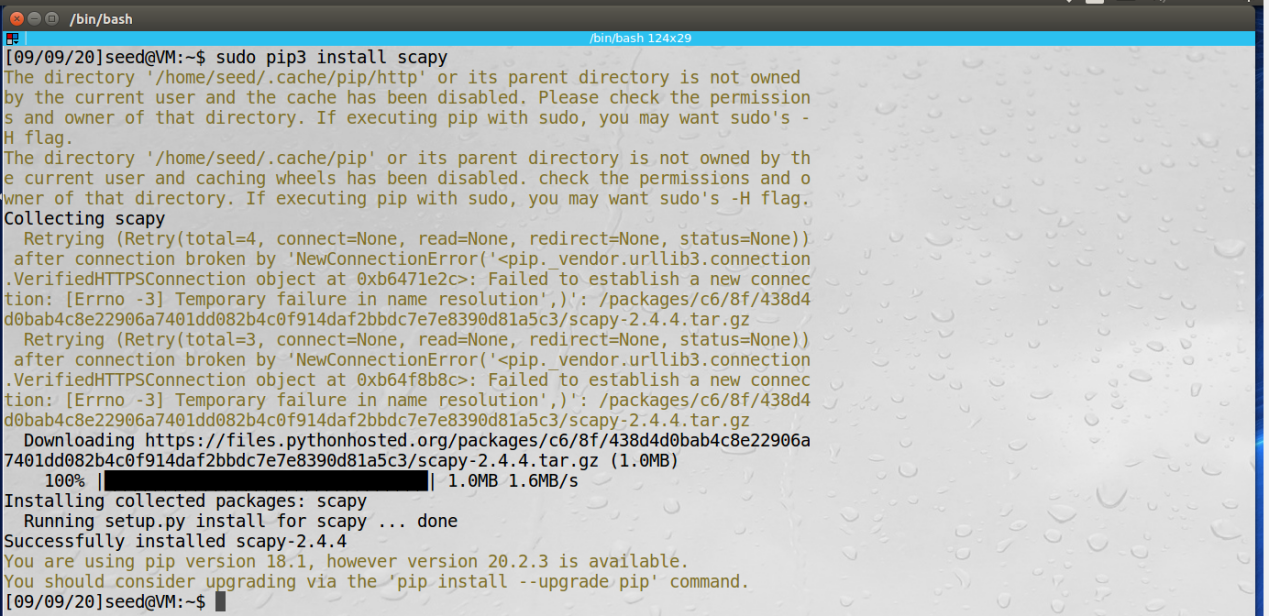
**Lab15-1: Packet Sniffing and Spoofing Lab**

