**网络安全综合课程设计实验3**

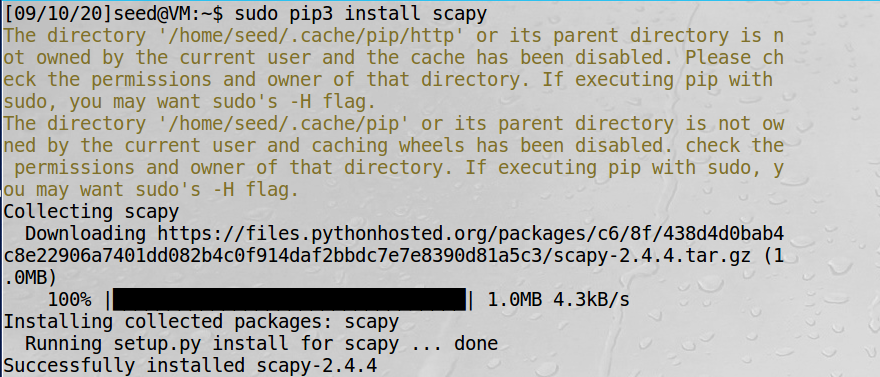
57117204 姚晓

2020/9/12

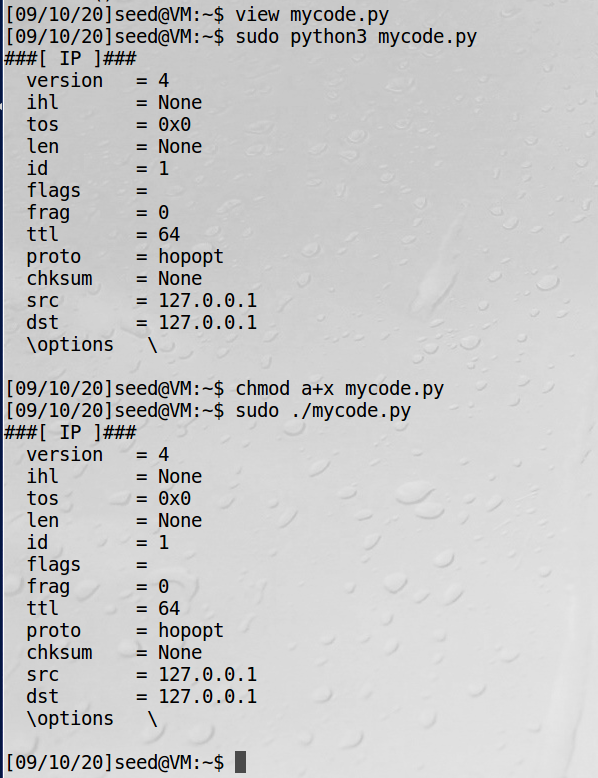
**Packet Sniffing and Spoofing Lab**

**Lab Task Set1: Using Tools to Sniff and Spoof Packets**

安装scapy：



运行实验手册上的代码，抓到如下报文：



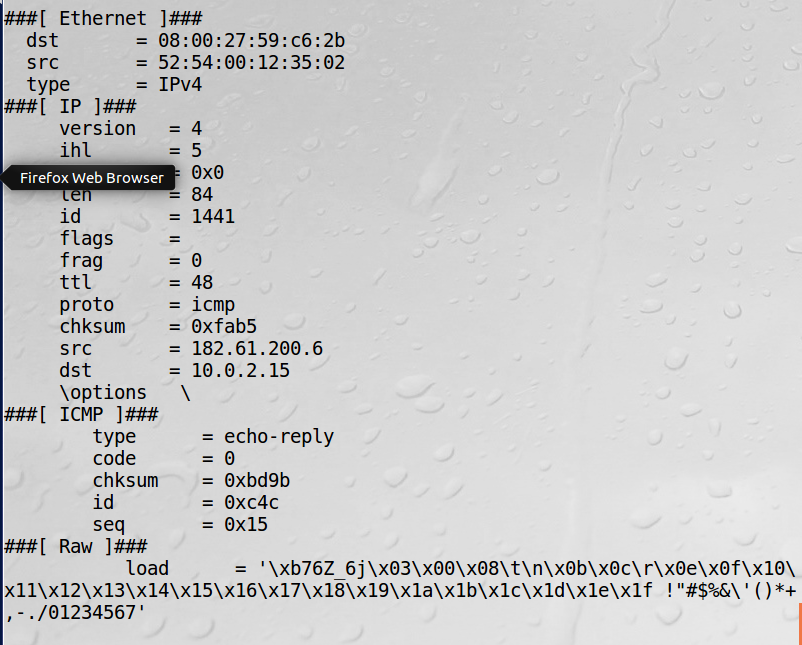
