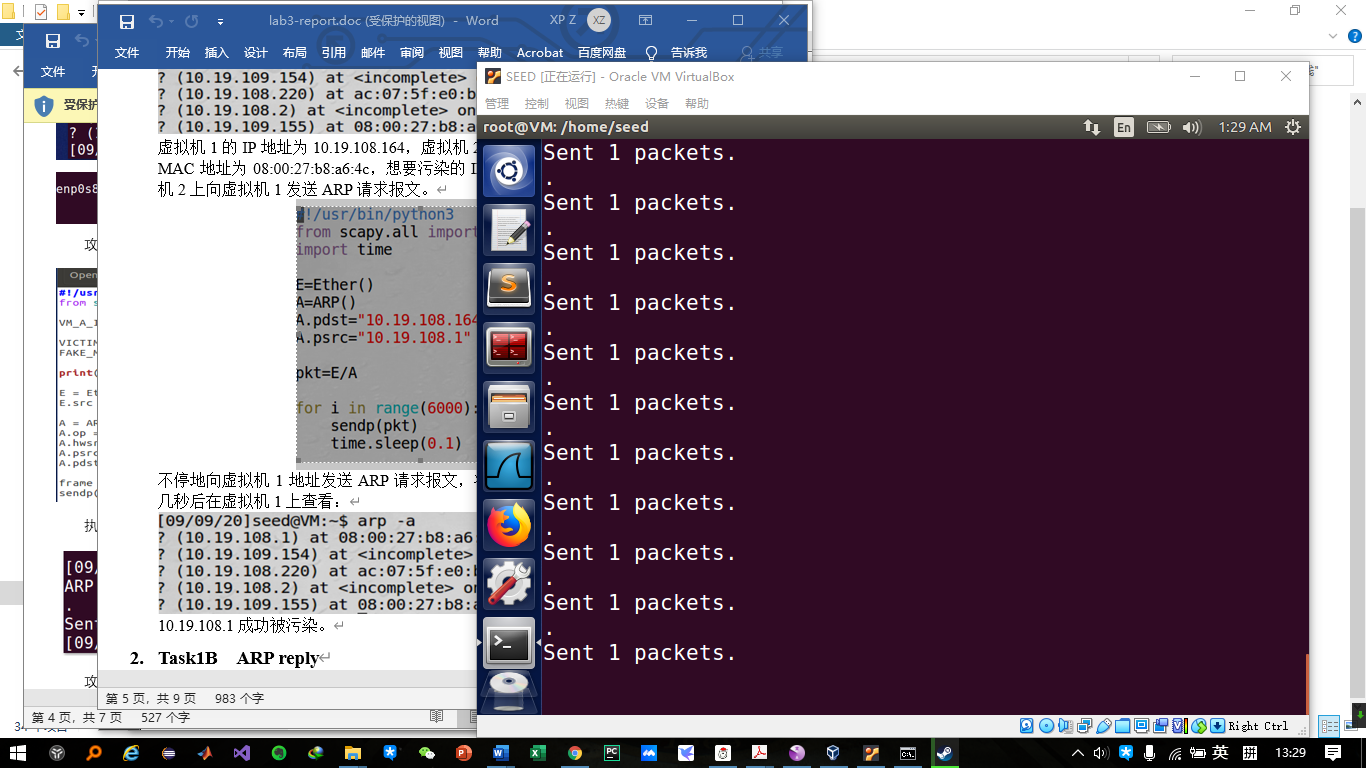
Task 1A (using ARP request).

