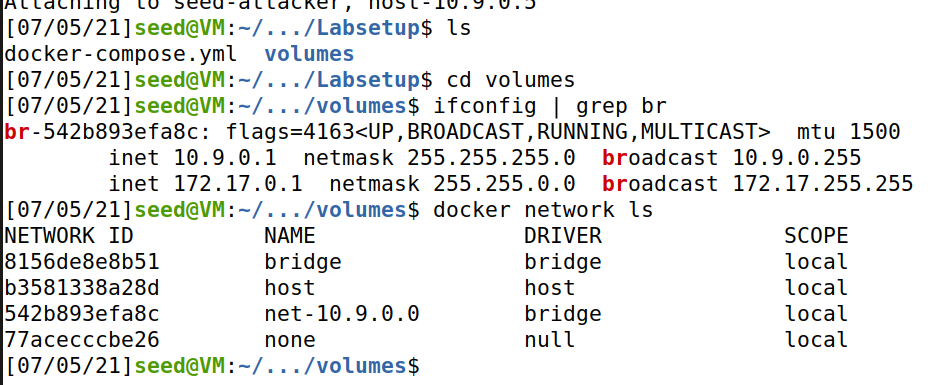
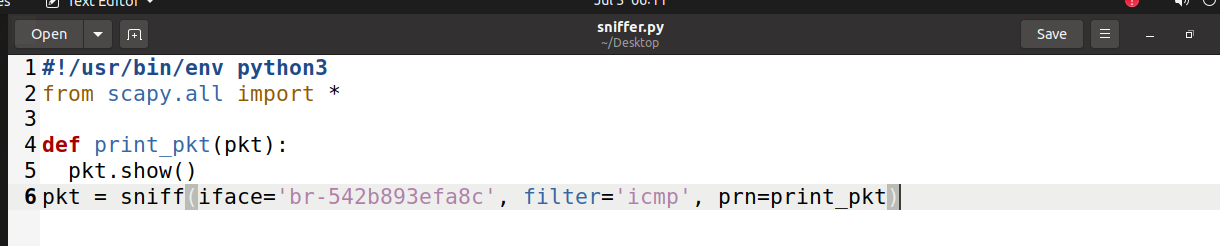
**Task 1.1A**