**Task1.1: Sniffing Packets**

**1.1A:**

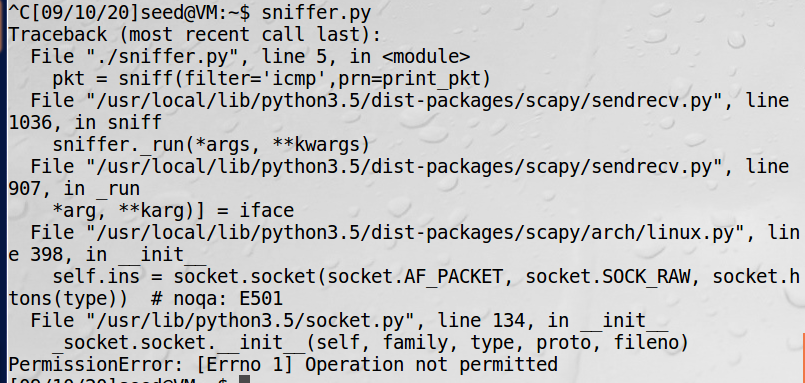
写入实验手册上的代码，并改为可编译模式：



用root模式运行代码，可以抓到报文：

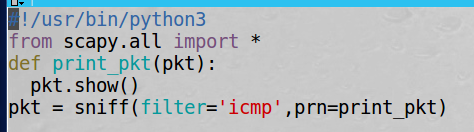


不是root权限运行代码，不能运行：

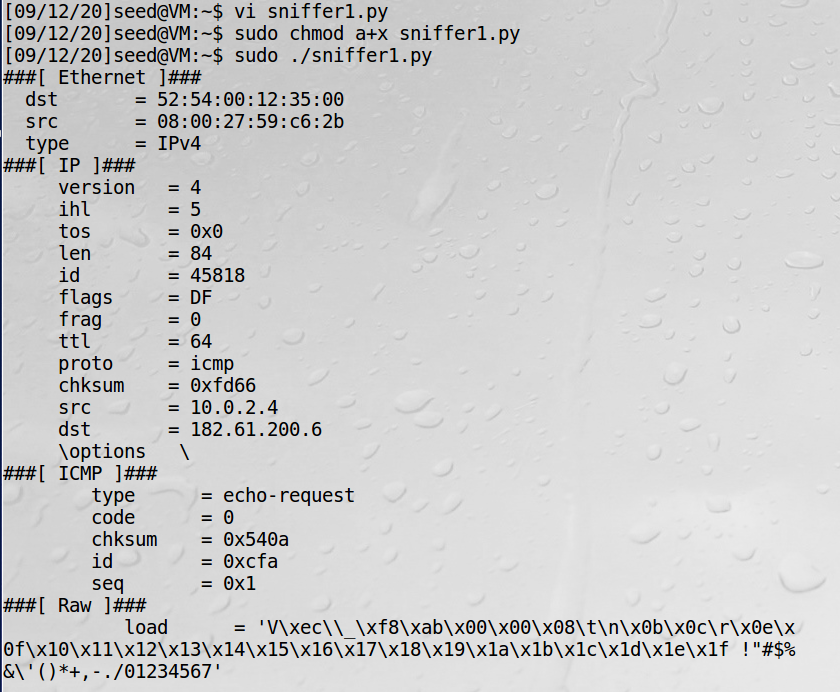


**1.1B:**

只捕获ICMP报文，将之前的代码的过滤器改为ICMP:



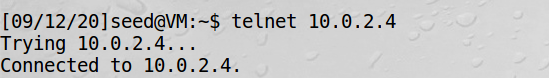
运行代码，得到结果：



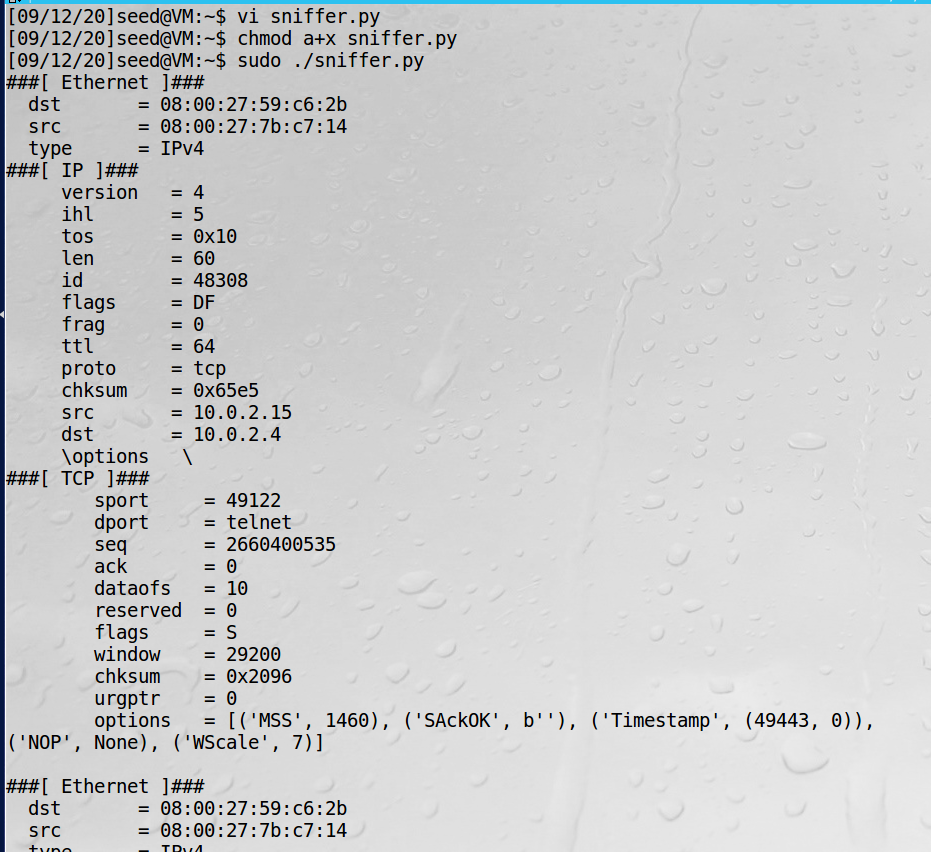
捕获来自特殊IP和目的端口为23的TCP报文，修改之前的代码，把过滤器改为如下所示：