**Task 1.1: Sniffing Packets**

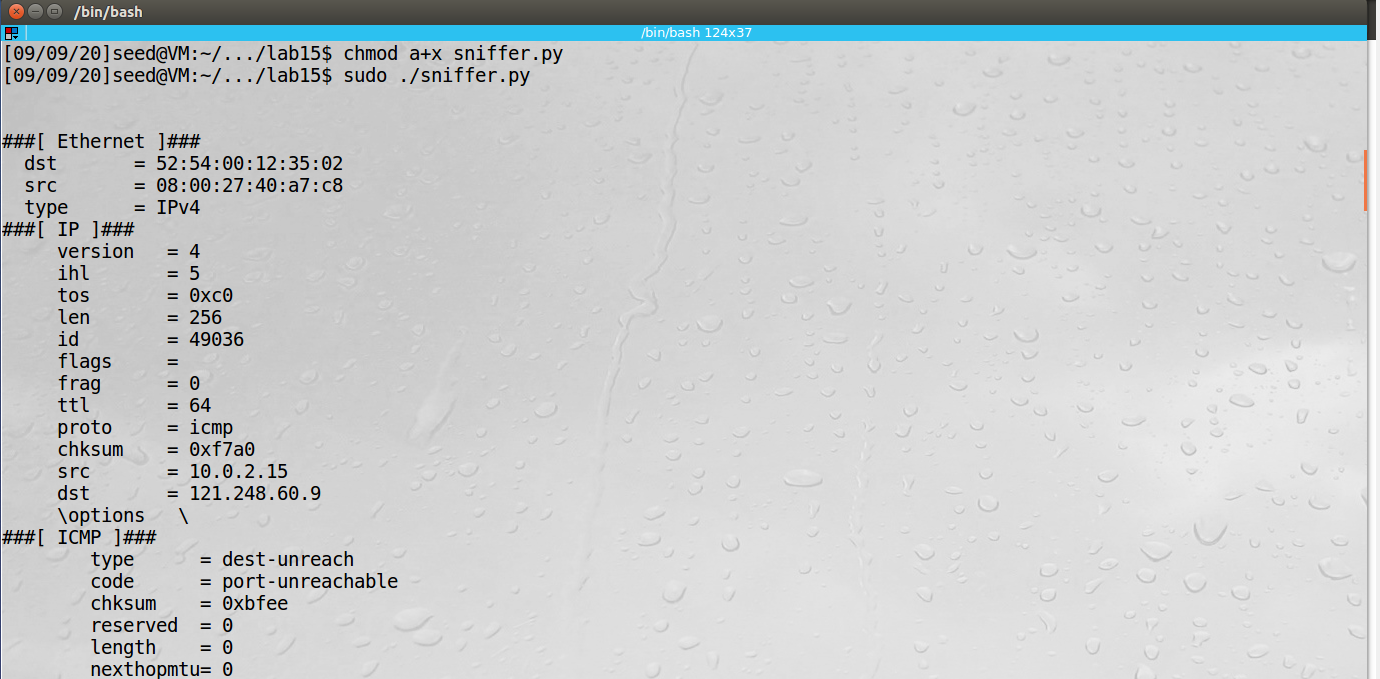
**Task 1.1A.**

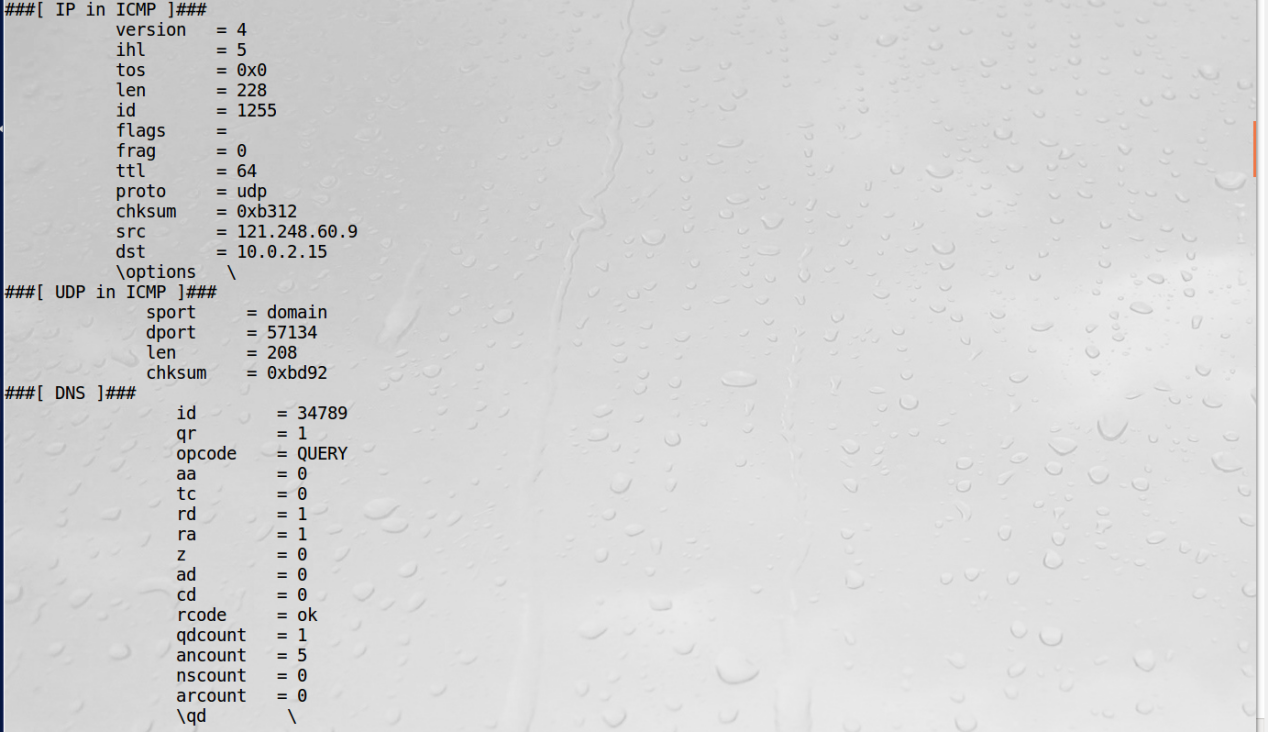
安装scapy并进行配置：





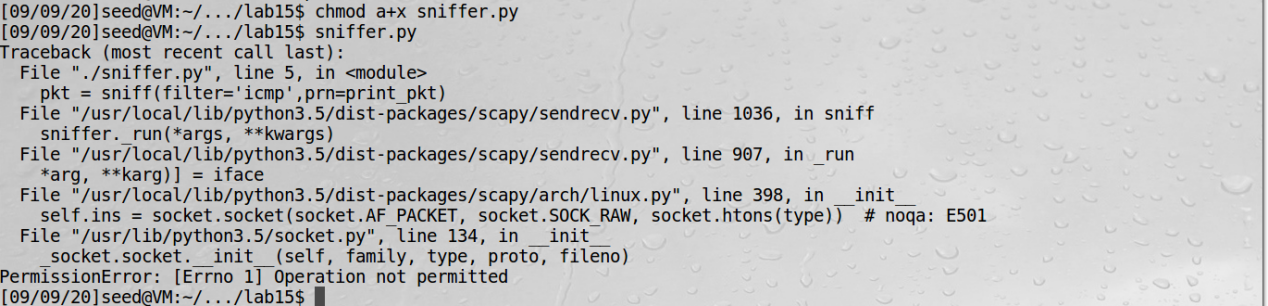