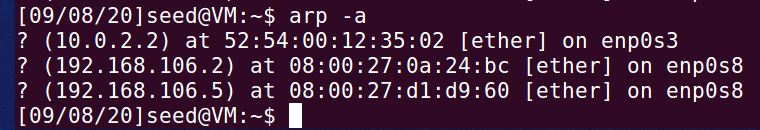
先用arp -a查看地址信息



再执行攻击代码

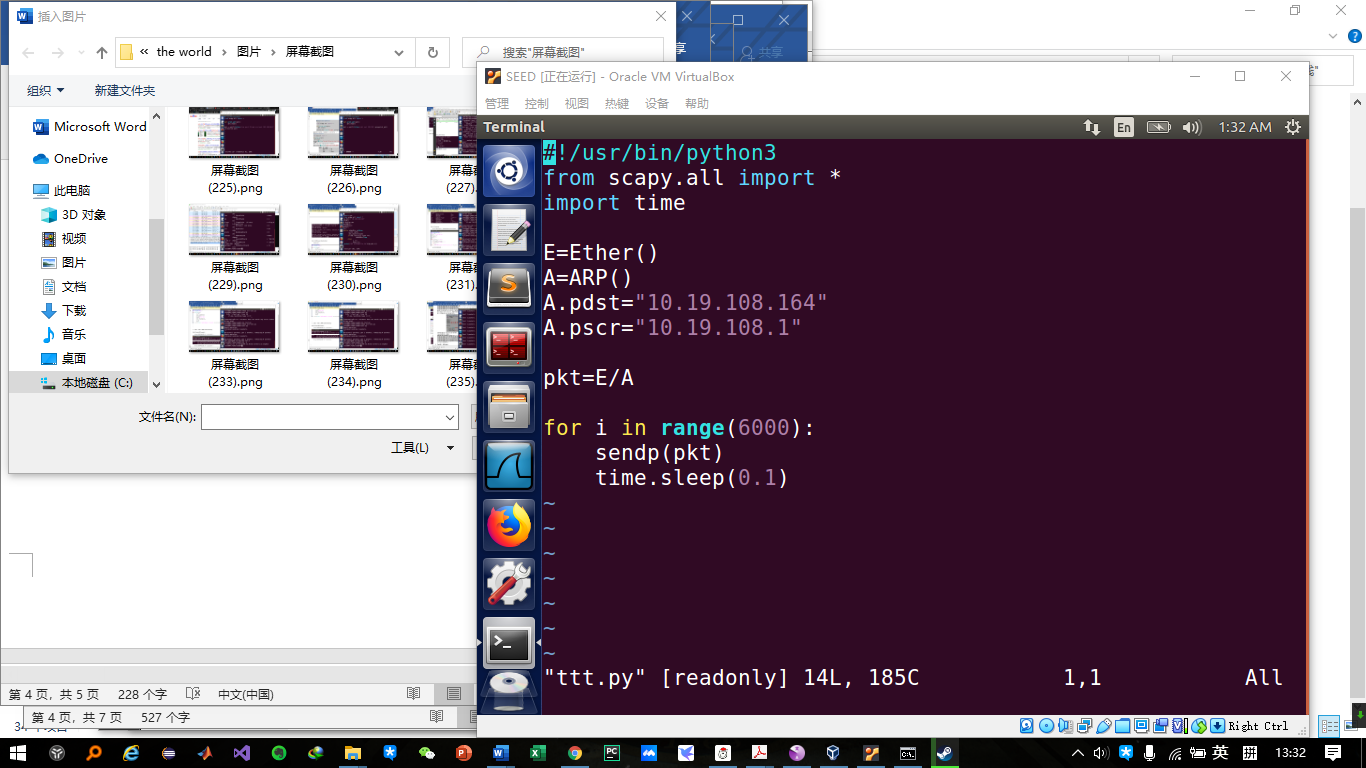




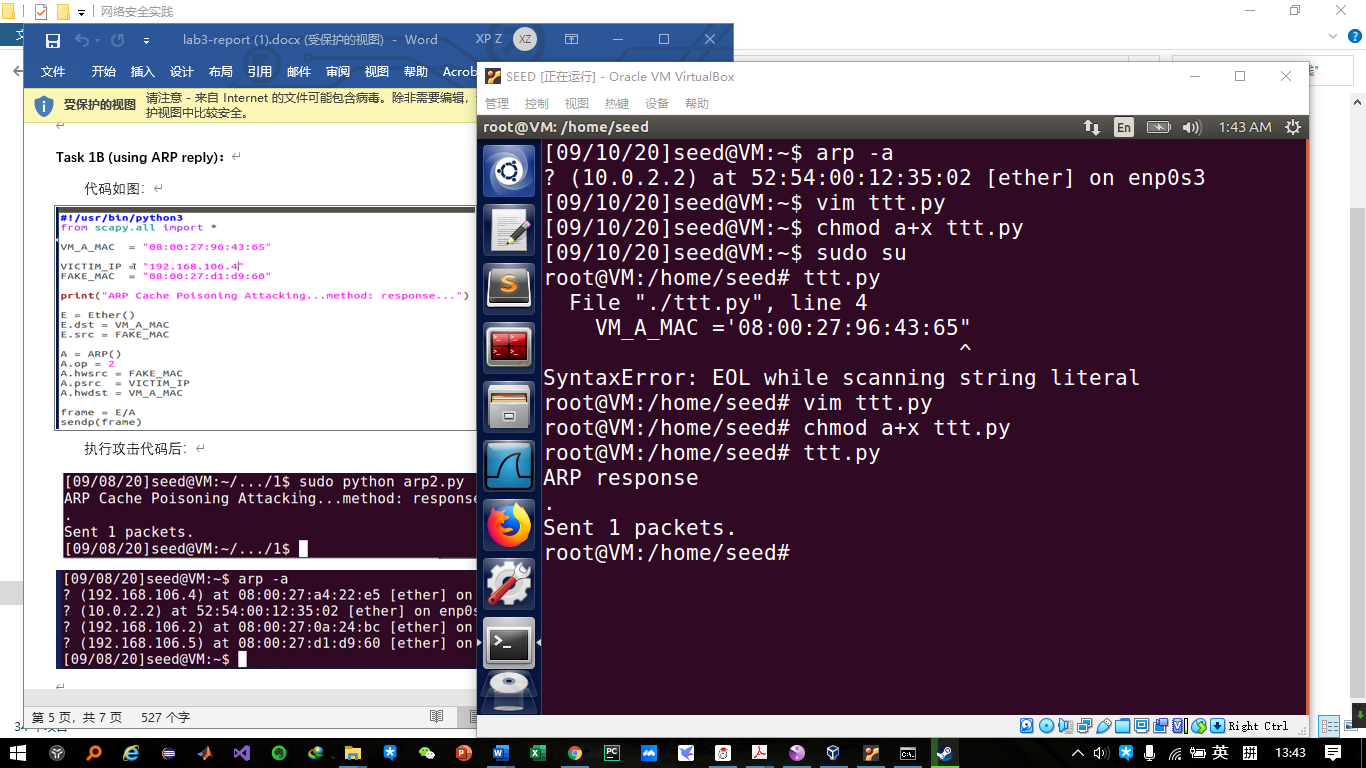
