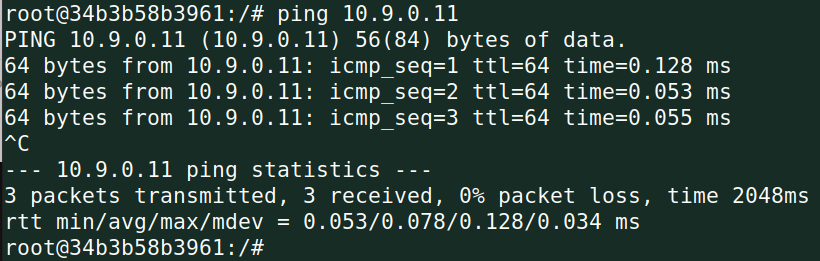
**VPN Lab**

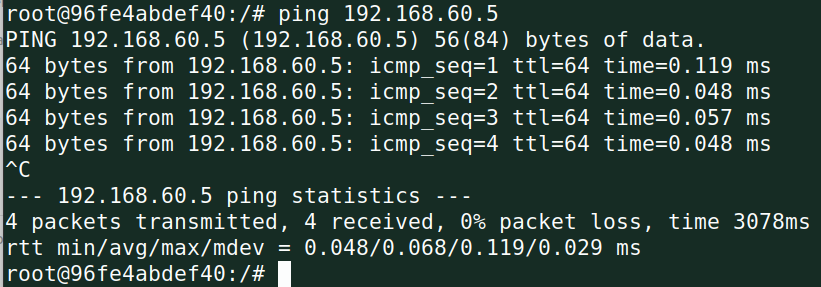
**57118215程文泽**

**Task 1: Network Setup**

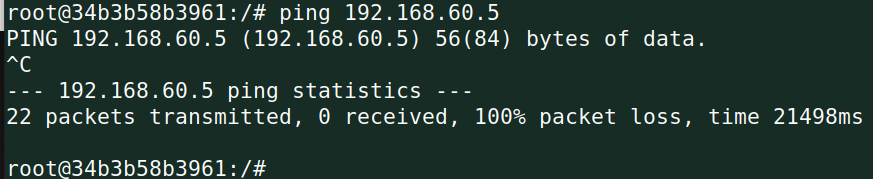
• Host U can communicate with VPN Server



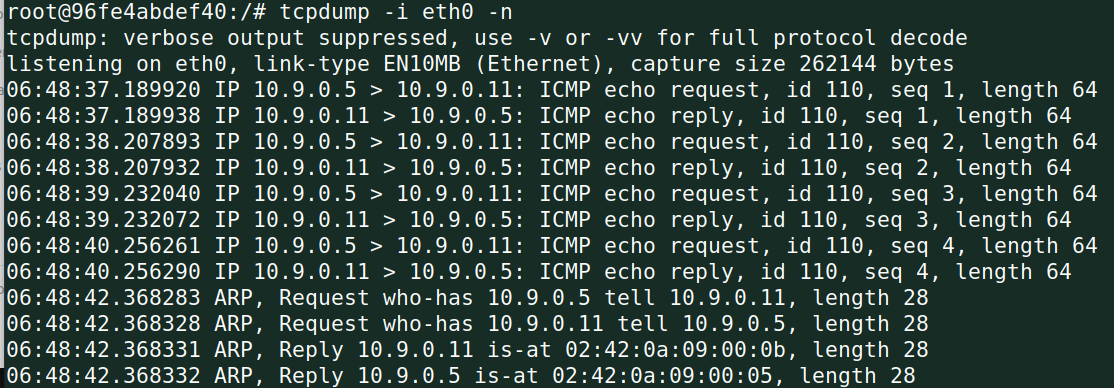
• VPN Server can communicate with Host V



• Host U should not be able to communicate with Host V.



