LAB 1 57118226舒钰淇

Task 1.1: Sniffing Packets

1. 将docker启动后查看其中的网络接口ID，并将sniffer的python程序中嗅探的接口设置为网桥。

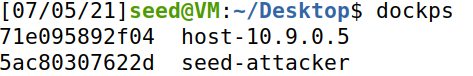


图1.1 网络接口ID



图1.2 sniffer.py

以root权限运行sniffer.py，ping docker中的主机IP时，可以嗅探到相应的数据包信息。

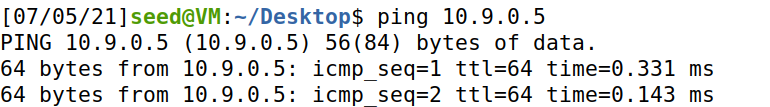


图1.3 ping docker中的主机IP

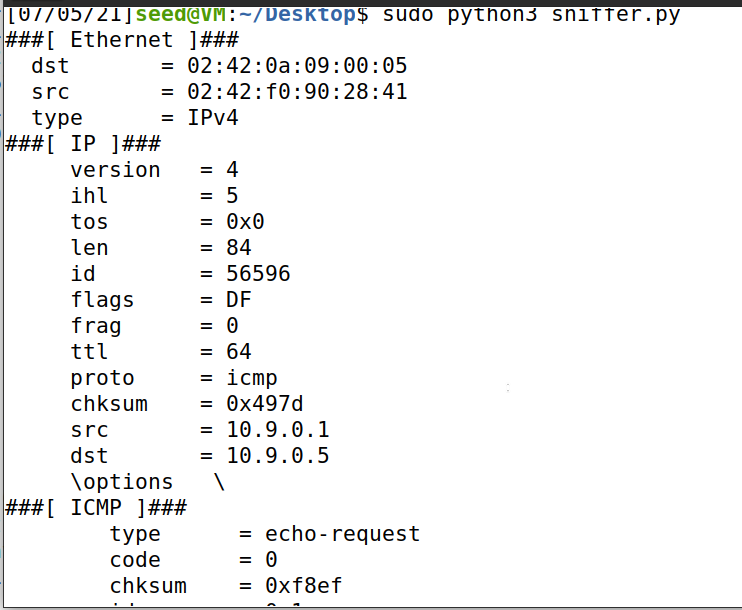


图1.4 嗅探到的数据包信息

直接在seed用户下运行程序，提示无权限。

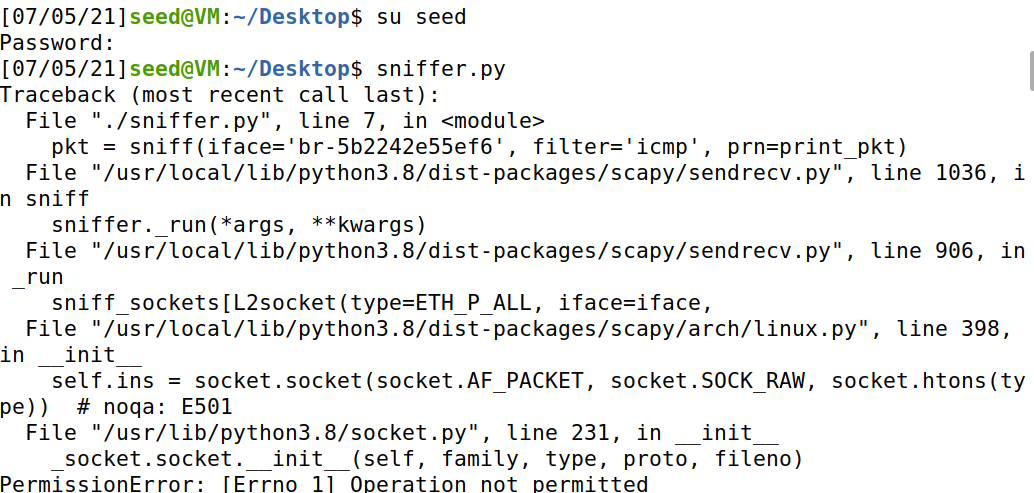
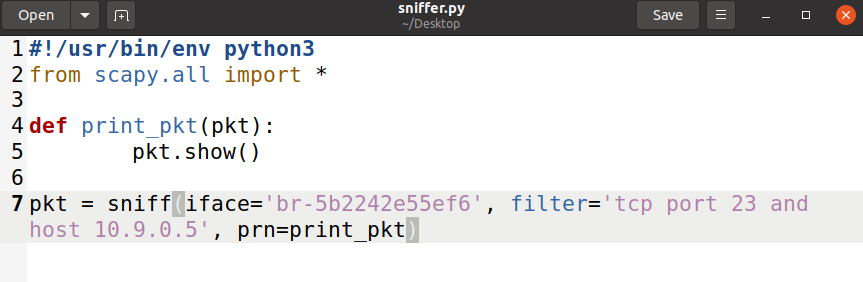