**进入实验所在文件夹，在docker中下执行命令 ifconfig | grep br**

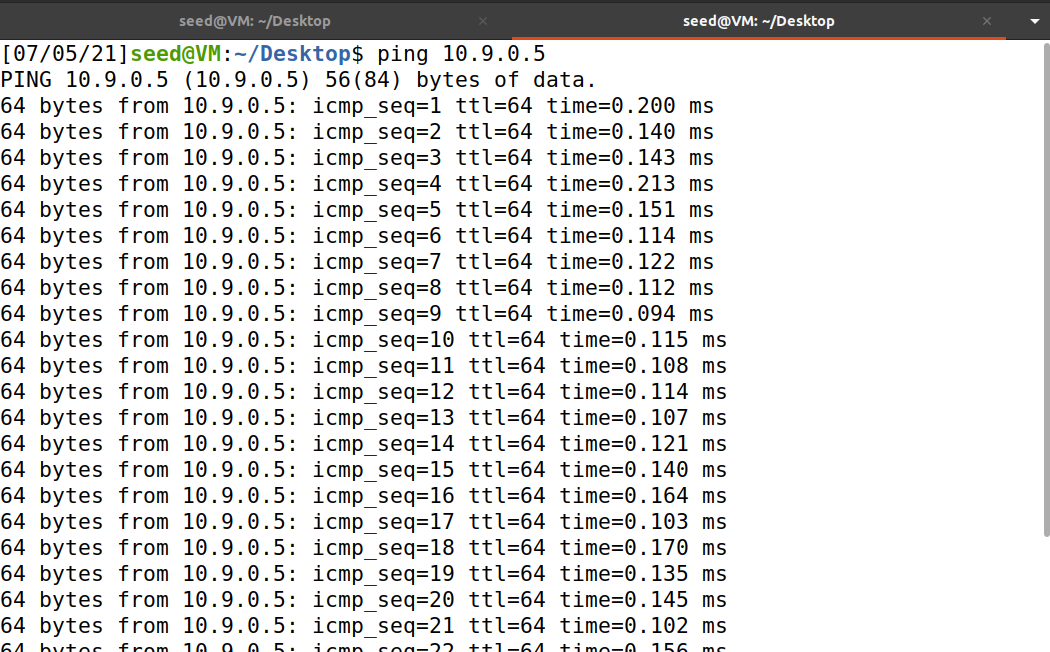
**获得br值**



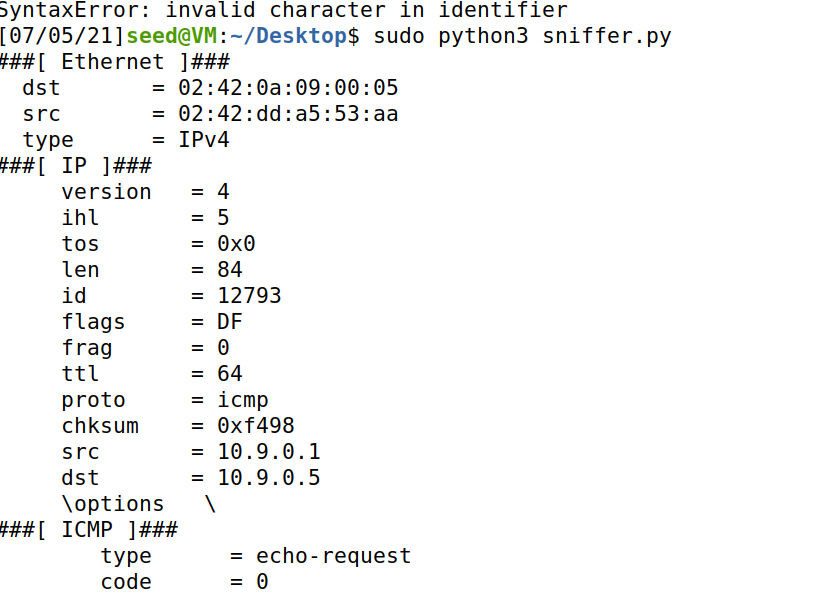
**创建sniffer.py进行流量嗅探**

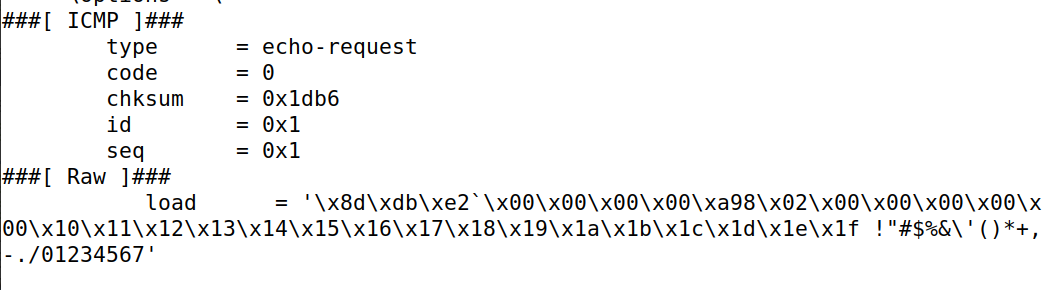


**Ping一个网址**

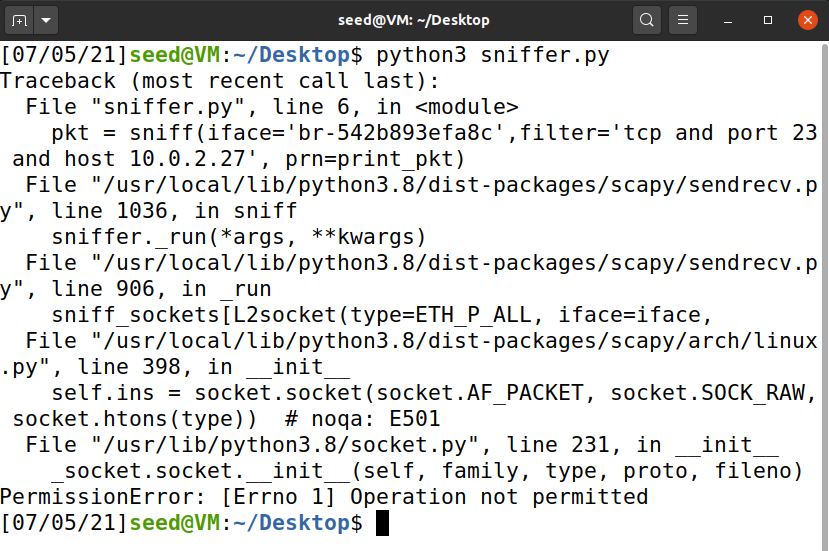


**在root权限下运行，抓包结果如下**





**不使用root权限，直接报错 [Errno 1] Operation