执行结果不停发送报文，结果如下：

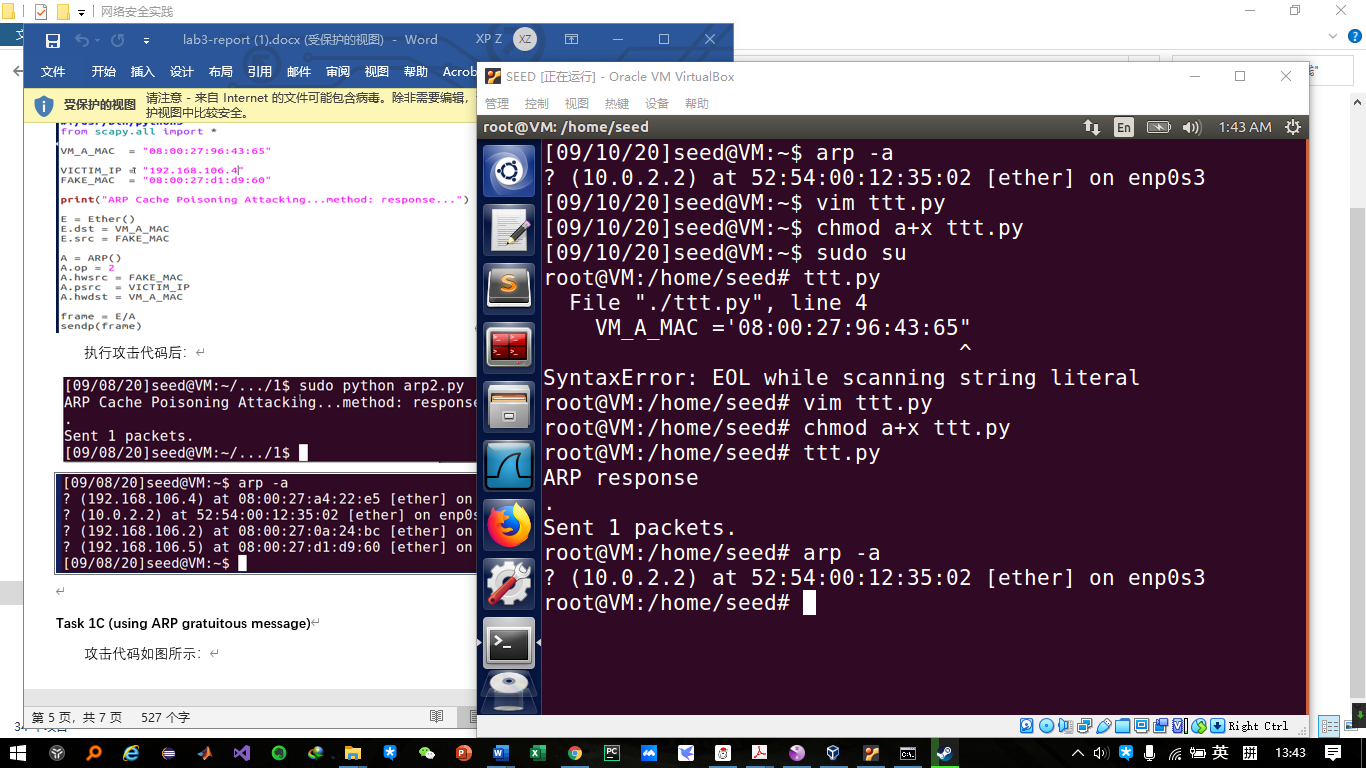
Task 1B (using ARP reply).