在管理员模式下，可以运行（省略部分结果）：







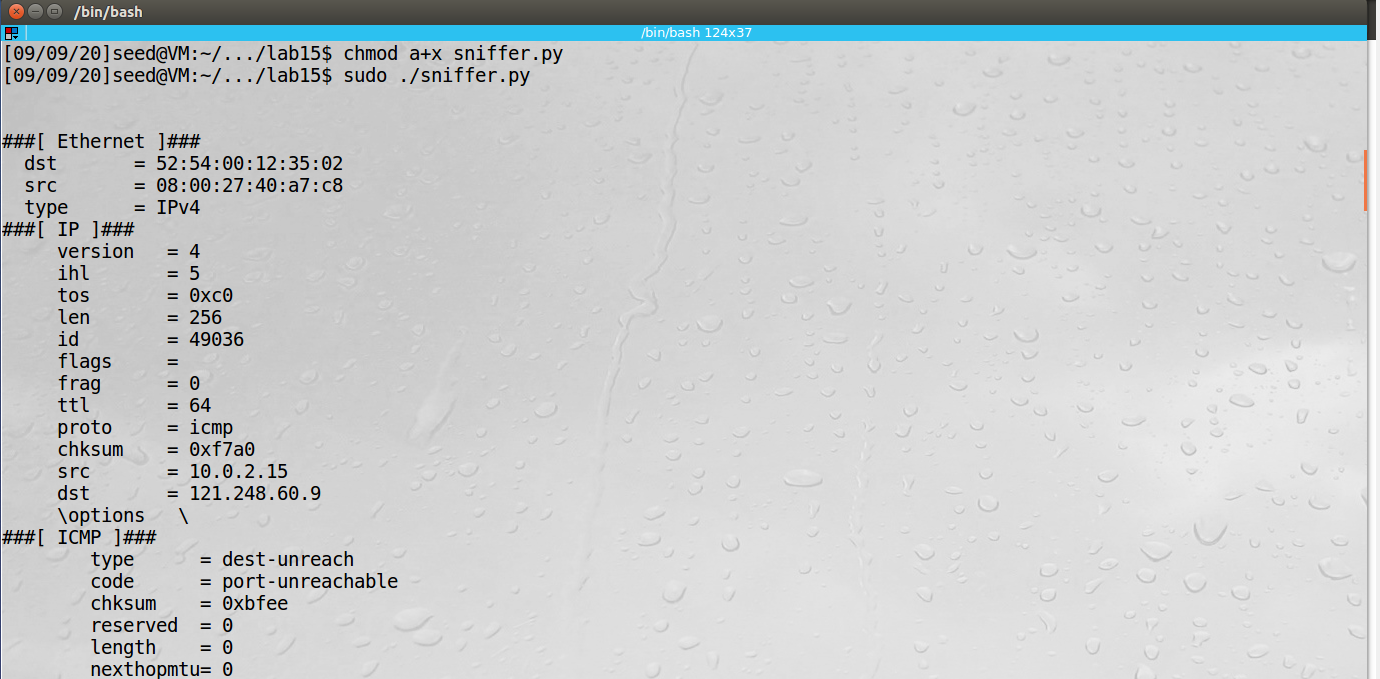
在非管理员模式下，无法运行测试代码：



**Task 1.1B.**

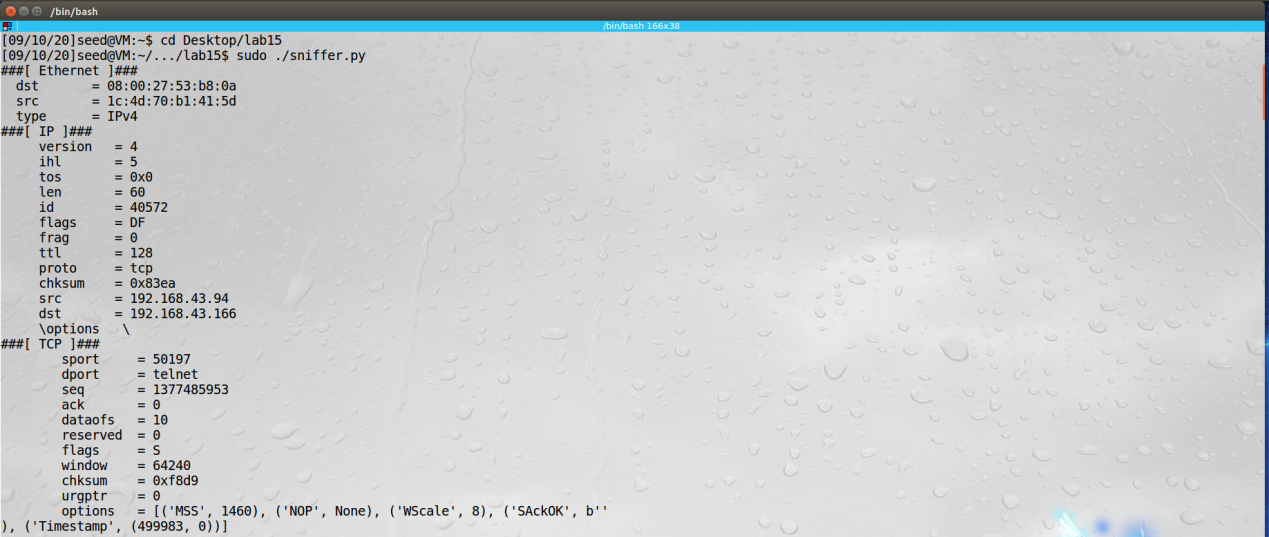
①仅捕获ICMP数据包，执行结果同上:



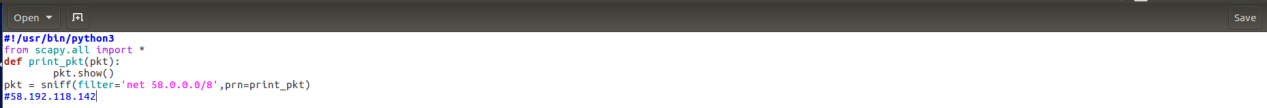


②捕获来自特定IP且目标端口号为23的任何TCP数据包（windows主机上需要开启telnet功能，省略部分结果）：





③捕获数据包来自或进入特定子网（省略部分结果）：

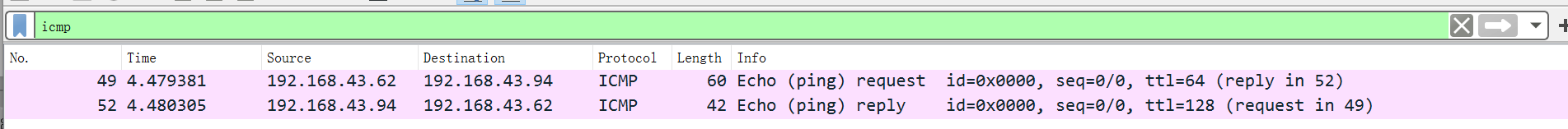




**Task 1.2: Spoofing ICMP Packets**

向主机发送伪造的（来自另一个ip为192.168.43.62的虚拟机）报文，在主机中捕获到了该报文以及主机的应答报文，说明成功伪装：





**Task 1.3: Traceroute**

使用Scapy估算路由器之间的距离，虚拟机和主机之间的路由器数量为1：

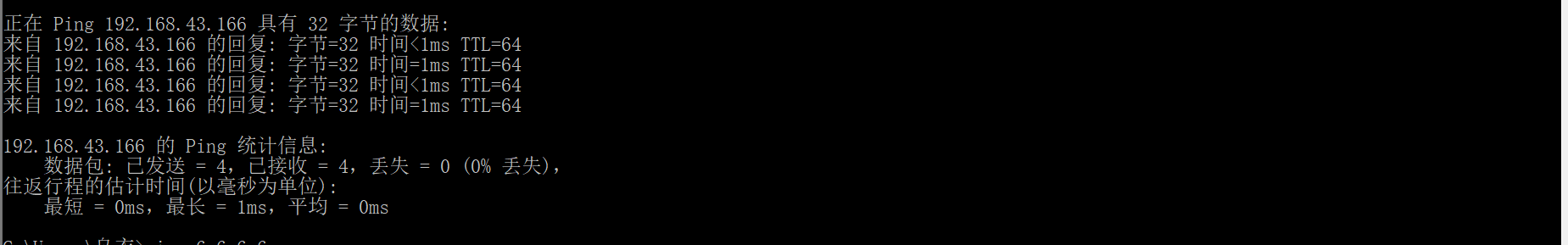




**Task 1.4: Sniffing and-then Spoofing**

攻击者通过伪造回复报文，对被攻击者进行欺骗：



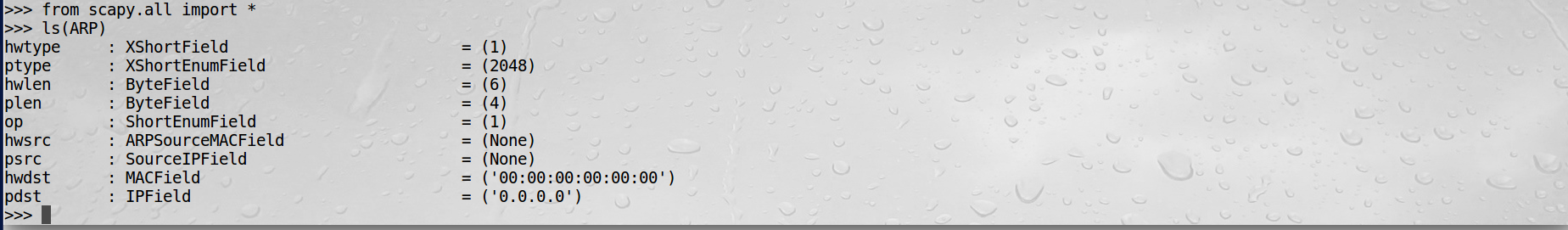


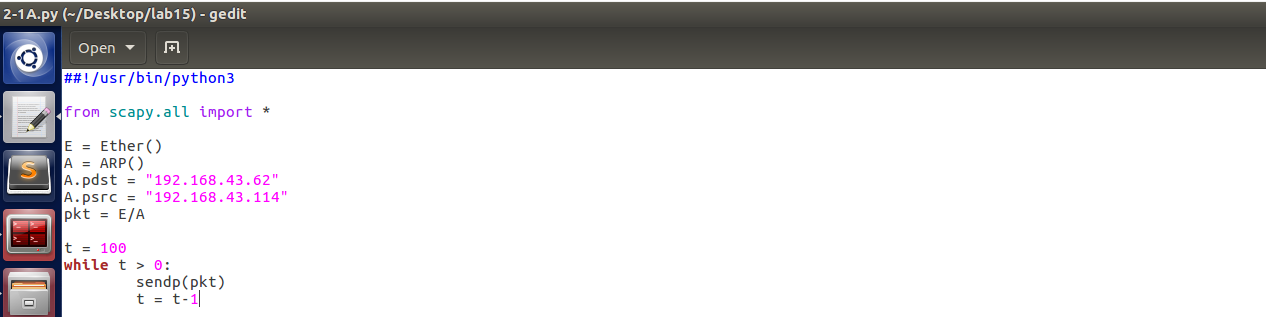


**Lab15-2: ARP Cache Poisoning Attack Lab**

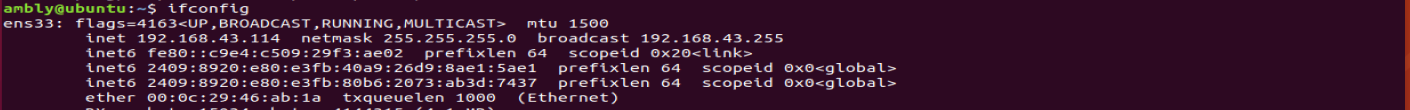