not permitted**

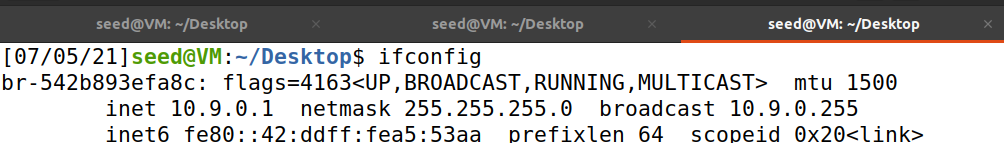


**Task 1.1 B**

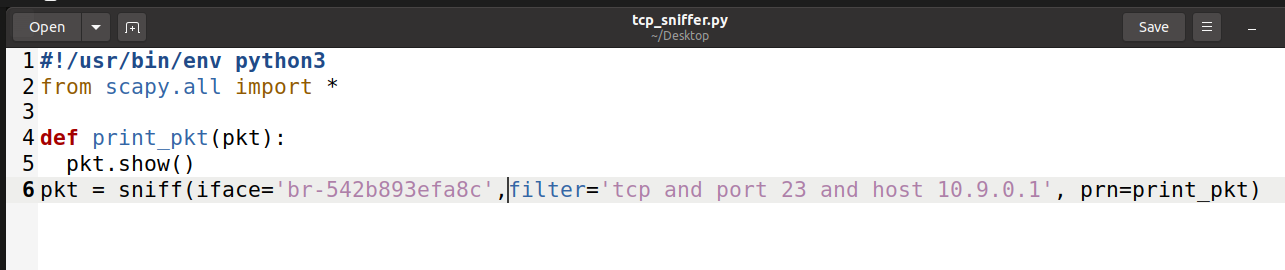
（1）捕获ICMP报文，操作与1.1A相同

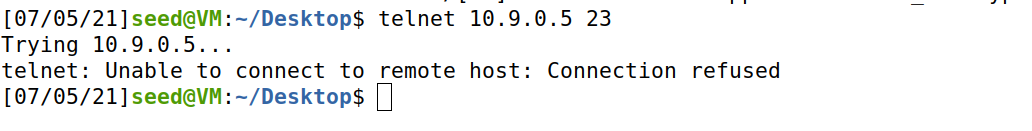
（2）捕获来自特定 IP 且目标端口号为 23 的任何 TCP 数据包

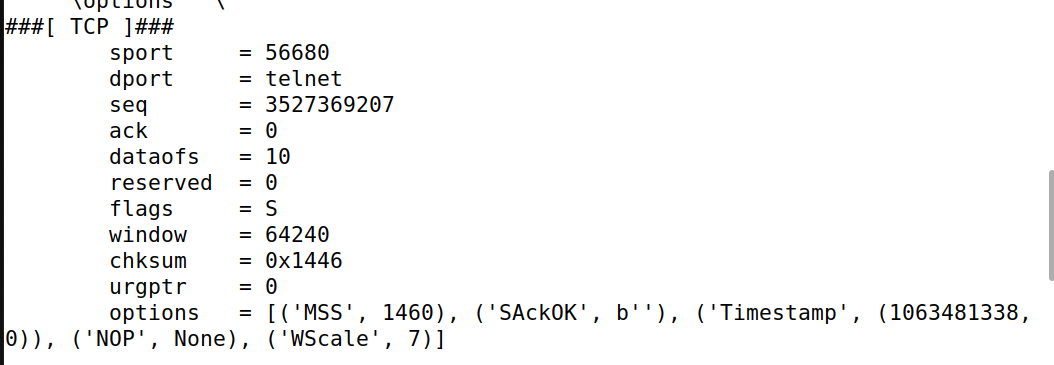
获取seedlab的ip



编写脚本程序：





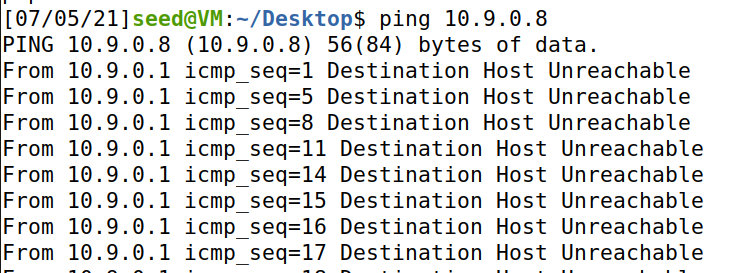