执行攻击代码



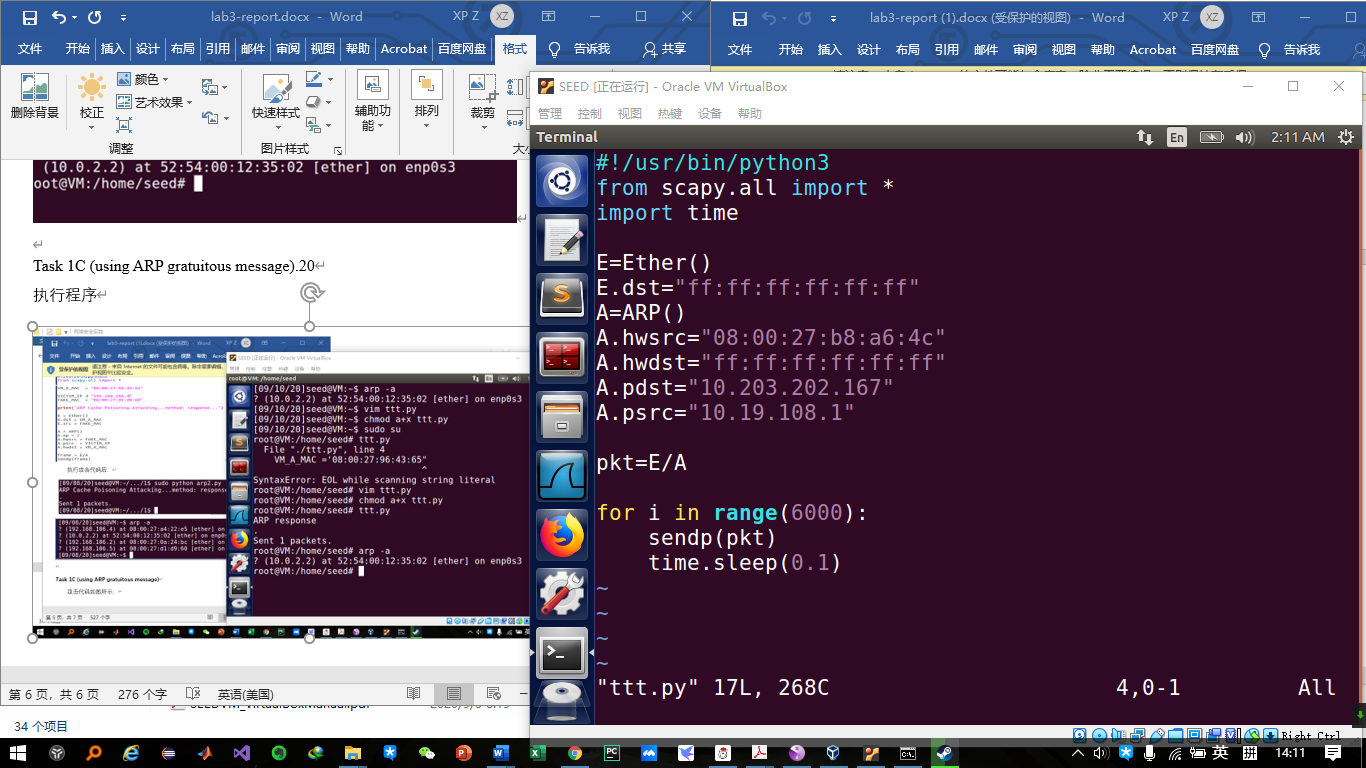
不停向虚拟机发送ARP报文





Task 1C (using ARP gratuitous message).20

执行程序

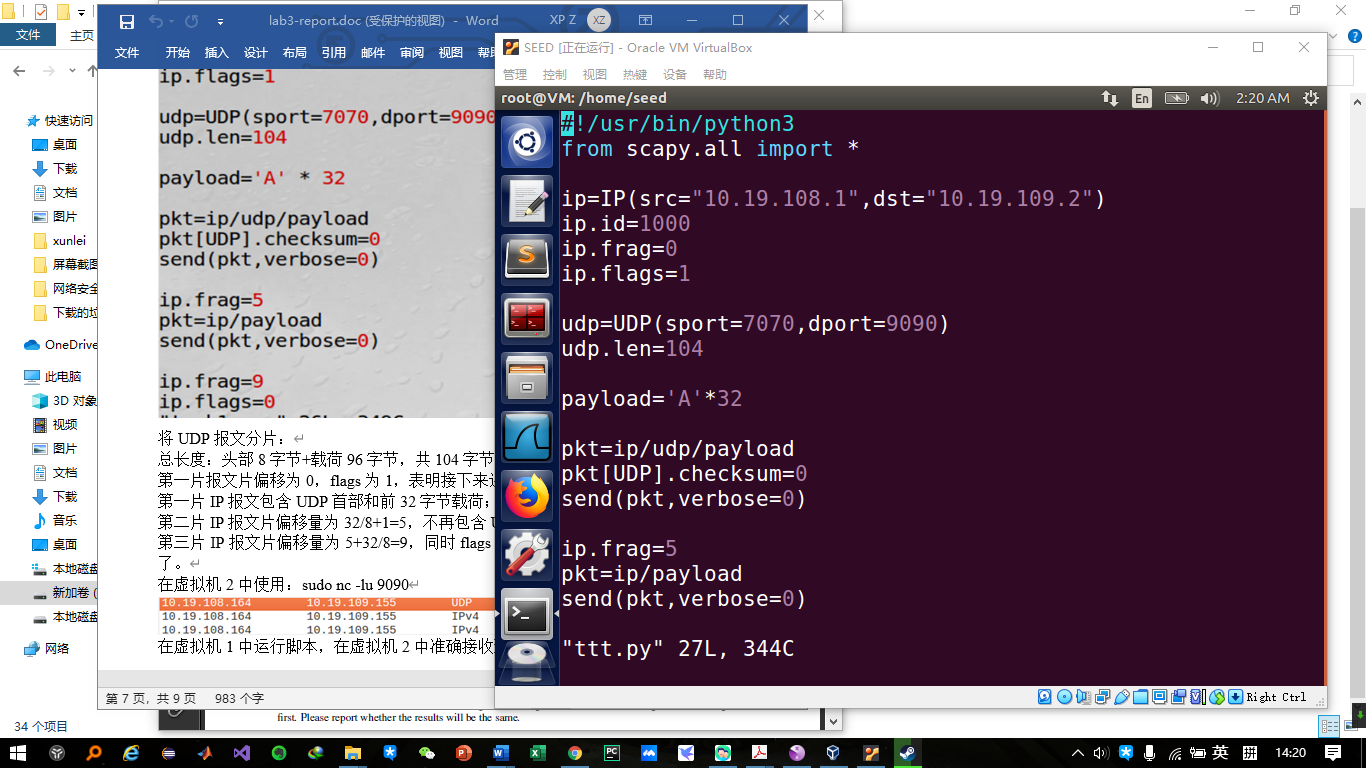


执行最后arp污染失败

IP/ICMP Attacks Lab

Tasks 1: IP Fragmentation

