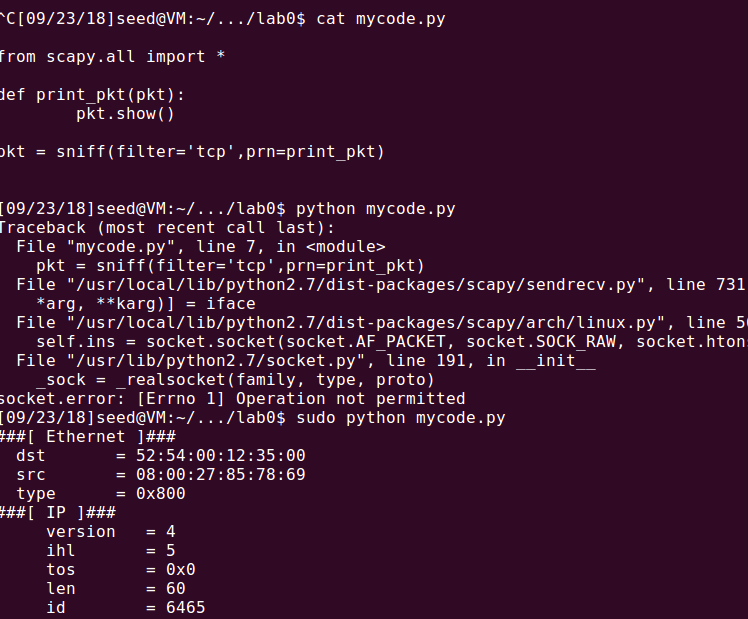
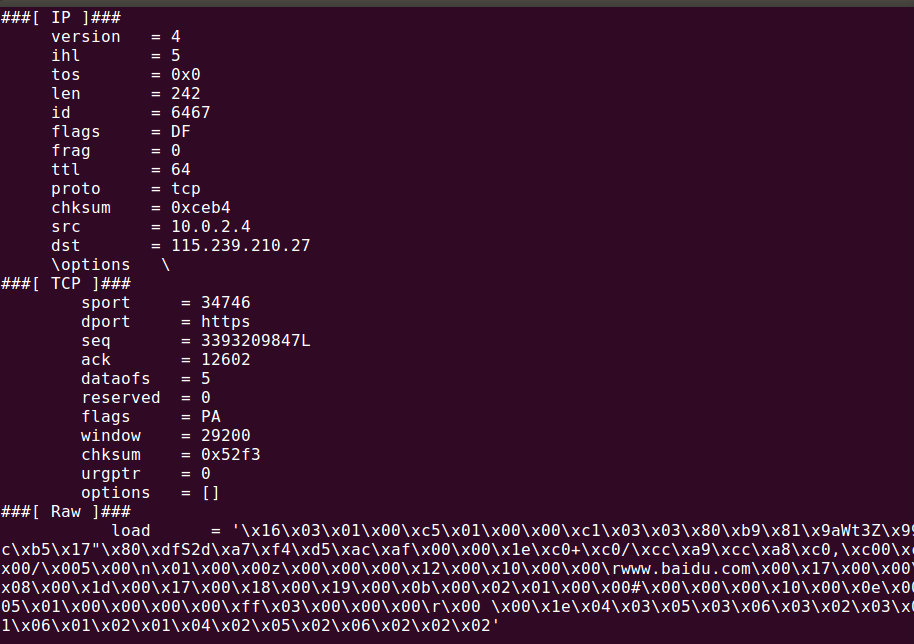
Lab report

Task Set 1: Using Tools to Sniff and Spoof Packets

Task 1.1: Sniffing Packets

Task 1.1A.