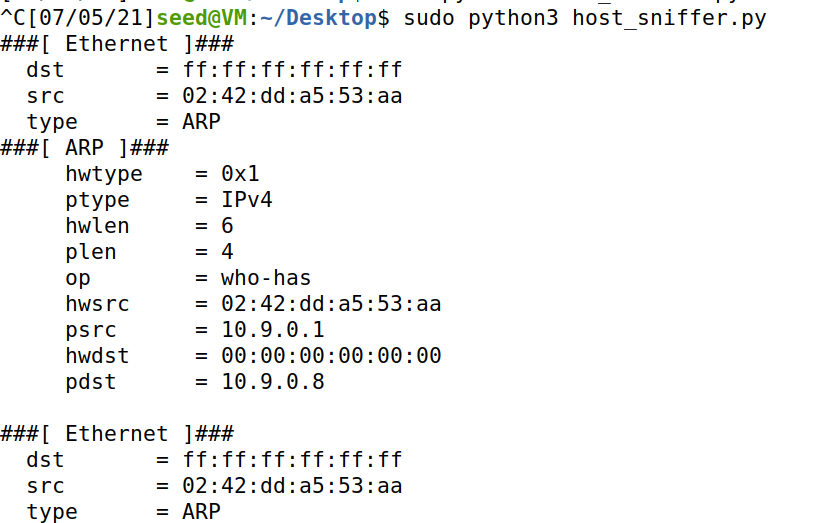
**（3）捕获来自特定子网的报文**

**编写脚本：**



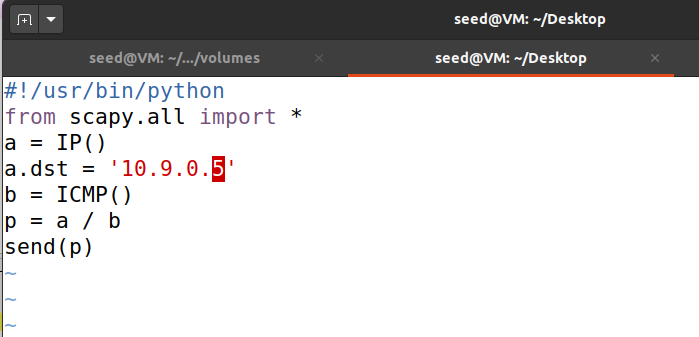
**嗅探结果如下：**

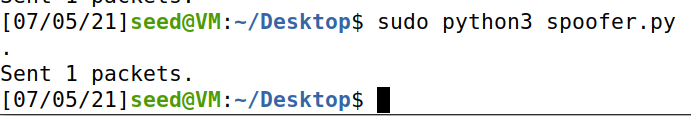




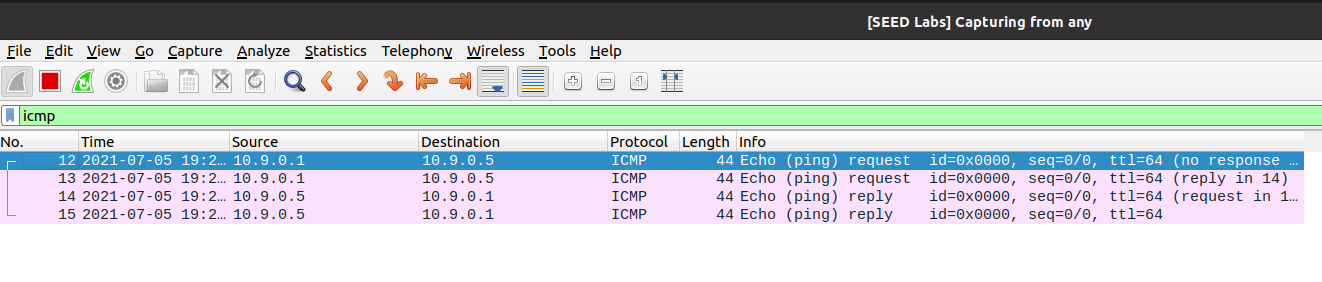
**Task 1.2: Spoofing ICMP Packets**

**编写脚本并运行**





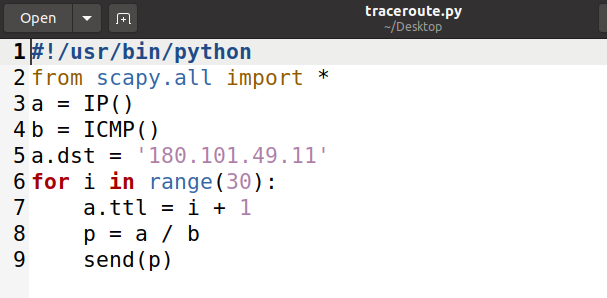