• Run tcpdump on the router

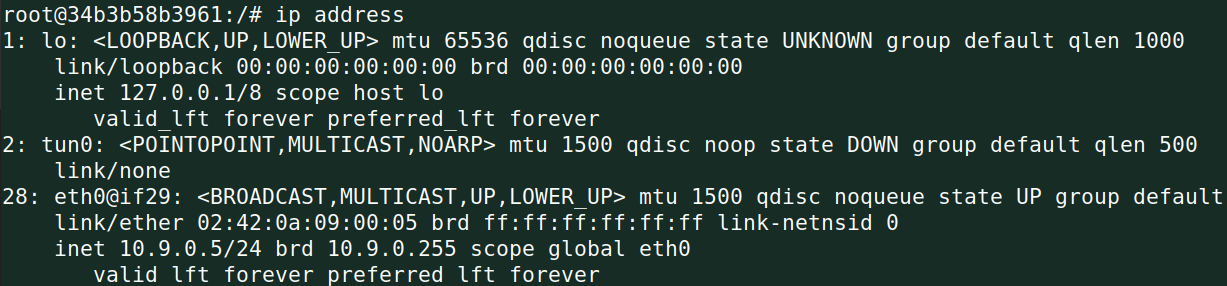


**Task 2: Create and Configure TUN Interface**

**Task 2.a: Name of the Interface**

在主机U上运行tun.py脚本，如下，随后查看主机上的所有接口，发现一个叫tun0的接口被创建。





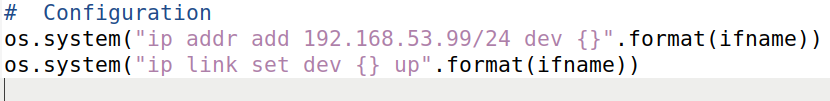
在脚本中修改接口名称为seed，运行脚本，查看主机U接口，结果如下：

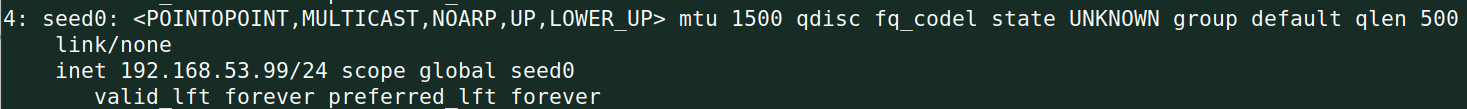




**Task 2.b: Set up the TUN Interface**

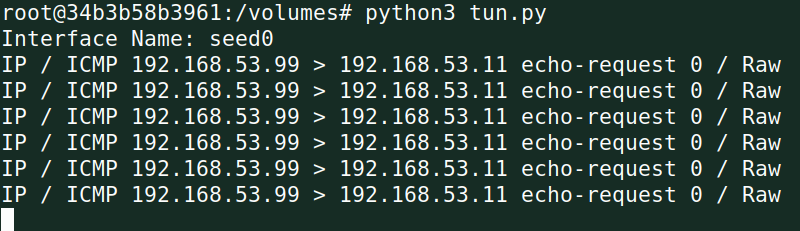
将配置代码加入脚本中，然后运行脚本，查看主机U接口，结果如下，可以看到接口被打开，且被配置了具体的IP地址

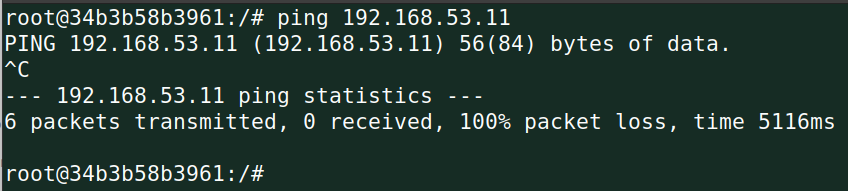




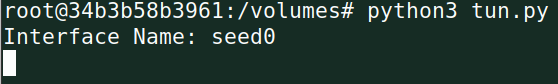
**Task 2.c: Read from the TUN Interface**

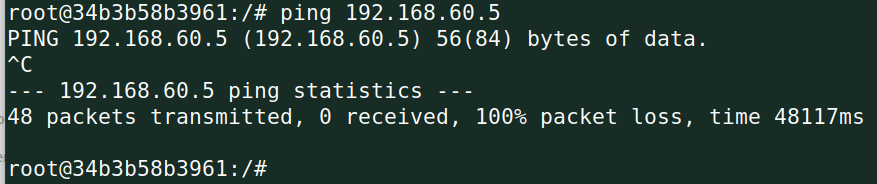
修改程序后，在主机U上ping 192.168.53.11，tun.py脚本输出了发出的ICMP报文的信息，如下图，我觉得这是因为seed0接口IP地址为目的地址同网段的所以会通过seed0接口发出ICMP请求报文，但该网段没有目的主机，所以无法收到回复报文。