**Task 1A (using ARP request):**

查看ARP对象：攻击者通过发送编造的ARP请求报文，对被攻击者的MAC地址进行更改：

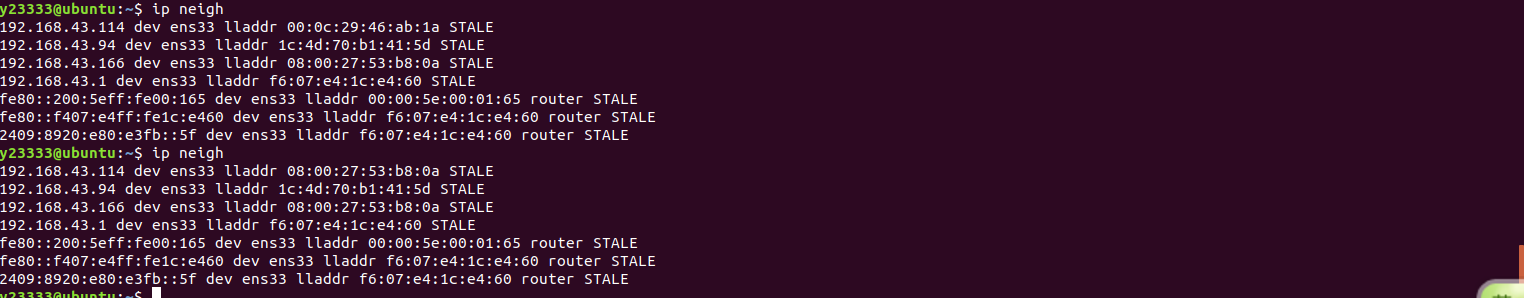






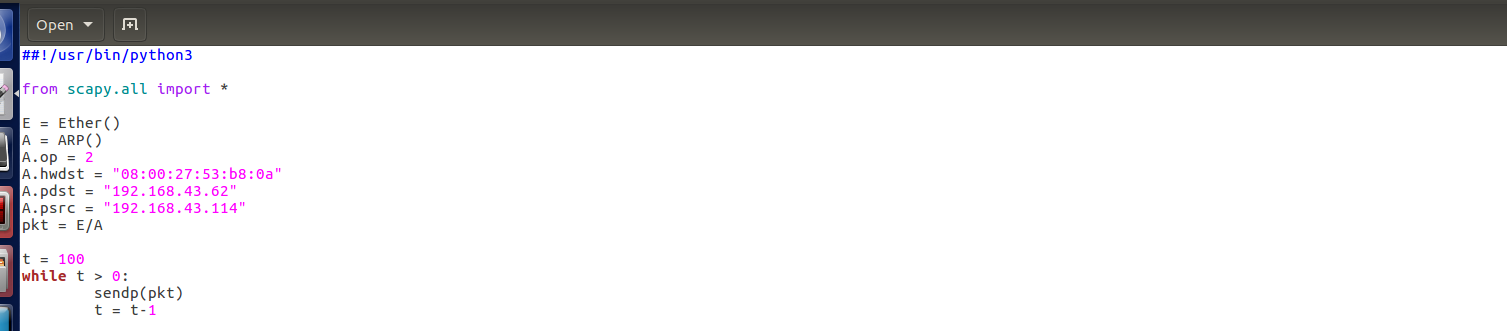


上下对比，更改成功：



**Task 1B (using ARP reply):**

恢复到正确的MAC地址后，攻击者编造ARP回复报文完成上述相同目的：



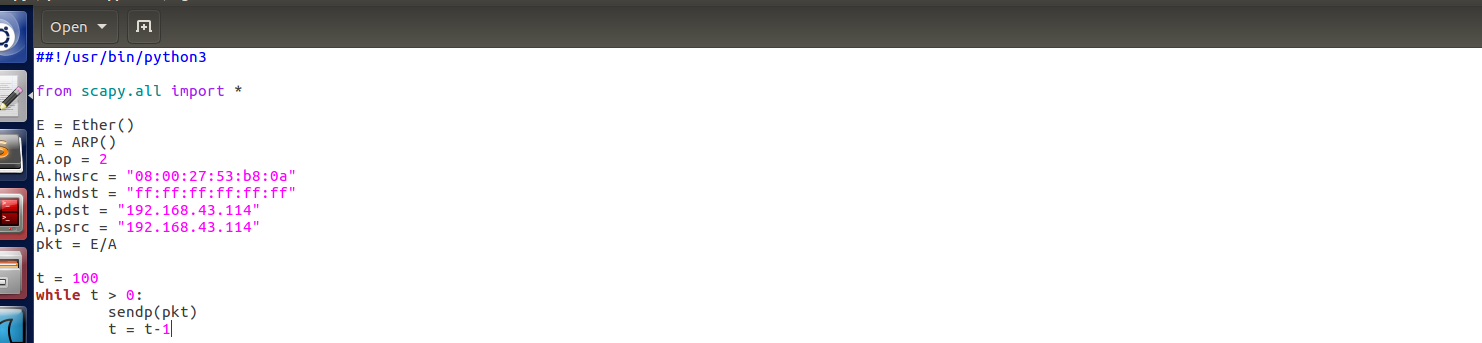
攻击成功：



**Task 1C (using ARP gratuitous message):**

恢复到正确的MAC地址后，攻击者编造循环的ARP gratuitous报文完成上述相同目的：





攻击成功：



**Lab15-3: IP/ICMP Attacks Lab**

**Task 1.a: Conducting IP Fragmentation**

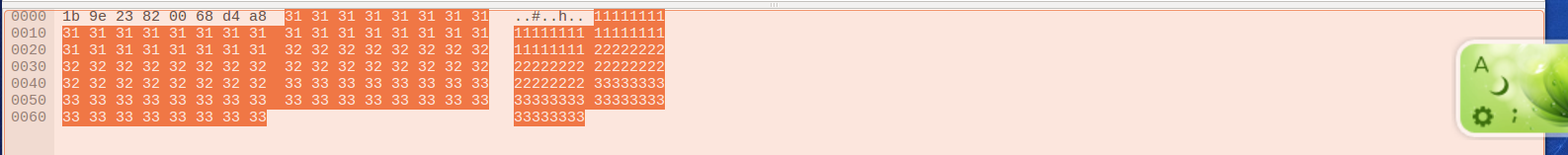
依据要求，将一个udp数据包划分为三个片段，分别构建这三个片段并将其发送给udp服务器：