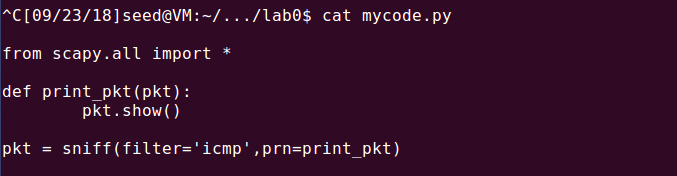
可以观察到，当root权限运行时，并且打开浏览器的网站时，会有大量的数据包打印在屏幕上，可以证明能收到网络包。若没有在root权限下运行程序，则程序无法运行，因为网络数据包在网卡上接受，要获取数据包需要通过底层的硬件，而这是内核程序控制，一般程序在无权限的情况中，无法直接与底层硬件交互，所以需要在root权限下运行此程序。



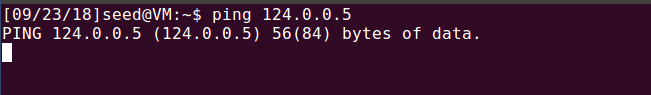
Task 1.1B

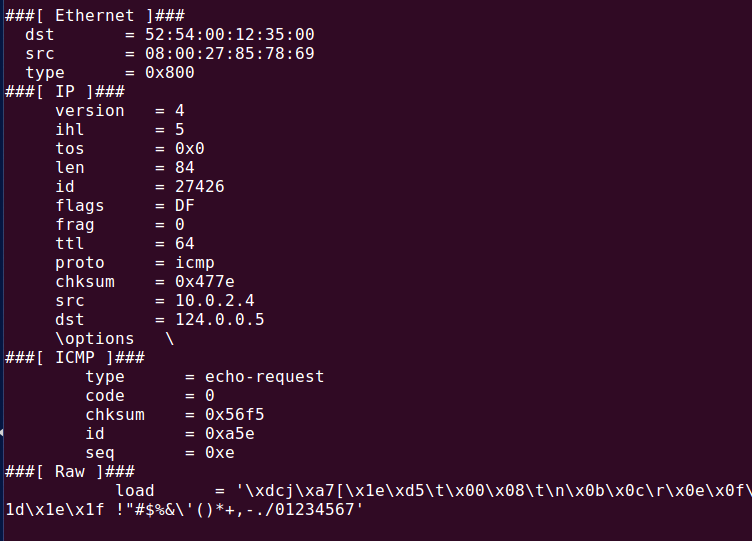
Capture only the ICMP packet

代码：



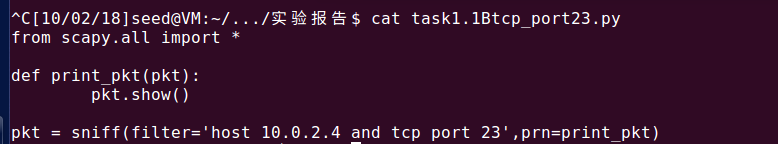
通过ping 124.0.0.5（即另一台未启动的虚拟机网址）获取icmp包。



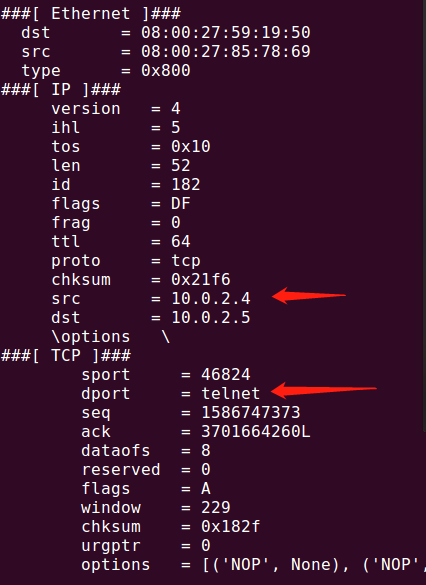


Capture any TCP packet that comes from a particular IP and with a destination port number 23

抓取网络包，sniff的filter函数，IP:10.0.2.4 port:23 以及是tcp报文 Code:



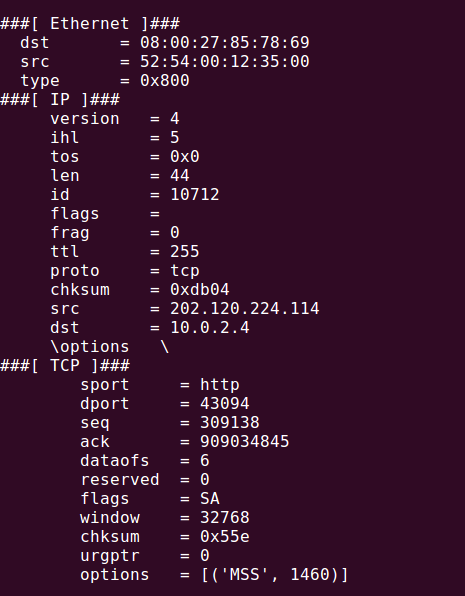
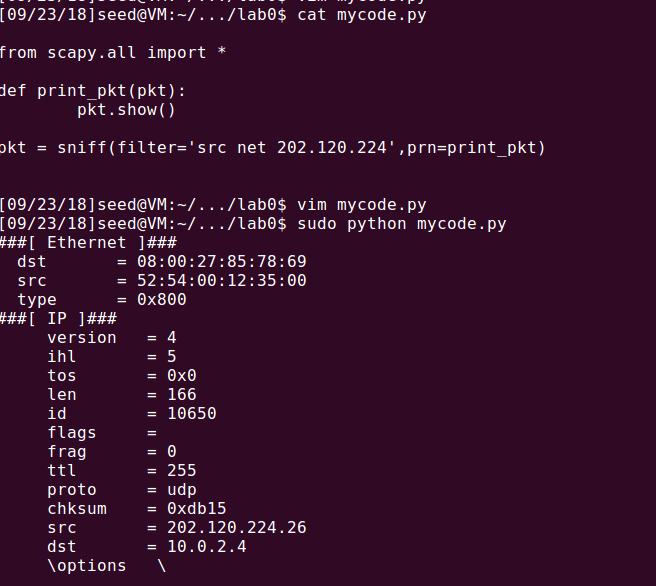
抓取的结果是：



Capture packets comes from or to go to a particular subnet. You can pick any subnet, such as

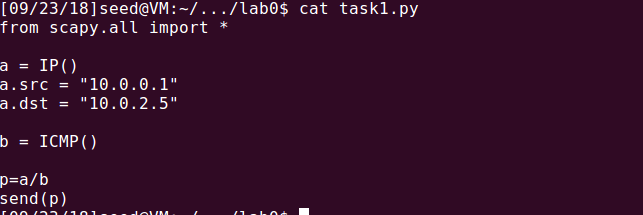
128.230.0.0/16; you should not pick the subnet that your VM is attached to.