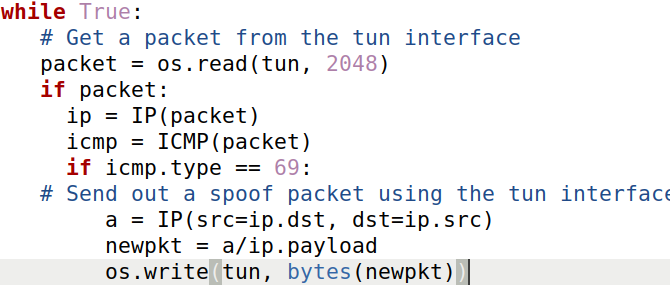
在主机U上ping 192.168.60.5，tun.py脚本无输出，如下图，这是因为目的地址与seed0接口IP地址不同且无相应的路由，所以ICMP请求报文不会从该接口发出，因此不会打印任何信息。

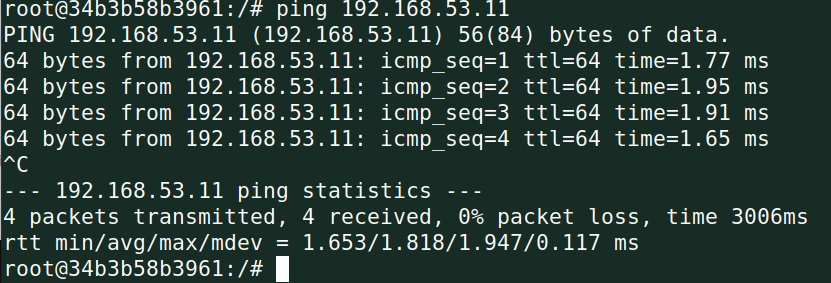




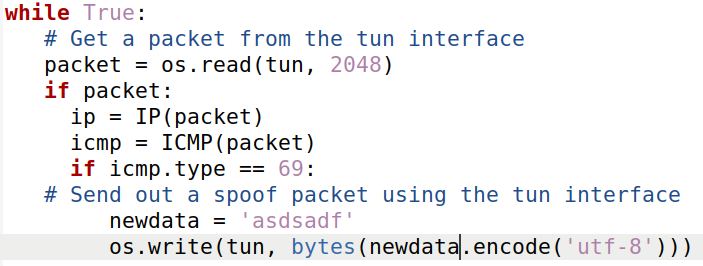
**Task 2.d: Write to the TUN Interface**

