**第一次实验报告**

57118236 袁金傲

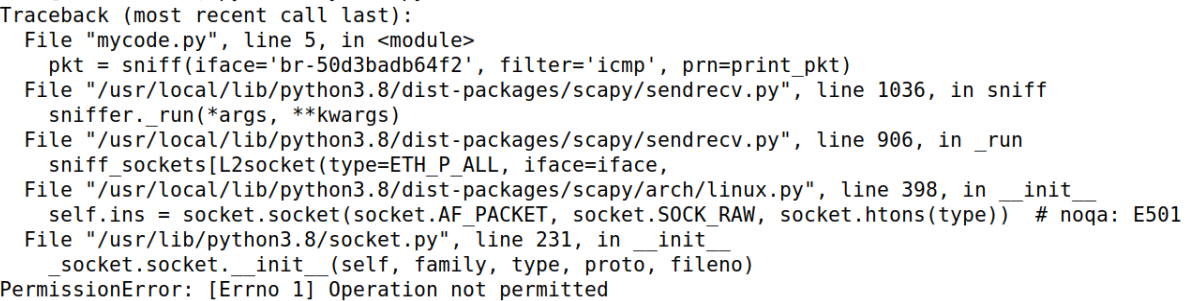
1. Task1.1

Task1.1A

在使用root权限后运行sniffer程序后输出如下：



而在使用su seed（退出root权限后）再运行sniffer程序，会出现如下的错误：

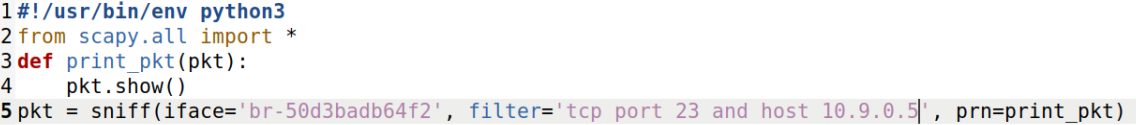


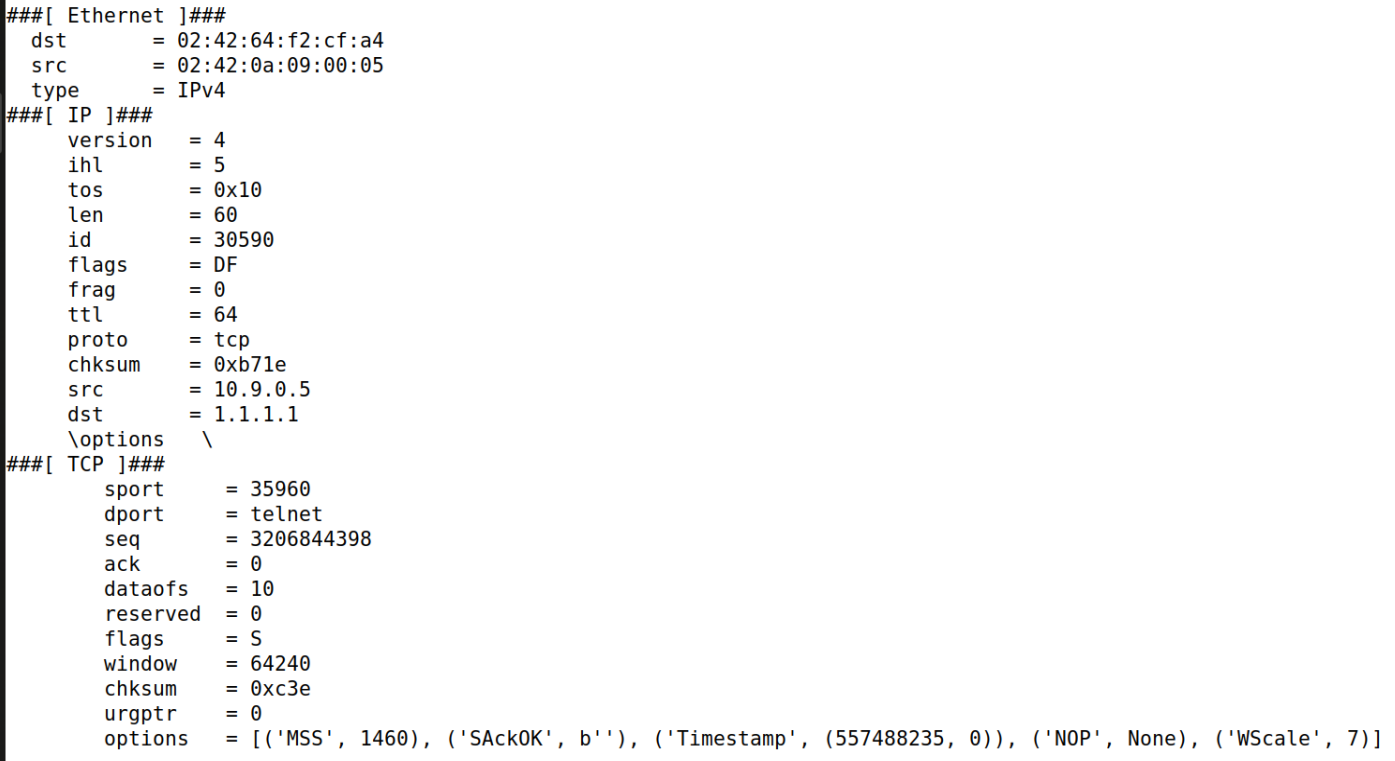
由此可见，sniffer程序仅可在root权限下才能正常运行，否则会因为socker问题而报错。