选择的子网是source 202.120.224.0/24



Task 1.2: Spoofing ICMP Packets

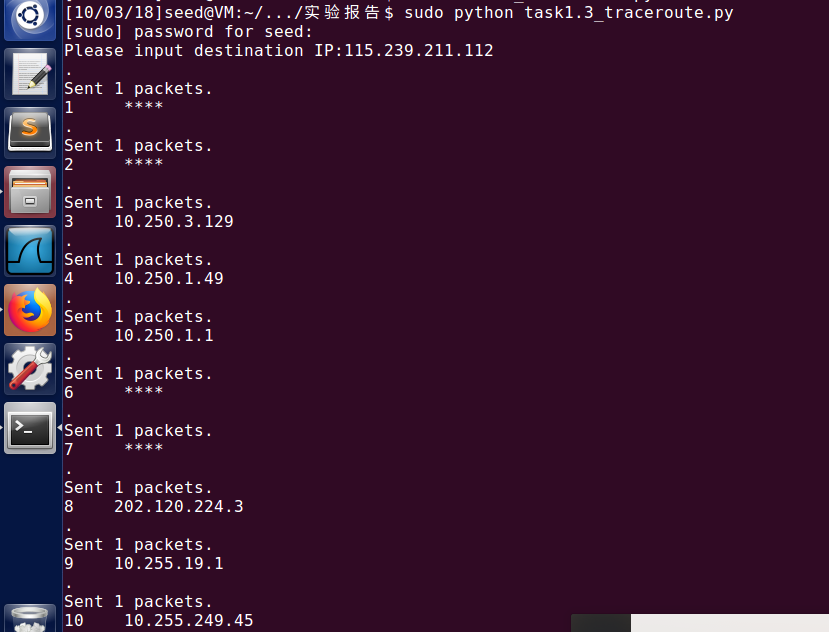
简单的欺骗，代码以及获取的结果：

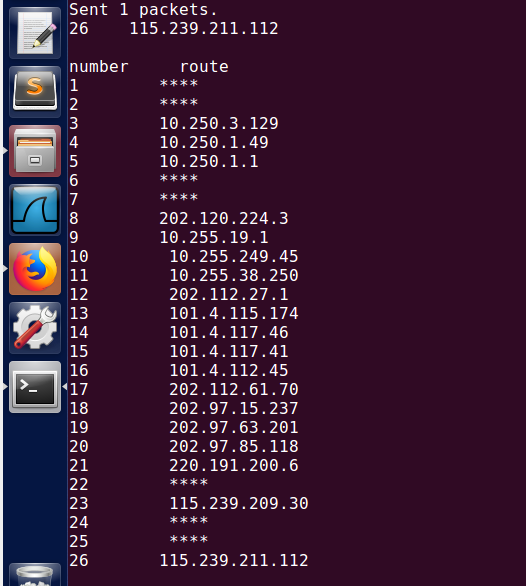


C:\Users\ERIC_H~1\AppData\Local\Temp\WeChat Files\fb34c67b09d82e36fe3f58d3258dfb2.png

Task 1.3: Traceroute

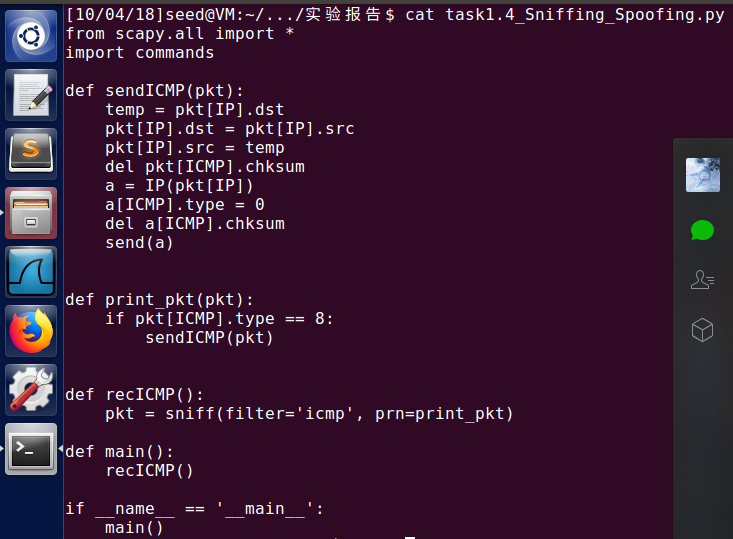
为了处理收包和发包以及程序的结束，我使用了线程以及信号量来处理细节，通过查阅scapy文档，发现scapy有自带的路由函数，可以直接使用。下图是我获取[www.baidu.com的一个IPv4](http://www.baidu.com的一个IPv4)的地址来获取其路由路径。发现其发送包时有echo，最后我通过文件的形式保存数据，得到其完整的路由路径。相关程序：task1.3\_traceroute.py