修改脚本如下，随后在主机U上ping 192.168.53.11,可以看到收到了来自192.168.53.11的回复。



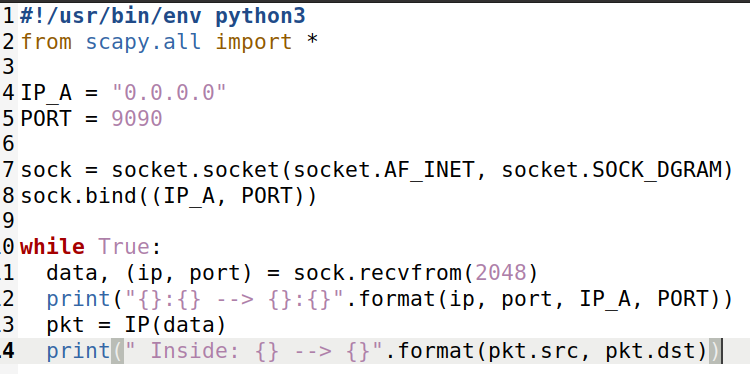


再次修改脚本如下，随后在主机U上ping 192.168.53.11，无任何反应，

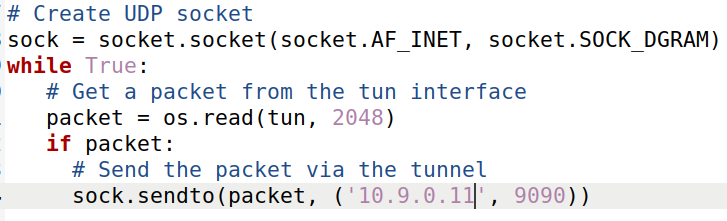


**Task 3: Send the IP Packet to VPN Server Through a Tunnel**

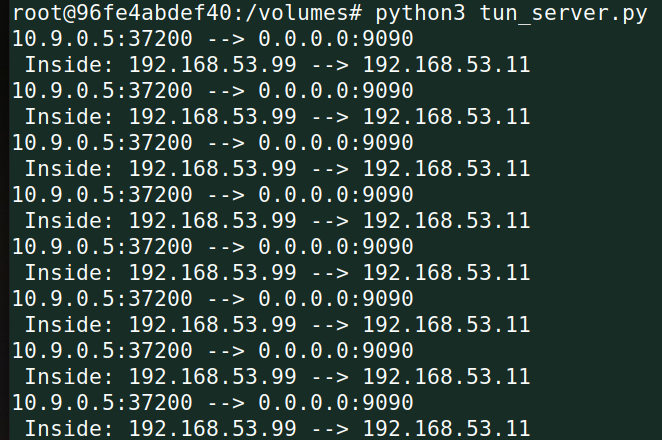
tun server.py脚本如下**：**



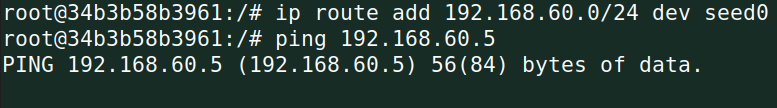
tun client.py脚本修改如下：

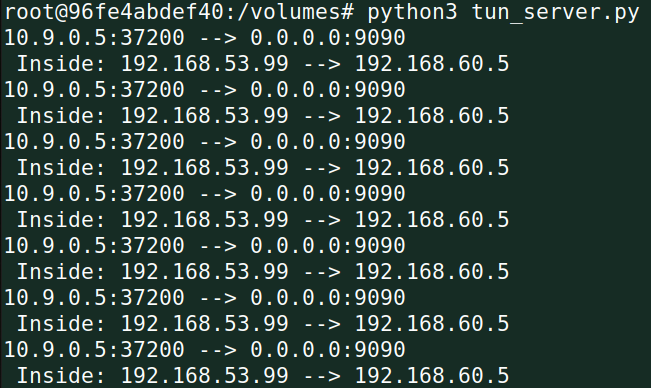


在VPN服务器上运行tun\_server.py程序，然后在主机U上运行tun\_client.py，随后在主机U上ping 192.168.53.11，可以在VPN服务器上看到以下结果，这是因为VPN服务器收到了来自主机U发来的UDP报文，随后通过tun\_serve1.py解析UDP报文，打印出相关信息。



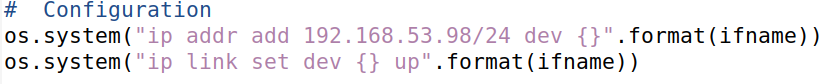
尝试ping 192.168.60.5，在VPN服务器上无任何显示，随后在主机U上添加一个路由如下，随后再次ping 192.168.60.5，可以在VPN服务器上看到相关信息