Task1.1B

1. 在仅捕获ICMP报文时，代码与上文相同，故输出也与上文相同。
2. 在捕获特定IP特定端口时，需要进行如下操作：

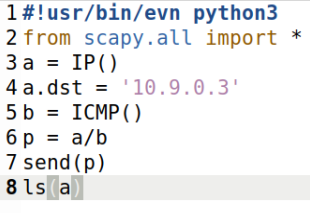
由于要监听特定端口，故需要将filter改为tcp port 23 and host 10.9.0.5



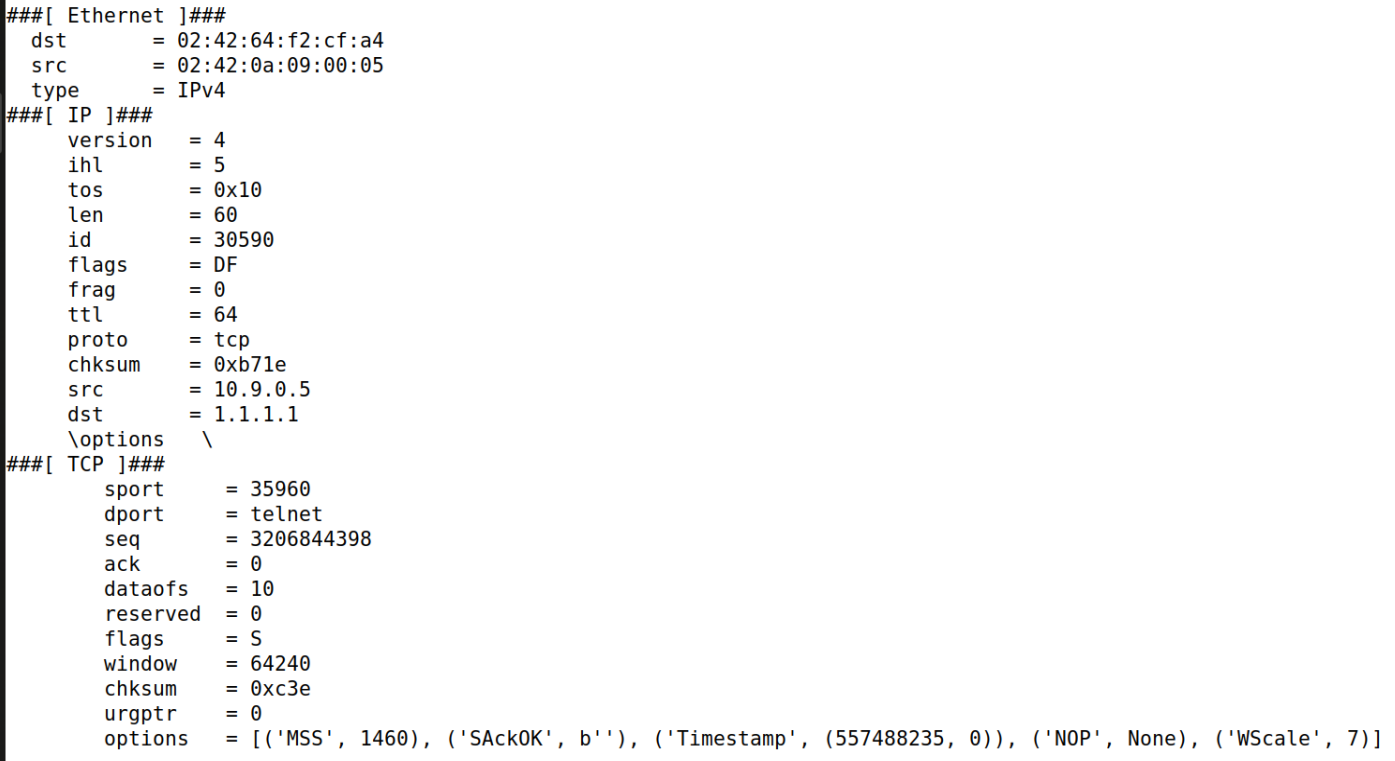
之后可以任意对一个特定的IP地址进行telnet连接并进行监听，输出为：  


二、Task1.2

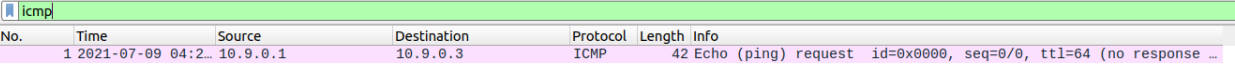
Spoofing.py代码如下：



在VM中运行spoofing.py可以得到如下输出：

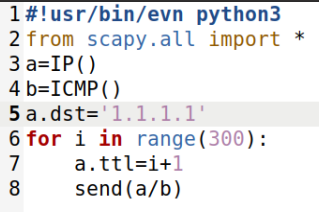


同时在wireshark进行监听，可以得到：



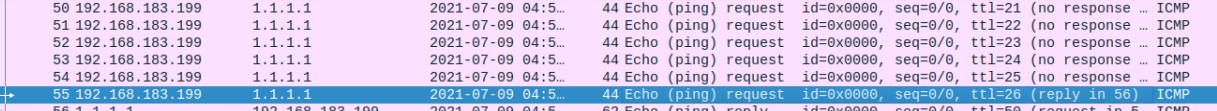
1. Task1.3