1. a



UDP报文片总长度：头部8字节+载荷96字节，共104字节；

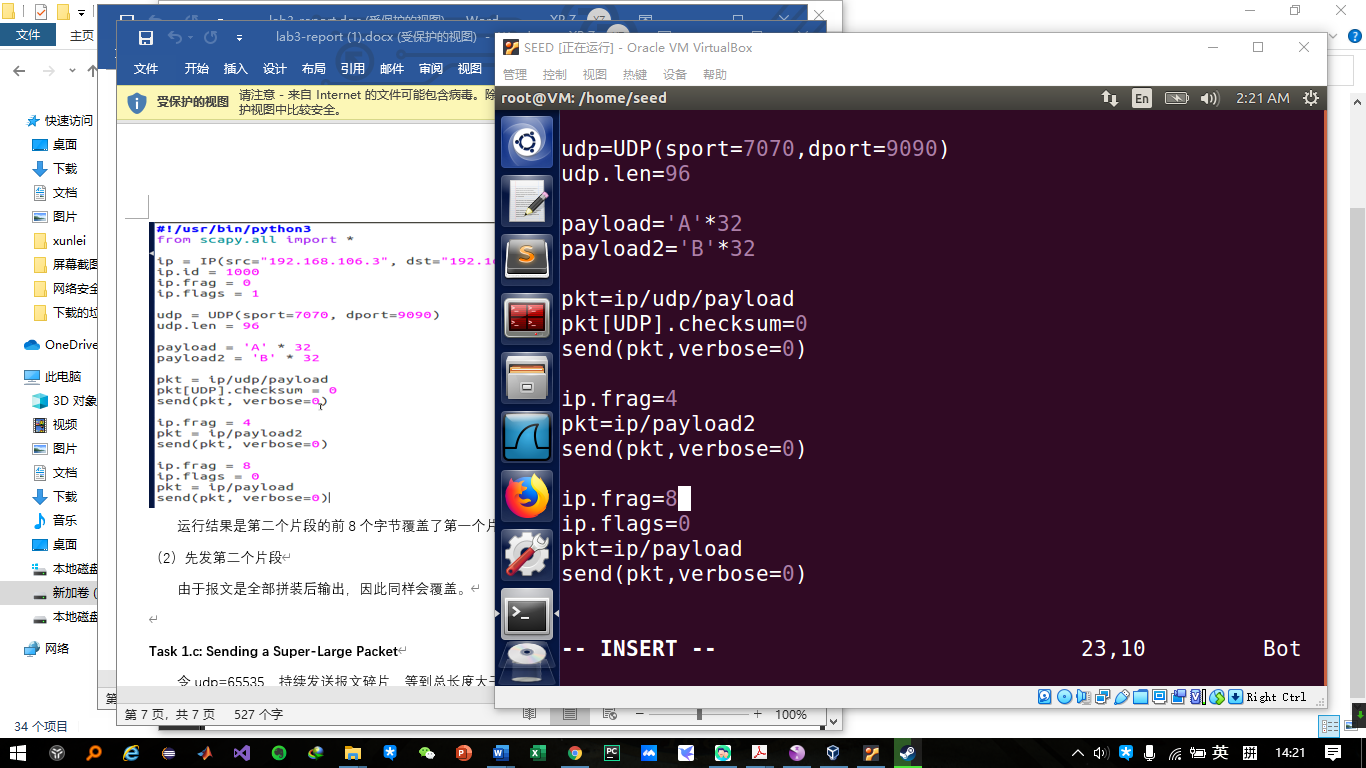
第一片报文片偏移为0，flags为1，表明接下来还有分片；

第一片IP报文包含UDP首部和前32字节载荷；

第二片IP报文片偏移量为32/8+1=5，不再包含UDP首部；

第三片IP报文片偏移量为5+32/8=9，同时flags设为0，表明后面没有分片

1. **b**