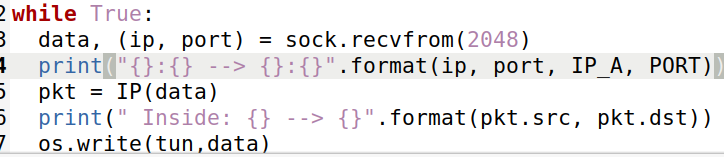




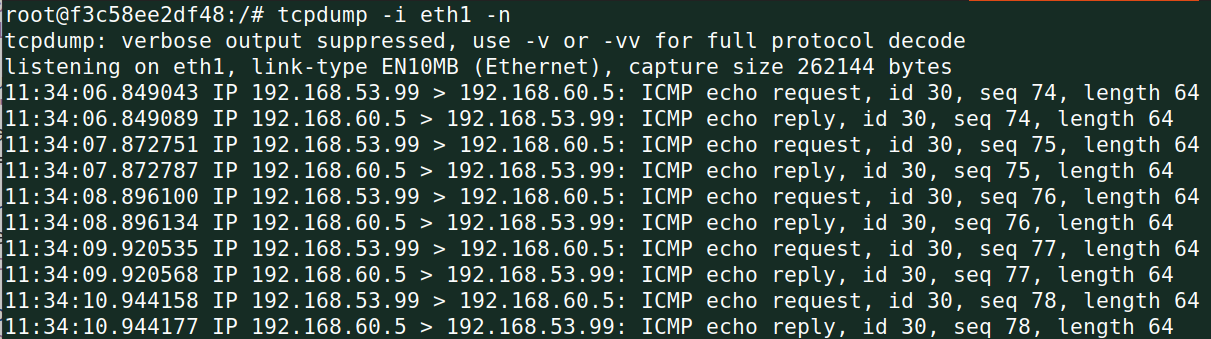
**Task 4: Set Up the VPN Server**

对tun\_server.py脚本进行修改，主要部分修改如下：



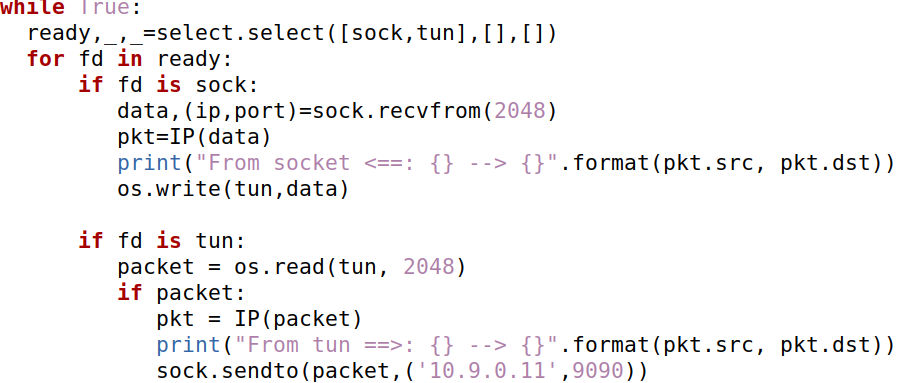


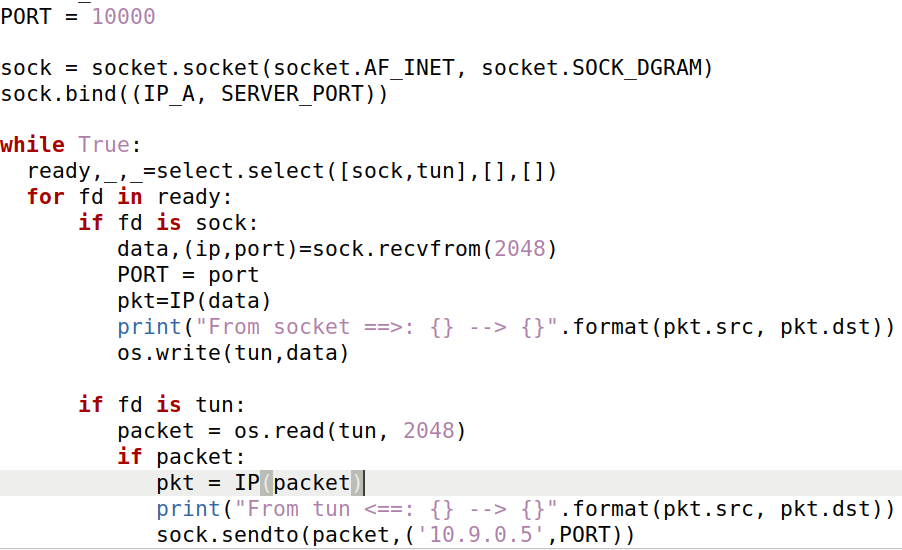
在VPN服务器上运行tun\_server.py程序，然后在主机U上运行tun\_client.py，同时在VPN服务器上开启tcpdump嗅探报文，随后在主机U上ping 192.168.53.11，可以嗅探发往192.168.60.5的ICMP请求报文，如下图：



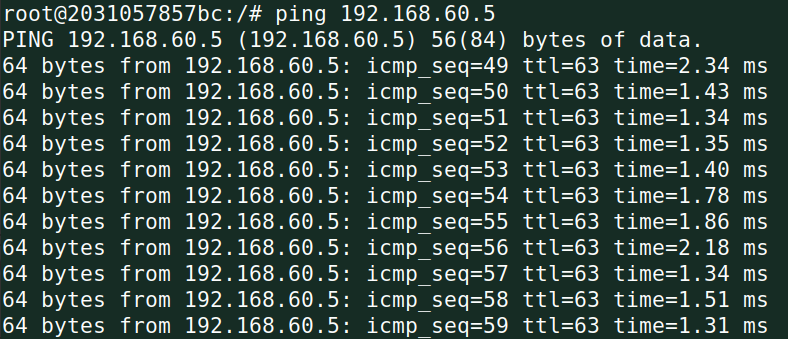
**Task 5: Handling Traffic in Both Directions**