启动udp服务器，并用wireshark捕获udp报文：



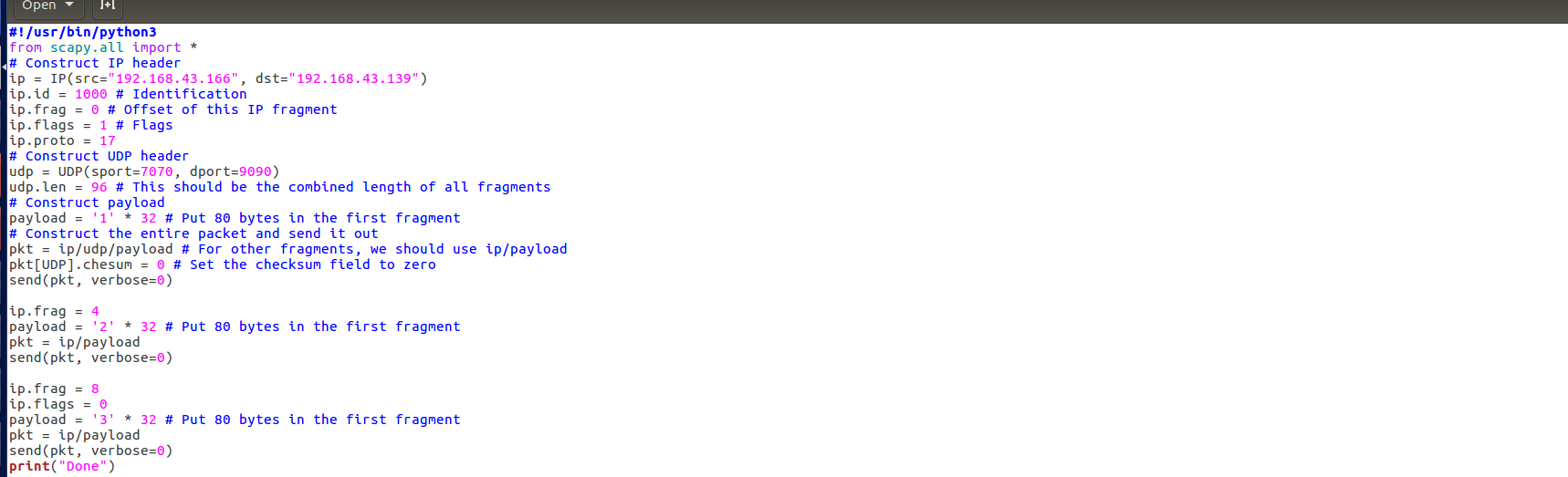




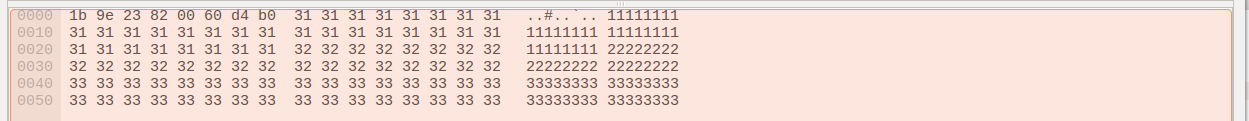
发送的三个片段合成了一个报文，并被服务器成功接收。

**Task 1.b: IP Fragments with Overlapping Contents**

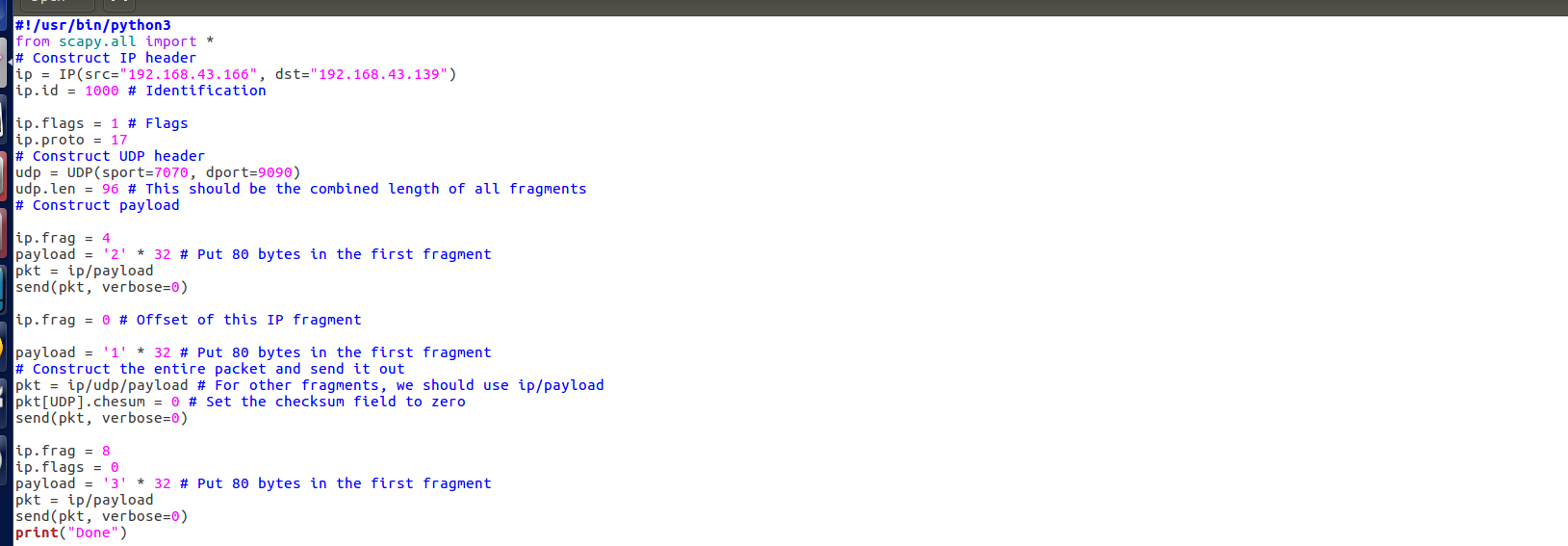
①对比上述实验1.a，将k设为8，即将第二个片段的前八的字符设置成与第一个片段重合：

