LAB7

57117131 周煦彭

TASK1:

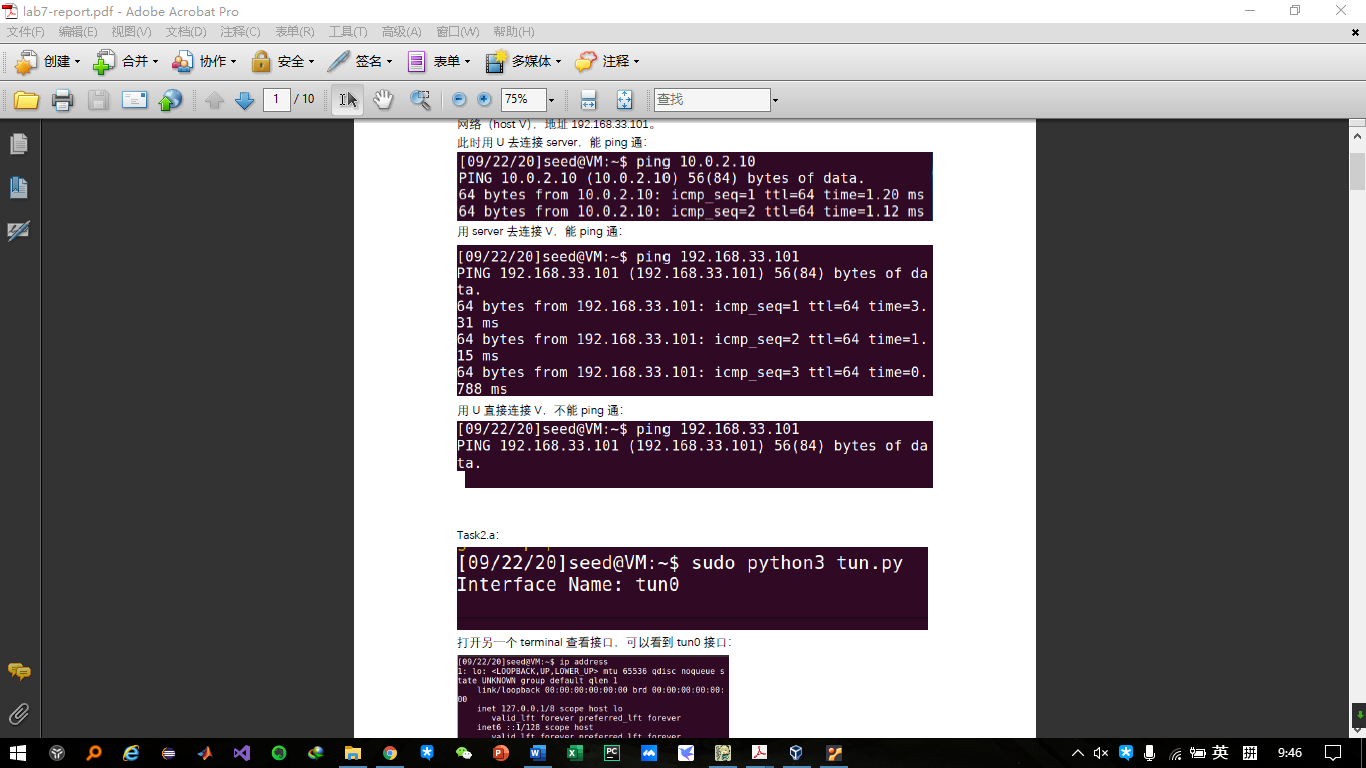
Host U:10.0.2.17

VPN server：10.0.2.10 192.168.33.4

HOST V：192.168.33.101