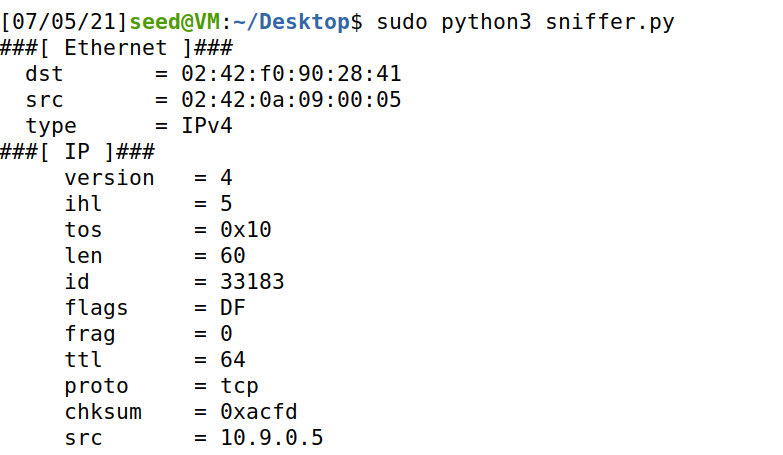
图1.5 无权限执行操作

1. BPF的使用
2. 只抓取ICMP报文如task1.1A所示
3. 捕获来自特定IP地址，端口为23的报文

修改sniffer.py如下图所示，捕获来自docker主机IP，端口为23的报文

图1.6 sniffer.py

运行sniffer.py，在docker主机上telnet 1.1.1.1，可以捕获到相应的报文



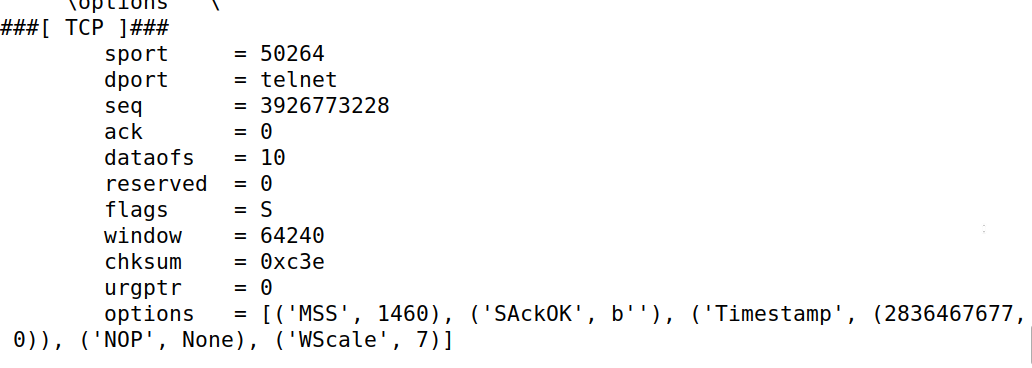


图1.7捕获到的telnet报文

1. 捕获来自或去向某一子网的数据包

修改sniffer.py如下图所示（由于中间重启过docker环境，所以网络ID改变），捕捉子网2.2.2.0下的报文

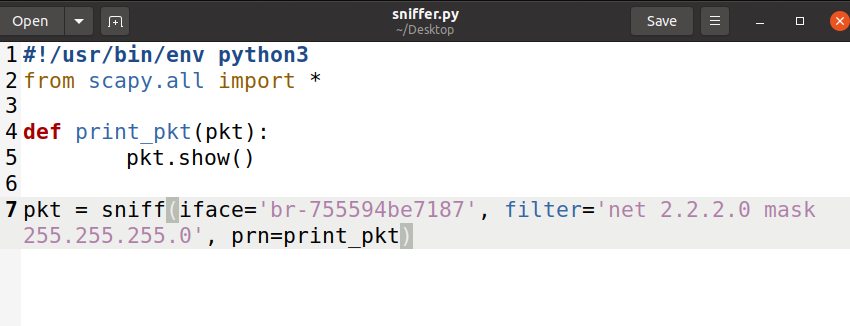


图1.8 sniffer.py

运行sniffer.py，在docker主机上telnet 2..2.2.2，可以捕获到相应报文。

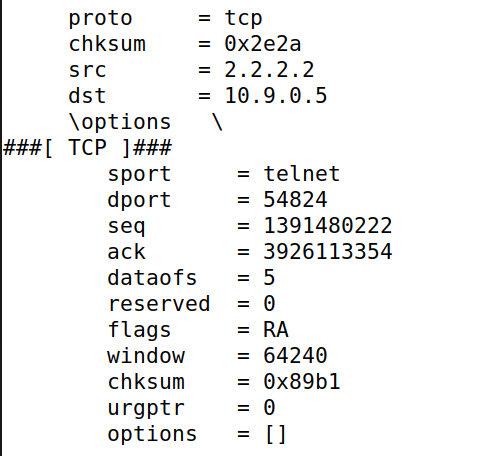