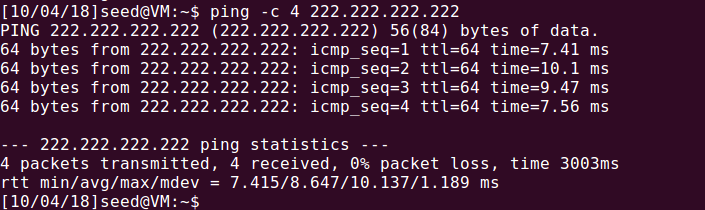


Task 1.4: Sniffing and-then Spoofing

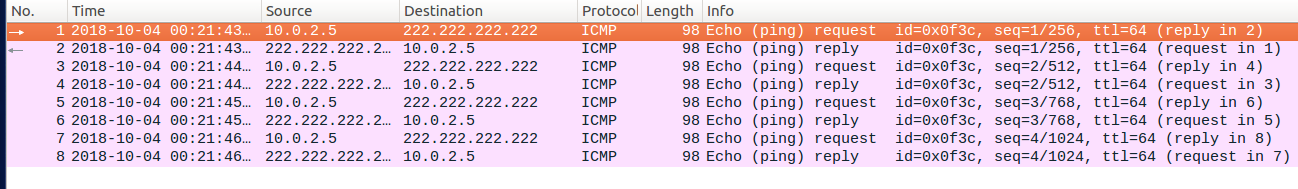
代码如下图：



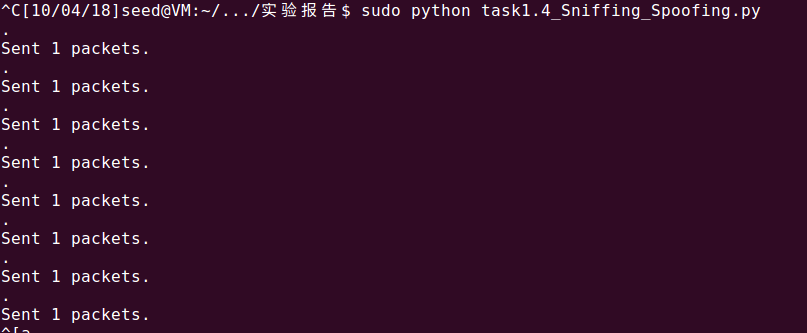
在虚拟机B（ip 10.0.2.5）上ping 222.222.222.222，在虚拟机A(Ip 10.0.2.4)上运行伪造报文程序，得到如图所示回复：



通过wireshark获取包的信息如图：



观察VMA上的程序的回显：



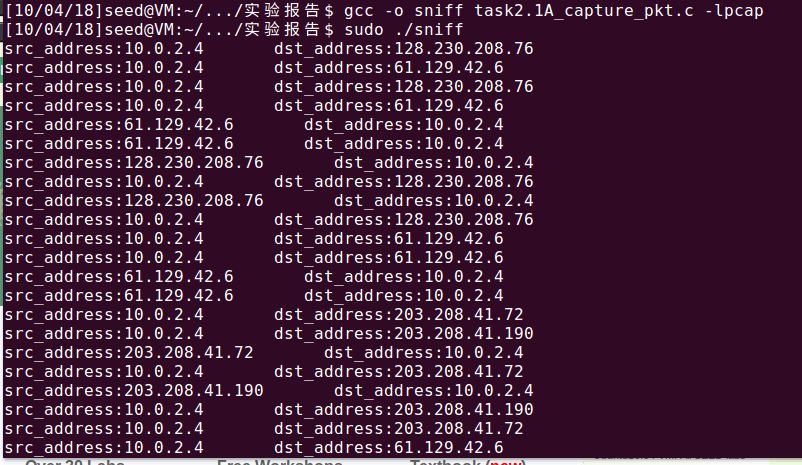
说明伪造过程完成。

Lab Task Set 2: Writing Programs to Sniff and Spoof Packets

Task 2.1: Writing Packet Sniffing Program

Task 2.1A: Understanding How a SnifferWorks

代码task2.1A\_capture\_pkt.c，效果如图：



• Question 1. Please use your own words to describe the sequence of the library calls that are essential

for sniffer programs. This is meant to be a summary, not detailed explanation like the one in the

tutorial or book.

首先，我们需要设置抓包的网卡，让其在获得包后，能够交给sniff去处理，所以接着必须要sniff能够与网卡建立连接，即会话机制。然后根据所设置的filter来选择相关的报文，将过滤好的报文交给sniff的相关程序，然后把报文解析后，根据其程序来进行不同的操作，最后当我们完成我们的需求或者完成相关的程序后，结束我们的会话。关闭程序，恢复其网卡的设置。

• Question 2. Why do you need the root privilege to run a sniffer program? Where does the program

fail if it is executed without the root privilege?