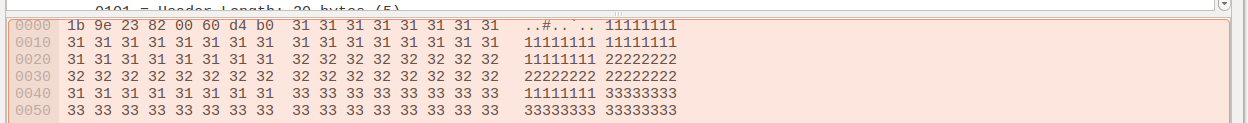


发现第二个片段的前八个字符丢失：



改为先发第一个报文，再发第一个保文，发现结果不变：

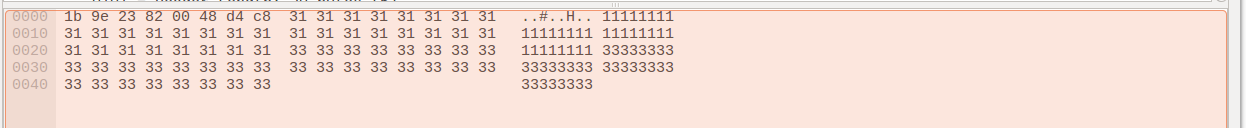




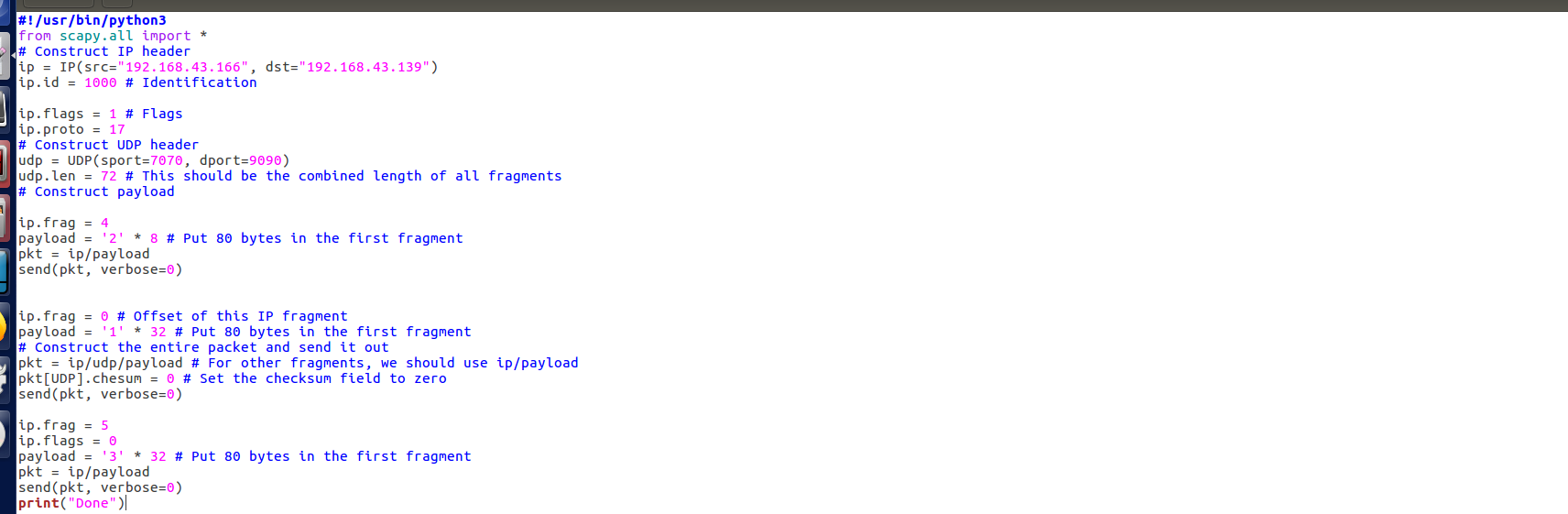
②第二个片段完全包含在第一个片段中（第二个片段的大小小于第一个片段）：

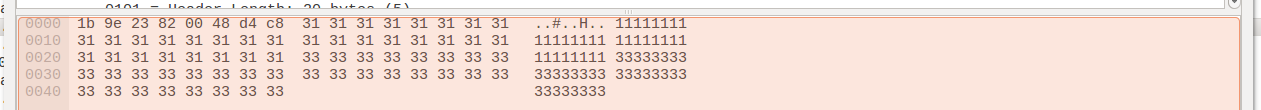


第二个片段完全被遮盖：



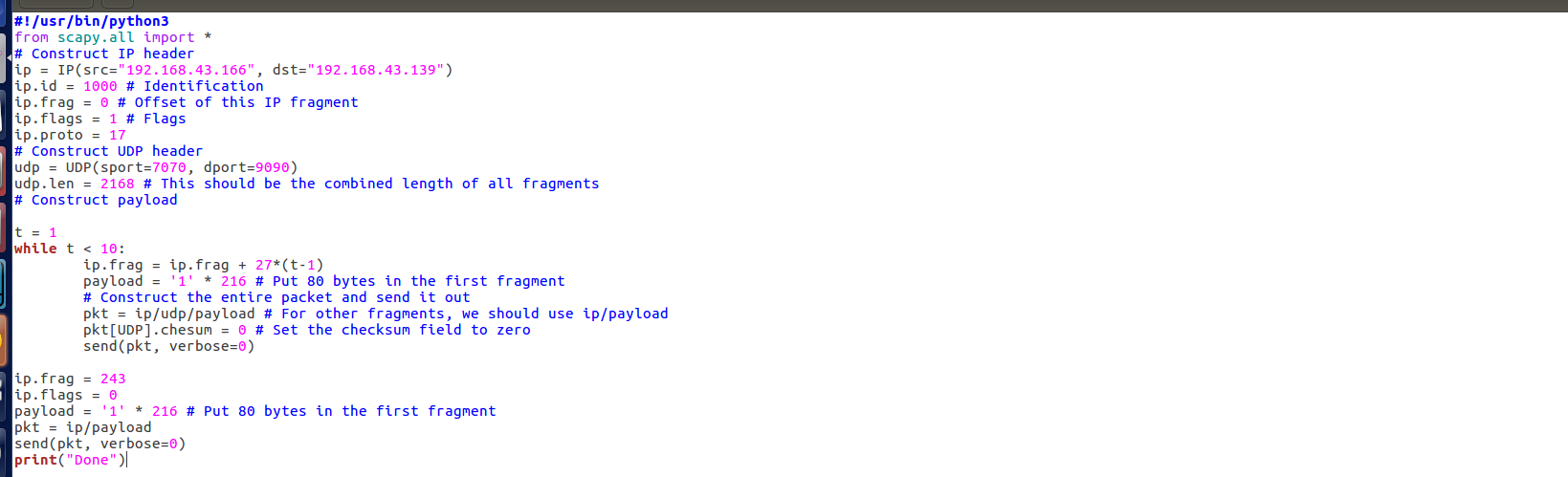