图1.9捕获到的telnet报文

Task 1.2: Spoofing ICMP Packets

用如下所示的spoofing.py，构造一个发送向docker主机10.9.0.5，伪造源地址为10.9.0.4的icmp报文。

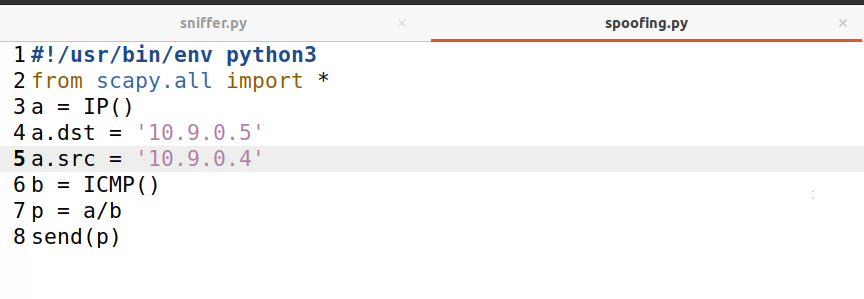


图2.1 伪造发送ICMP报文的spoofing.py代码

使用TASK 1.1A的sniffer.py对docker主机进行嗅探，运行spoofing.py，可以捕获到伪造的ICMP报文。

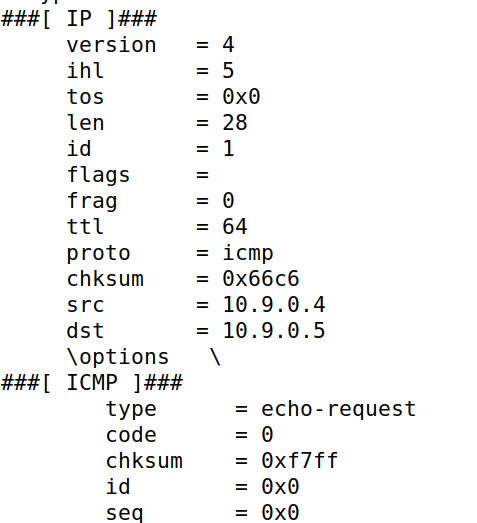


图2.2 捕获到的伪造ICMP报文

Task 1.3: Traceroute

编写如下所示的traceroute.py，向百度的IP地址发送ICMP报文，通过修改TTL来查看途径的路由器。



图3.1 traceroute.py



TTL=1



TTL=3



TTL=5



TTL=6



TTL=7



TTL=8



TTL=9



TTL-10

TTL=10时，收到ICMP答复