修改tun\_client.py和tun\_server.py脚本如下：

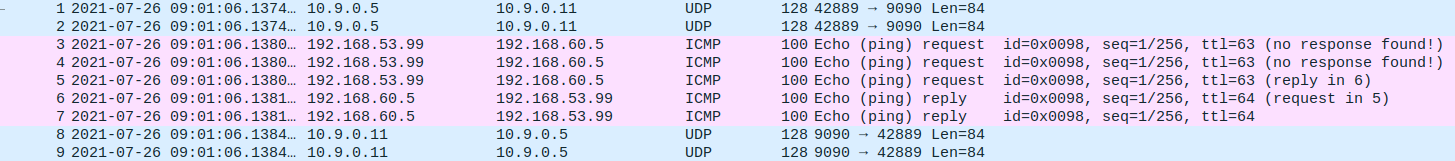




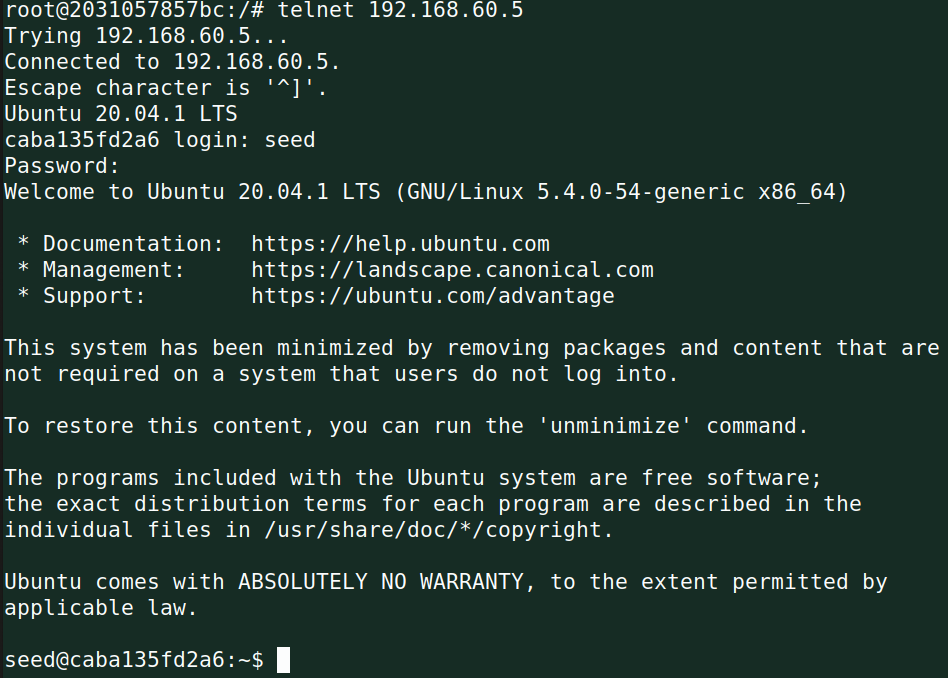
在主机U上ping主机V，结果如下：

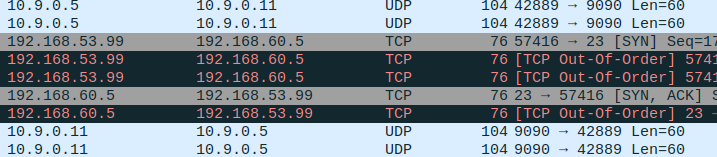


使用wireshark抓包如下，可以看到VPN客户端先将ICMP请求报文包装到UDP报文发给VPN服务器，随后VPN服务器解析UDP报文，将其中的ICMP请求报文发给目的地址即主机V，随后主机V发送ICMP回应报文，VPN服务器收到ICMP回应报文后，将其包装到UDP发给VPN客户端，最后VPN客户端解析UDP报文，收到了来自主机V的ICMP回应报文。



在主机U上telnet远程登入到主机V，同时抓包，结果如下，成功实现，





**Task 6: Tunnel-Breaking Experiment**

在主机U上telnet主机V，保持连接，随后停止tun\_client.py脚本，在telnet窗口进行输入，没有任何显示，结果如下：



然后再次开启关掉的脚本，之前输入的都会显示出来，这是因为停止脚本后进行输入，产生的tcp报文不能发往主机V，所以得不到来自主机V的回应，所以无法显示输入，再次开启脚本后，会立即建立隧道，而之前的TCP报文也因此可以通过隧道发给主机V，最后会重新在主机U上显示。