需要root权限时因为他要访问网络接口，获取其底层的硬件交互。前面已经说明Linux中没有root访问权限时不可能网卡的。而Sniff需要原始套接字，允许应用程序直接发送数据包括，绕过操作系统网络软件中的所有应用程序。

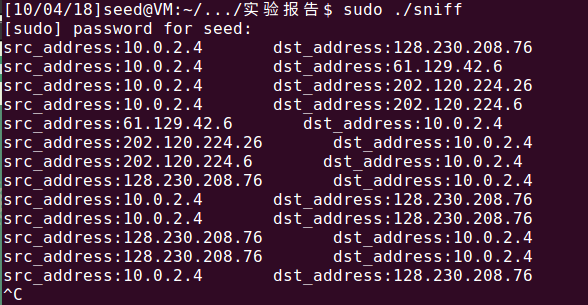
• Quesiton 3. Please turn on and turn off the promiscuous mode in your sniffer program. Can you

demonstrate the difference when this mode is on and off? Please describe how you can demonstrate

this.

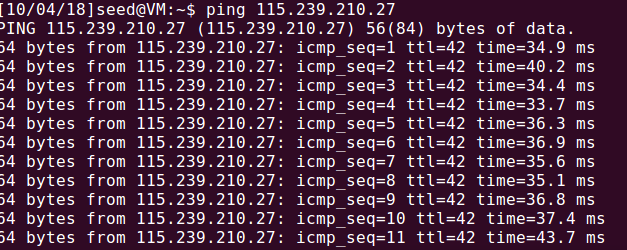
混杂模式的设置在pcap\_open\_live（）函数的第三个参数，0为混杂模式，1为非混杂模式，所以我们可以通过修改参数来开启混杂模式。我们可以通过其抓包的源地址和目的地址是否有本机ip来证明它，如下图一个中所受到的包地址中必含有10.0.2.4，而另一个是没有这中情况的，且源地址和目的地址更加多样化。





Task 2.1B: Writing Filters

ICMP:使用BPF "icmp and host 115.239.210.27 and 10.0.2.4"来过滤，（代码：task2.1B\_filter\_ICMP.c）,结果如下图：



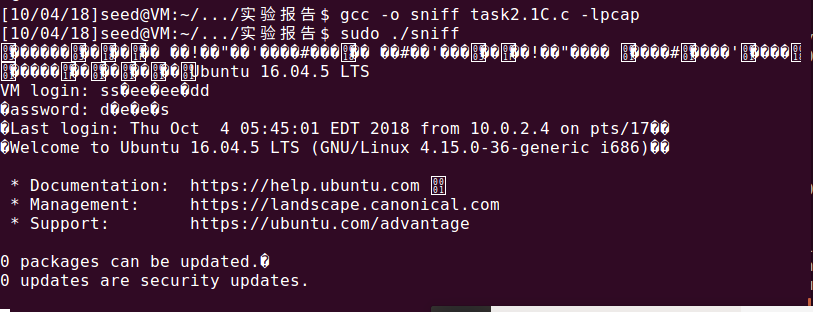


TCP :使用"tcp dst portrange 10-100"来过滤，（代码：task2.1B\_filter\_TCP.c）结果如图：



Task 2.1C: Sniffing Passwords.

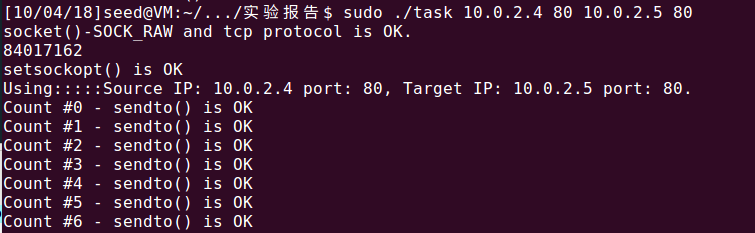
通过打印所抓的包，我们可以看到每次写入是，请求连接后输入命令后会有回显，但是密码没有回显，通过打印telnet的data，我们可以得到敏感信息，包括登录账号和密码：如下图密码：dees，账号：seed



Task 2.2: Spoofing

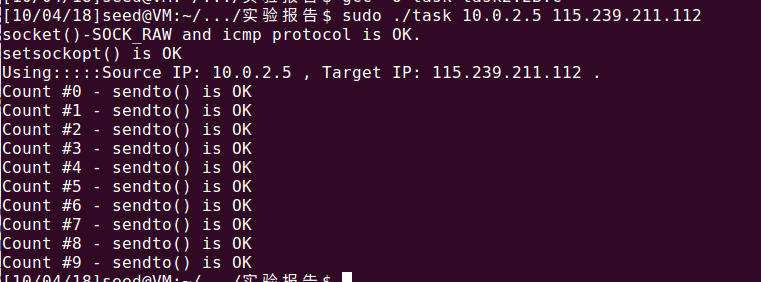
Task 2.2A: Write a spoofing program.