Task 1.4 Sniffing and-then Spoofing

编写如下ICMP欺骗程序，当嗅探到docker主机ICMP请求报文时，发送对应的答复报文。

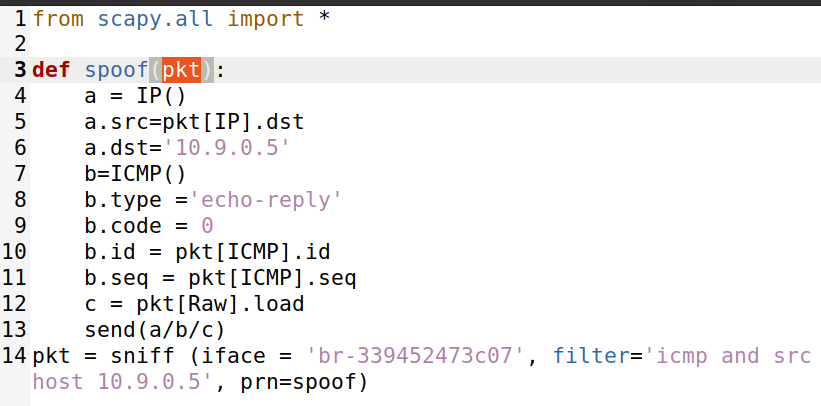


图4.1 py程序

先在docker主机上ping 1.2.3.4,显示Unreachable，在虚拟机上打开py程序，docker主机上收到了虚拟机发送的伪造ICMP答复报文。

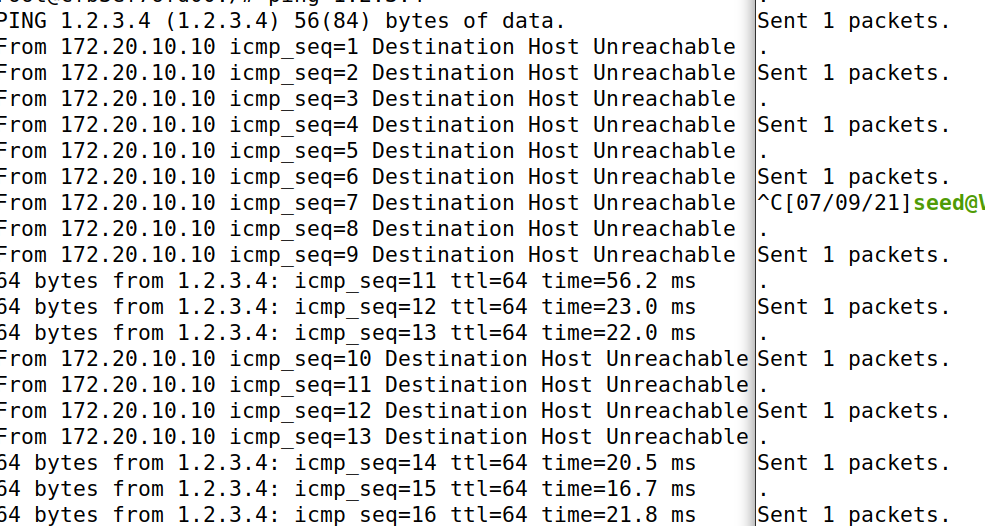


图4.2 ping 1.2.3.4

继续运行py程序，在docker主机上ping局域网上不存在的10.9.0.99,由于会先使用ARP询问，所以没收到ICMP答复报文。

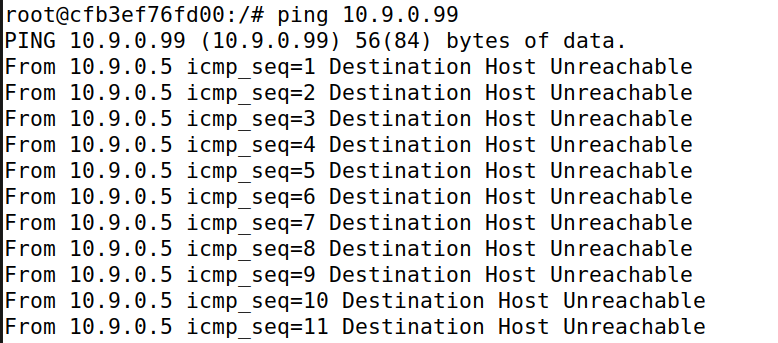


图4.3 ping 10.9.0.99

在docker主机上ping存在的外网IP，由于会收到来自外网的ICMP回复和虚拟机发出的伪造ICMP答复，因此出现DUP提示。

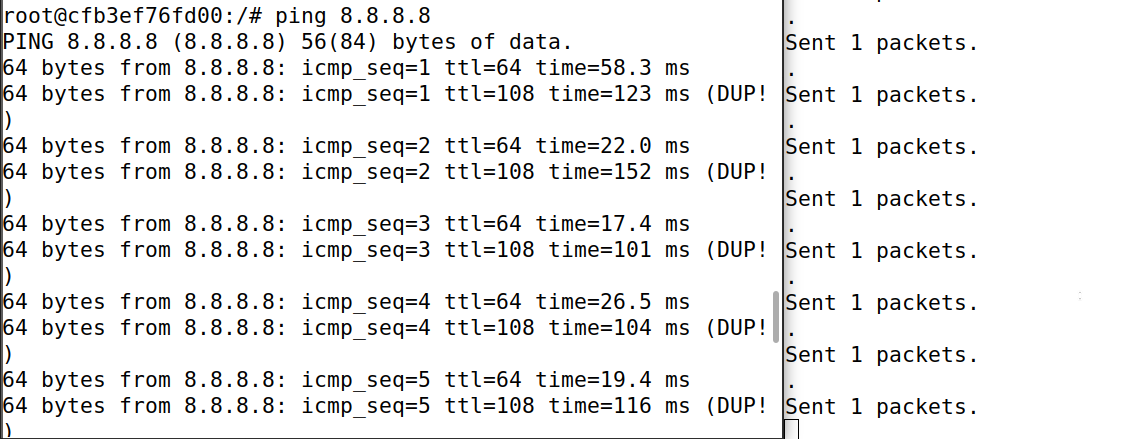


图4.4 ping 8.8.8.8