**在wireshark中抓包，结果如下：**



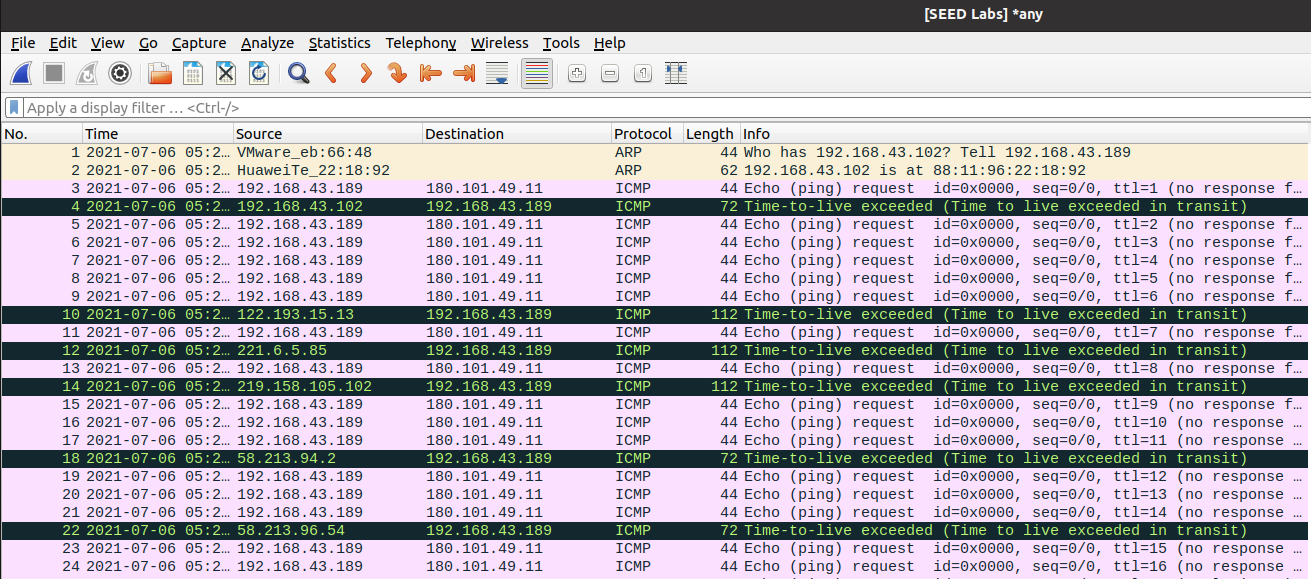
**Task 1.3 Traceroute**

**编写python脚本traceroute.py**

**功能为重复向目标ip发送三十次报文，TTL逐渐增加,以此来测试路径**

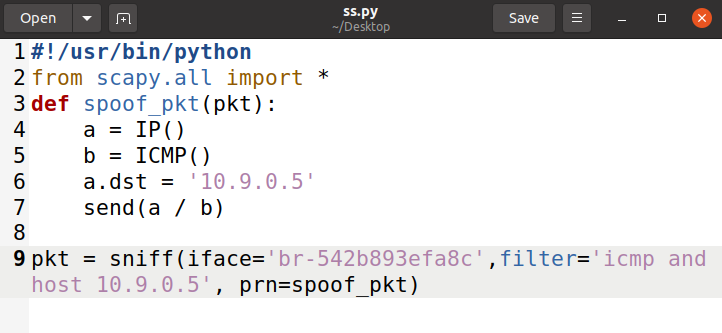


**使用wireshark抓包，查看结果，可以看到加黑部分为丢包后，路由返回的ICMP错误消息，由此可以看出源节点192.168.43.189与目标ip 180.101.49.11之间有着6个中间路由**

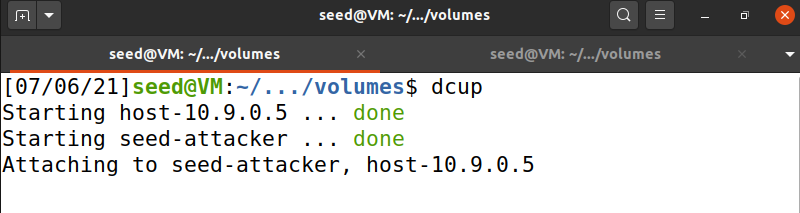