然后运行该代码，在IP地址为10.0.2.15的虚拟机里使用telnet连接该主机：

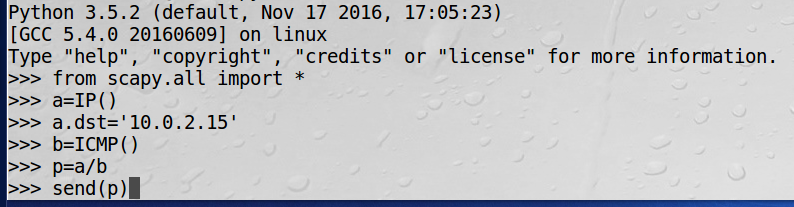


得到结果如下：



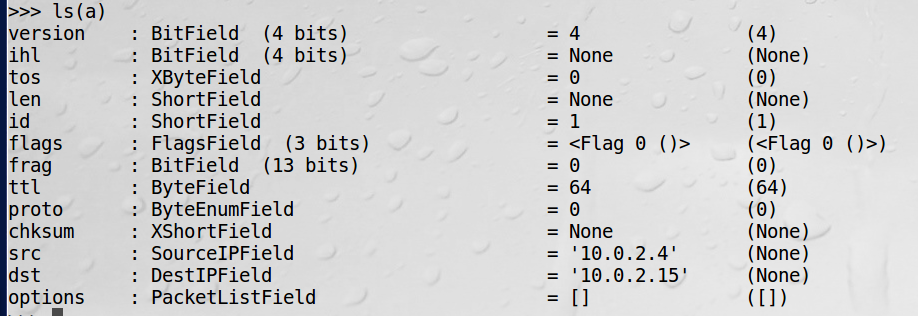
**Task1.2: Spoofing ICMP Packets**

运行实验手册给的代码，构造了一个ICMP报文，并发送到另外一台主机（IP地址为10.0.2.15）：



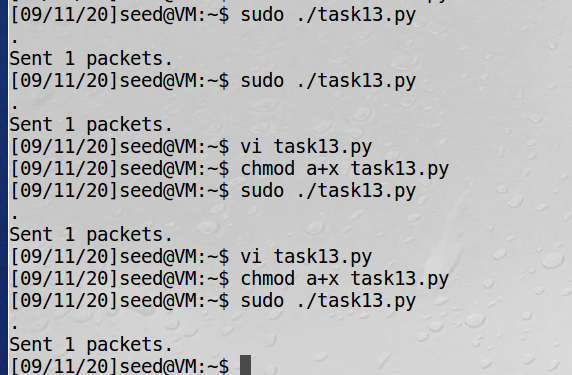
用wireshark抓包，得到结果，成功：





**Task1.3: Traceroute**

重复运行实验手册给的代码，第一次我设置的ttl=3，依次增加1：

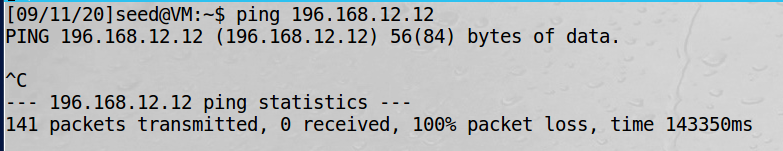


用wireshark抓包，发现ttl增加到5的时候就没有超时的报文了，所以估计跳数约为5.

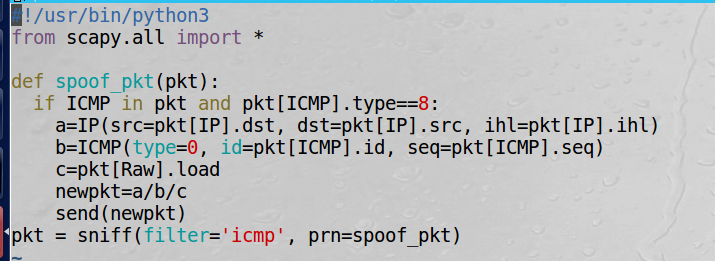


**Task1.4: Sniffing and-then Spoofng**

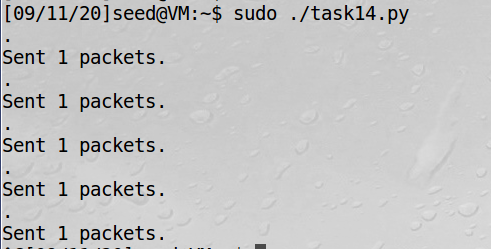
运行前在本机ping196.168.12.12（自己乱写的一个IP地址），ping不通：



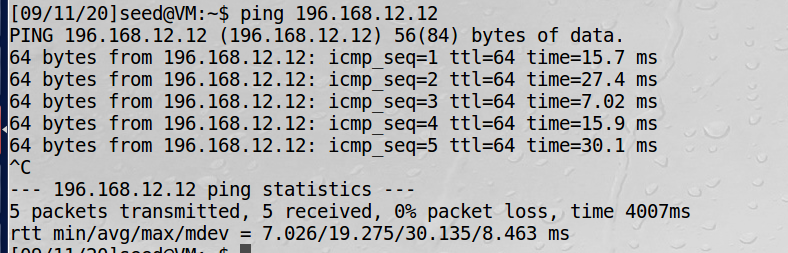
然后构造代码，该代码通过捕获ICMP报文，将其源地址和宿主地址对调，并设hiICMP类型为reply，再发出后就可以伪造ICMP的reply：



运行该代码，发送了报文表示伪造了报文：



运行后可以ping通，说明伪造成功：

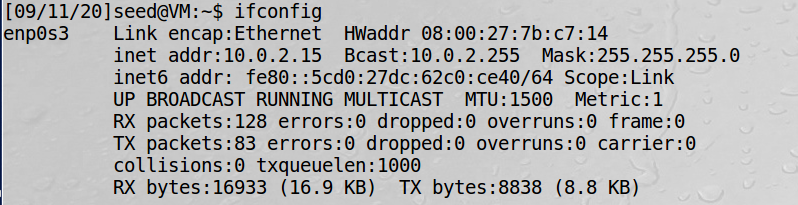


**ARP Cache Poisoning Attack Lab**

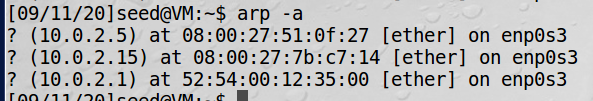
**Task1: ARP Cache Poisoning**

三个虚拟机分别为Ubuntu01（IP地址为10.0.2.4），Ubuntu02（IP地址为10.0.2.15），Ubuntu03（IP地址为10.0.2.5），实验目的是污染Ubuntu01的arp高速缓存表，使其中Ubuntu02IP地址对应的Mac地址变为Ubuntu03。