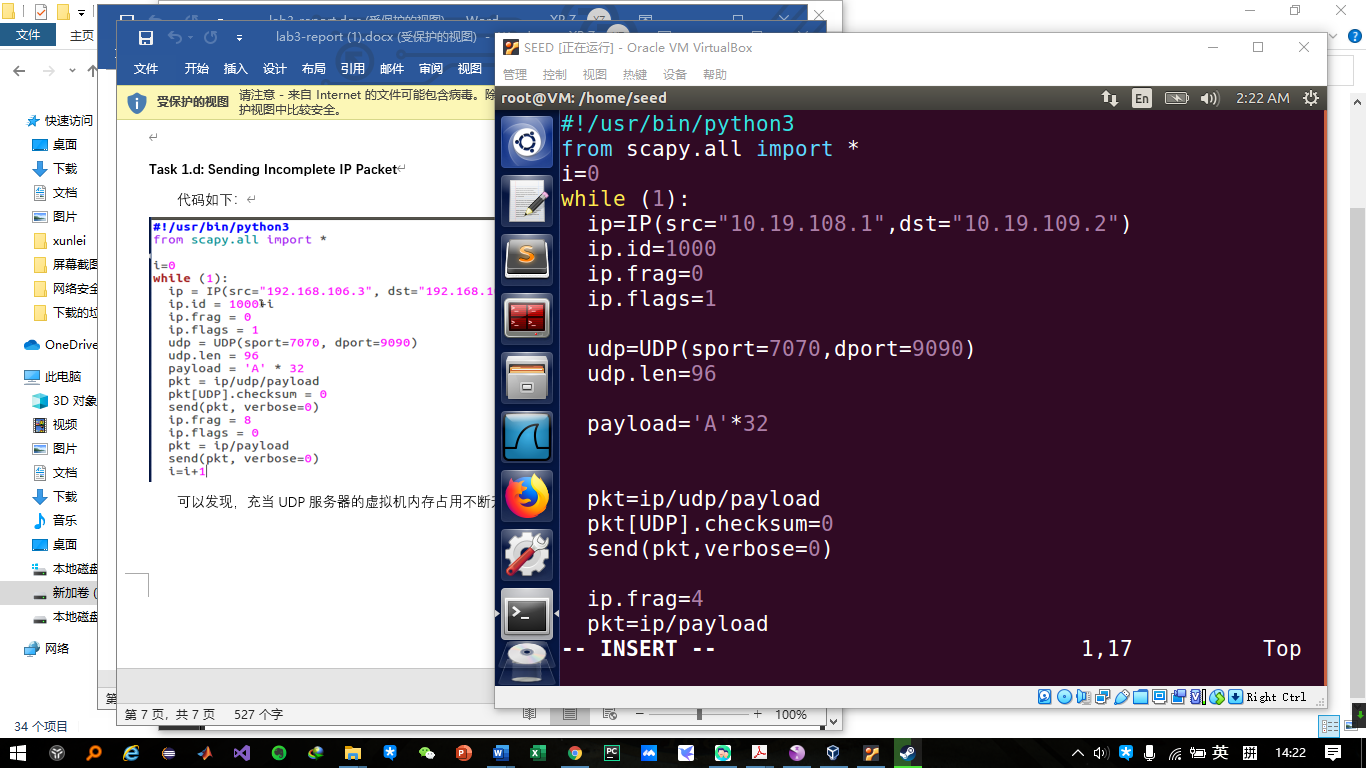


片偏移量设为4，第三片8，UDP长度为96. 交换第二片IP报文与第一片IP报文发出顺序，结果相同，因为内核重组IP报文是在获得全部IP报文之后进行的

**1.c**

将IP头中len字段设为0xFFFF，，当分片总长度超过UDP长度后，设置其flags为0， UDP服务器崩溃了

1. **d**

****

UDP服务器的内存在不断提高