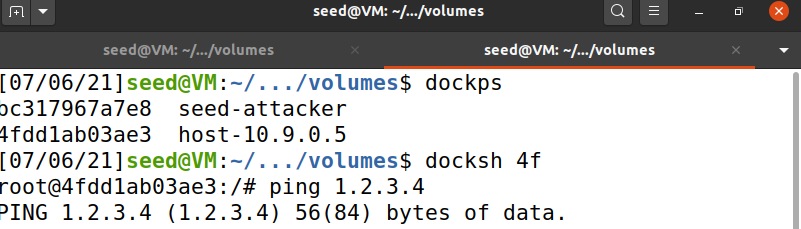
**Task 1.4 Sniffing and-then Spoofing**

**在该实验中，我们需要使用两台VM，A和B。让B去ping一个IP，这个IP是否存在不重要，然后在A中运行嗅探程序，监控LAN中的通信状态。理论上，只要B发出了ICMP包，A就会感知，并且发出伪造的响应。**



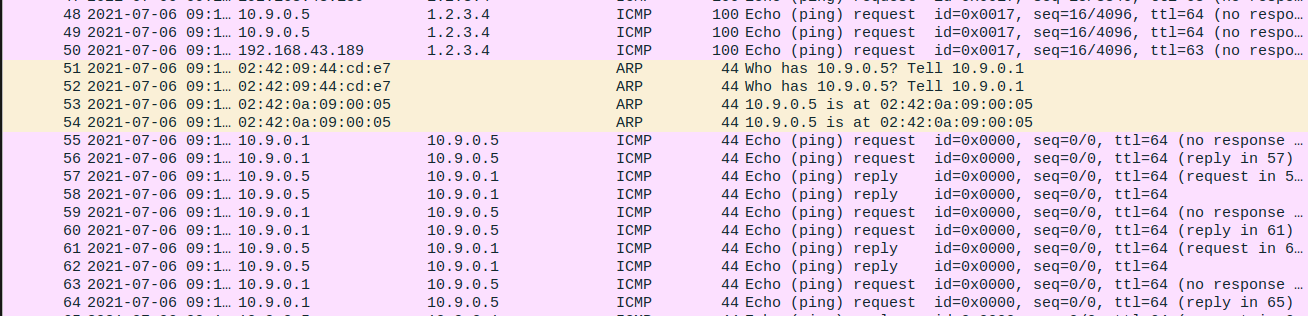
**首先需要进入host，在docker中打开主机**



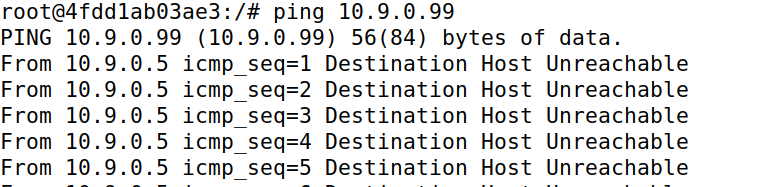