如下为Ubuntu02和Ubuntu03的信息和最开始的Ubuntu01的Arp表信息：

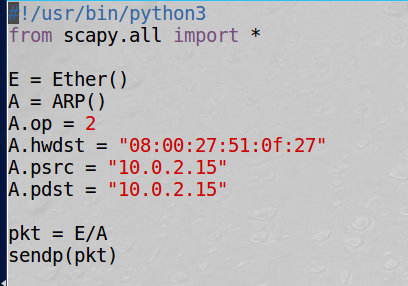


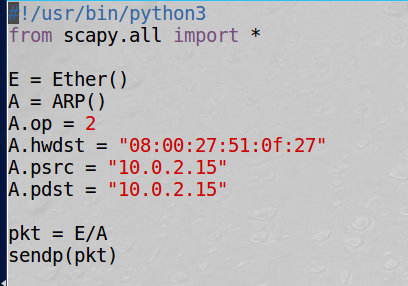




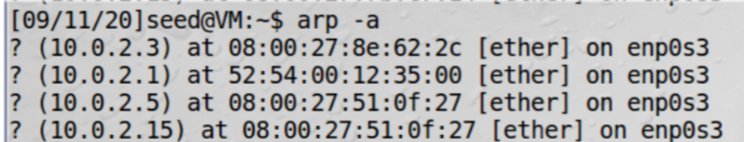
**Task 1A (using ARP request)**

构造ARP requst报文，代码如下：



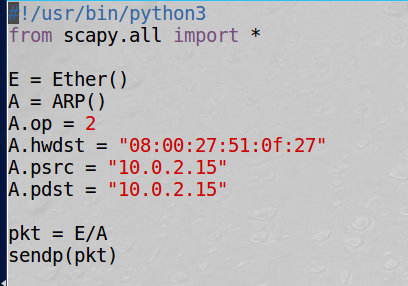


运行代码后，在Ubuntu01中查看arp表，得到如下结果，发现Ubuntu02的IP地址的MAC地址变成了10.0.2.5的MAC地址：



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

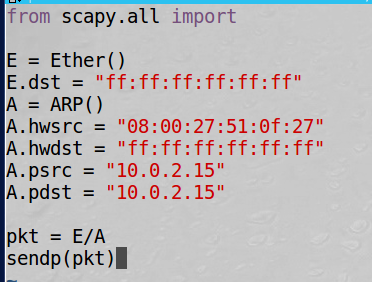
构造ARP reply报文：



运行结果同上。

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

构造广播的ARP gratuitious报文：



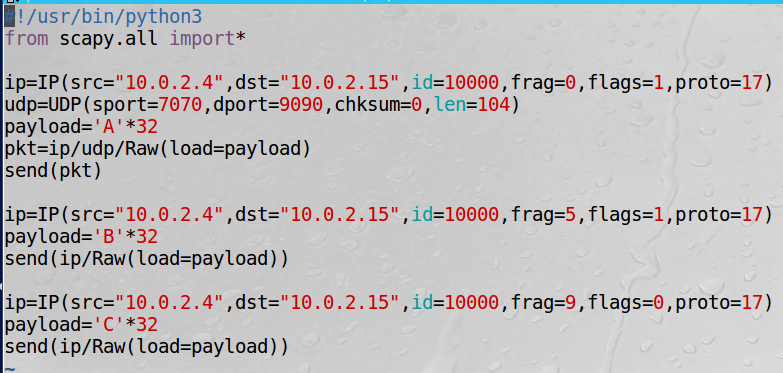
运行结果同上。

**IP/ICMP Attacks Lab**

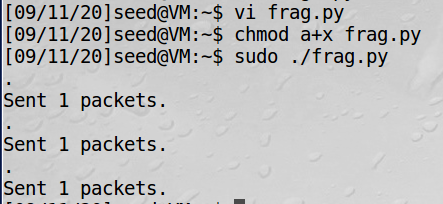
**Task1: IP Fragmentation**

**Task1.a: Conducting IP Fragmentation**

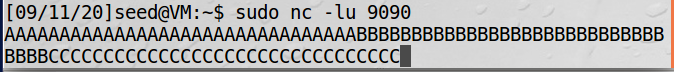
下面为构造IP分片的代码：其中他们的id都为10000，第一个分片的frag=0即偏移量为0，flags=1表示后面还有分片，第一个分片的frag=5即偏移量为（8+32）/8=5，flags=1表示后面还有分片，第一个分片的frag=9即偏移量为（40+32）/8=9，flags=0表示后面没有分片了，总长度len为8+32\*3=104。