交换第一个片段和第二个片段的发送顺序后，结果不变：





**2.3 Task 1.c: Sending a Super-Large Packet**

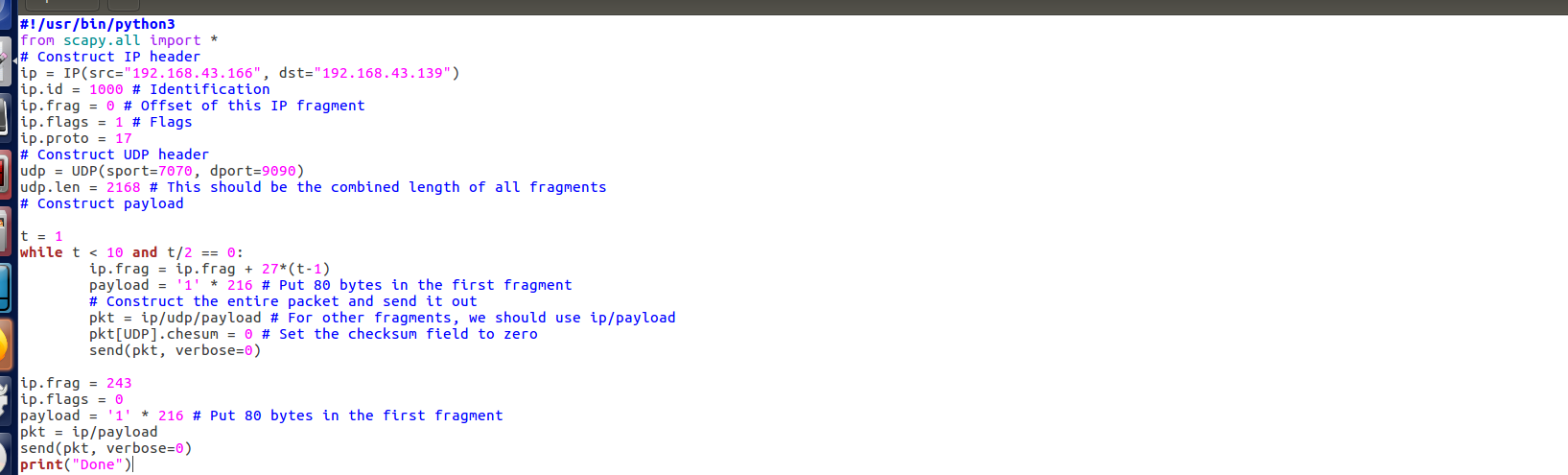
构造大于216个八位位组的数据包并发送至udp客户端：



被攻击者没有相应不良响应。

**2.4 Task 1.d: Sending Incomplete IP Packet**

向udp客户端发送大量不完整的IP数据包：



Wireshark观察不到相应异常，但理论上由不完整的IP数据包将保留在内核里，被攻击者内存将被严重占用。