Traceroute.py代码如下：



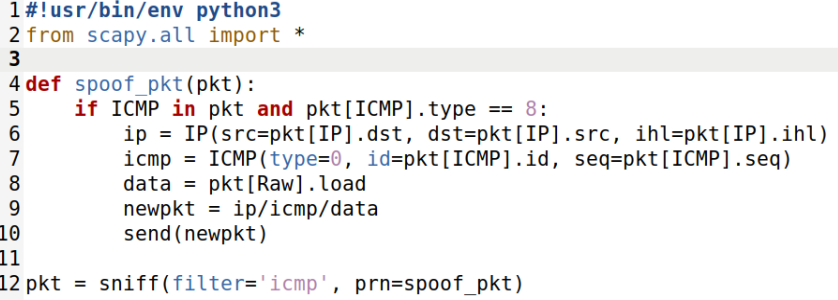
通过发送300个数据包（虽然上限只有255个数据包）并进行ttl递增，在ttl=26时有reply

故约进行了26跳。

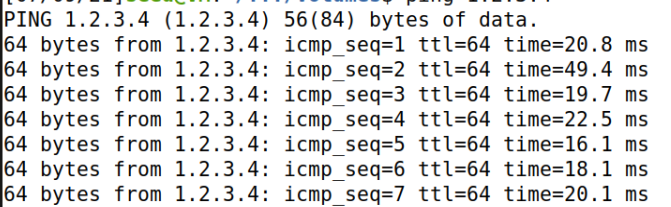


1. Task1.4

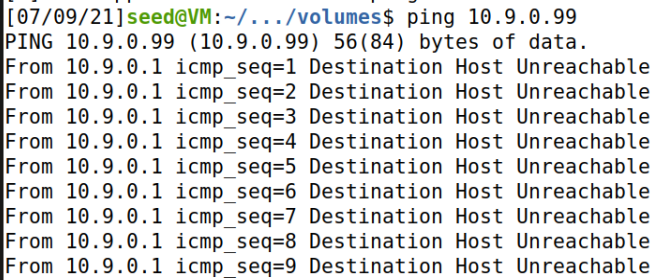
代码如下：



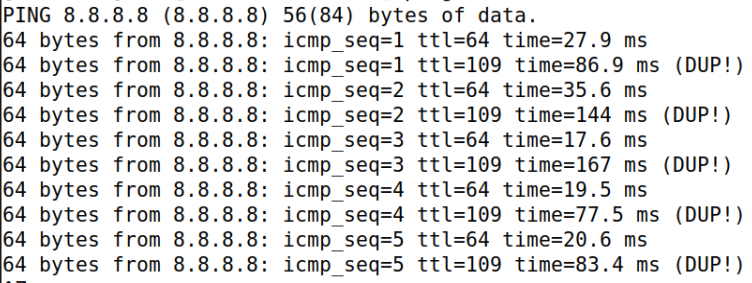
Ping1.2.3.4：



Ping10.9.0.99：



Ping8.8.8.8：



在ping1.2.3.4时，由于该地址需要向外部发送报文而被攻击者所监听，虽然该地址不存在，但攻击者仍会向主机发送正常的报文；

在ping10.9.0.99时，并不会经过攻击者的地址，故攻击者无法探测到ICMP报文，也就无从伪装；

在ping8.8.8.8时，会出现重复的报文，由于我们知道ttl=64，故也可以分辨真假报文。