运行代码发送了报文：

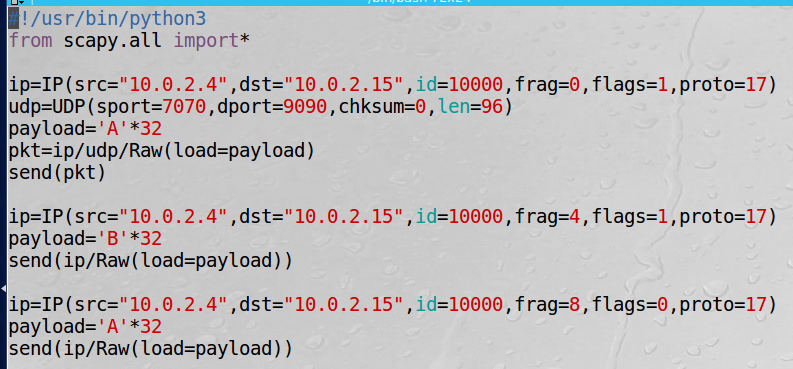


在另一个主机用nc –lu 9090命令得到结果：分别是32个A，32个B，32个C。

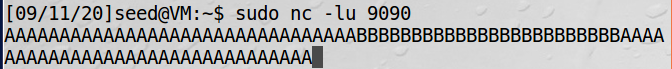


**Task1b: IP Fragments with Overlapping Contents**

修改代码让第二个分片的偏移量为4，第三个分片为8：

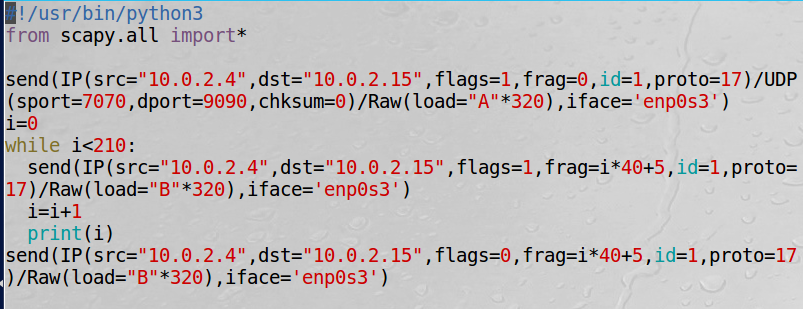


得到结果发现只有32个A和24个B和32个A，即有一部分被覆盖了：



**Task1.c: Sending a Super-Large Packet**

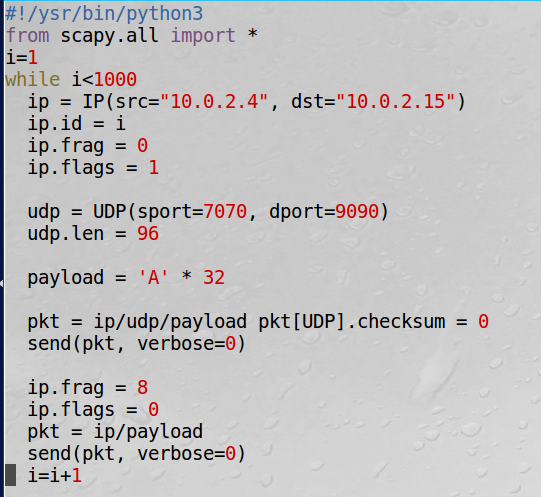
修改代码，循环给另一台主机发送超大的报文：



通过分片发送大于65320字节的报文，接收方在收到超过65320的报文后偏移量从0开始继续增加。

**Task1.d: Sending Incomplete IP Packet**

修改代码，id不断改变，不发送第二个分片，只发送第三个：



受攻击主机变得缓慢，内存占用高。