**Ping 1.2.3.4#a non-existing host on the Internet**

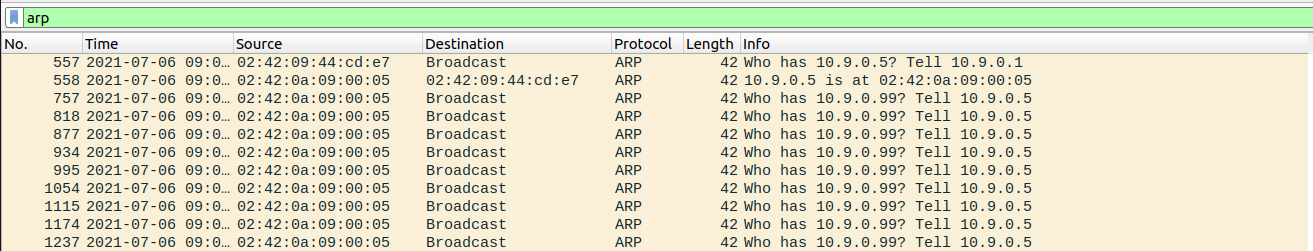
**可以看到对于互联网上不存在的主机，先用MAC地址进行广播，告知目标和本机对应的IP地址，然后劫持了报文，出现报文欺骗**



**Ping 10.9.0.99#a non-existing host on the LAN**



**可以看到报文一直在用MAC地址，利用广播的方式询问，但是一直得不到响应**



**Ping 8.8.8.8#an existing host on the Internet**

**可以看到主机对8.8.8.8进行访问，后来被劫持，10.9.0.1对10.9.0.5的请求进行响应**