C:\Users\ERIC_H~1\AppData\Local\Temp\WeChat Files\917863f26235bbbf1ba51816e903ee8.png

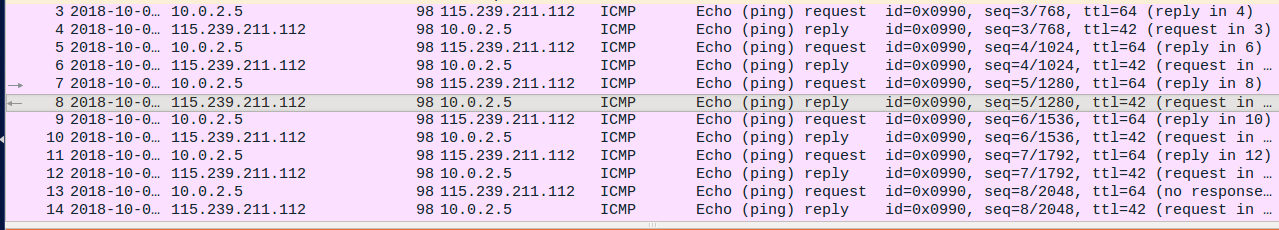


Task 2.2B: Spoof an ICMP Echo Request.

发送报文：



收到报文：



Questions. Please answer the following questions.

• Question 4. Can you set the IP packet length ﬁeld to an arbitrary value, regardless of how big the actual packet is?

不可以，因为当长度太长或与实际不符合是，系统会认为报文已经错误，不会把它继续传下去。

• Question 5. Using the raw socket programming, do you have to calculate the checksum for the IP header?

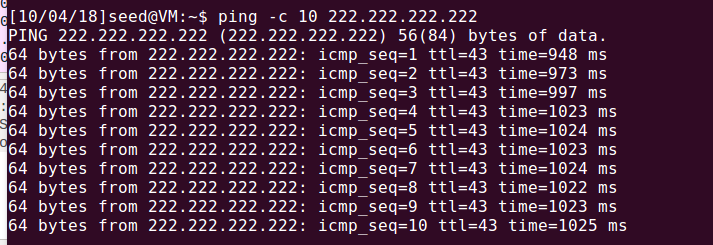
需要ICMP的校验和，我并不需要计算IPheader的校验和。

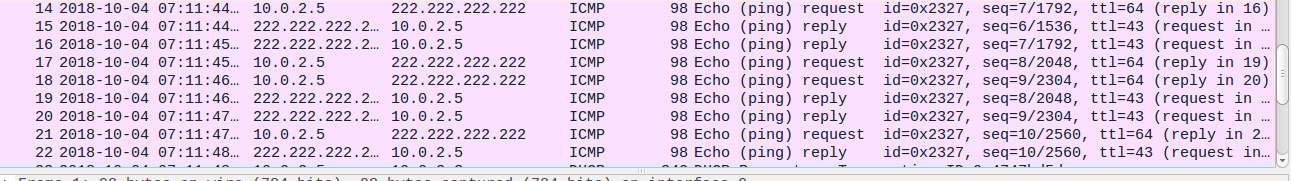
• Question 6. Why do you need the root privilege to run the programs that use raw sockets? Where does the program fail if executed without the root privilege?

因为我们需要直接访问网卡，直接发送报文，而不是应用程序。并且raw\_socket规则我们可以任意写，如果不用root权限，将会是我们的网络无法正常运转。且一般低于1024的端口号，如果没有root权限是无法绑定的。

Task2.3: Sniff and then Spoof

程序代码在task2.3.c中，我用VMB ping 222.222.222.222这个没有注册的IP地址，可以发现它能够正常回复显示此正常地址，说明我们伪造的报文是正确的：





发送信息：