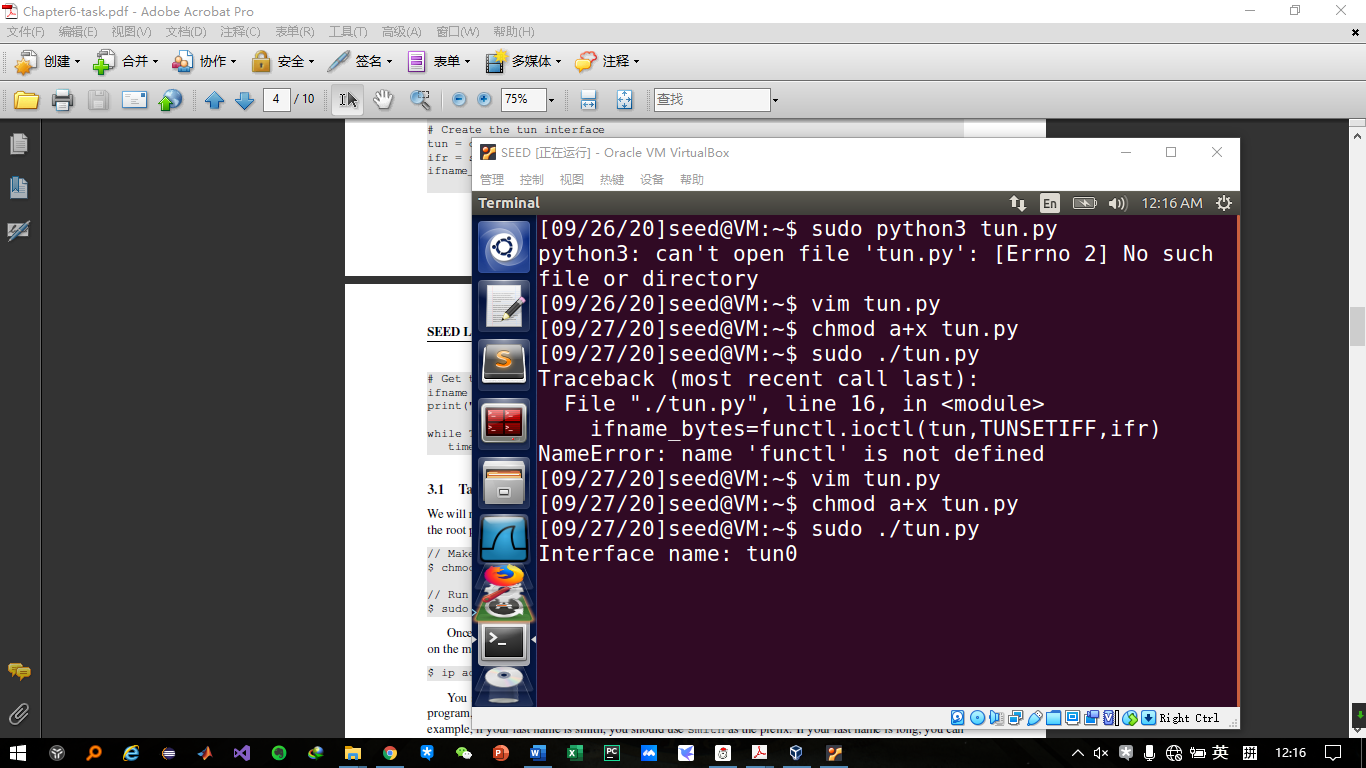
去ping server，发现ping成功

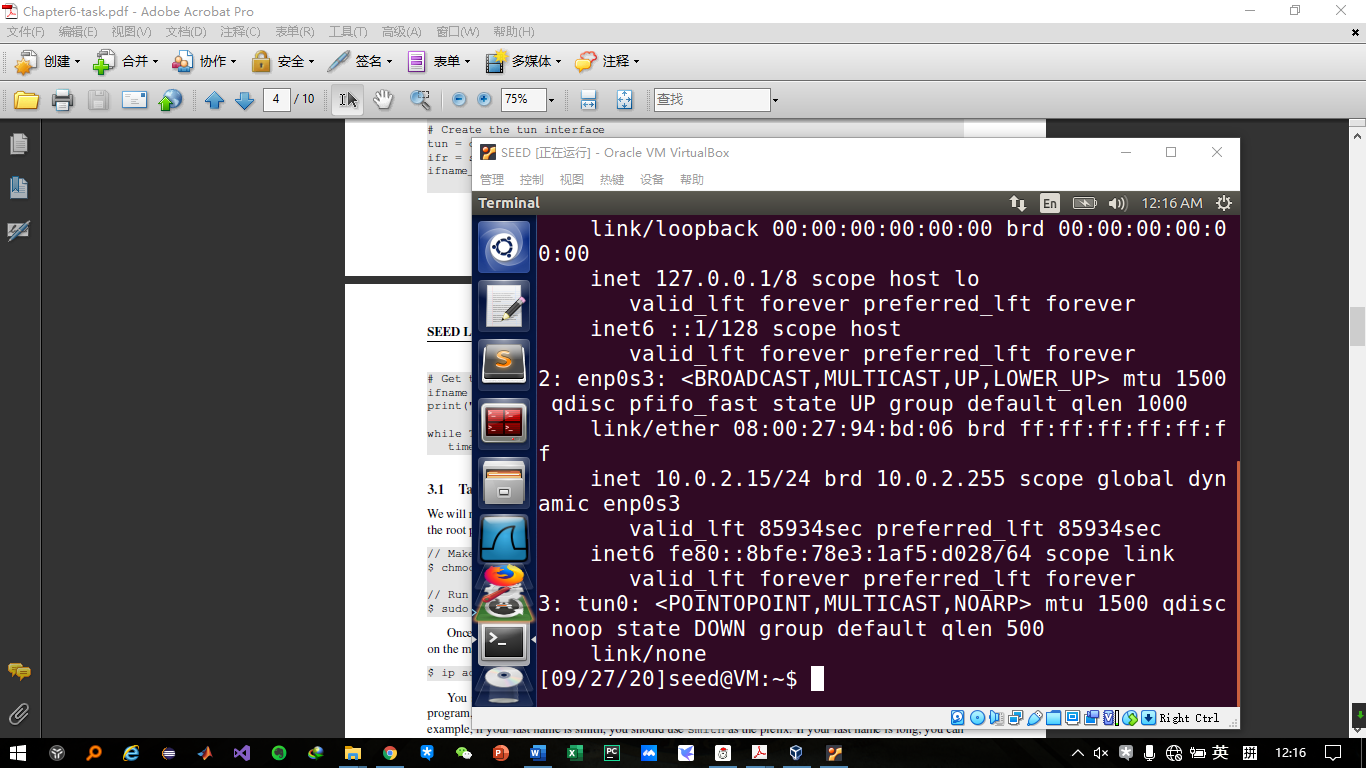
但是用U连接V后，ping结果失败



TEXT2：

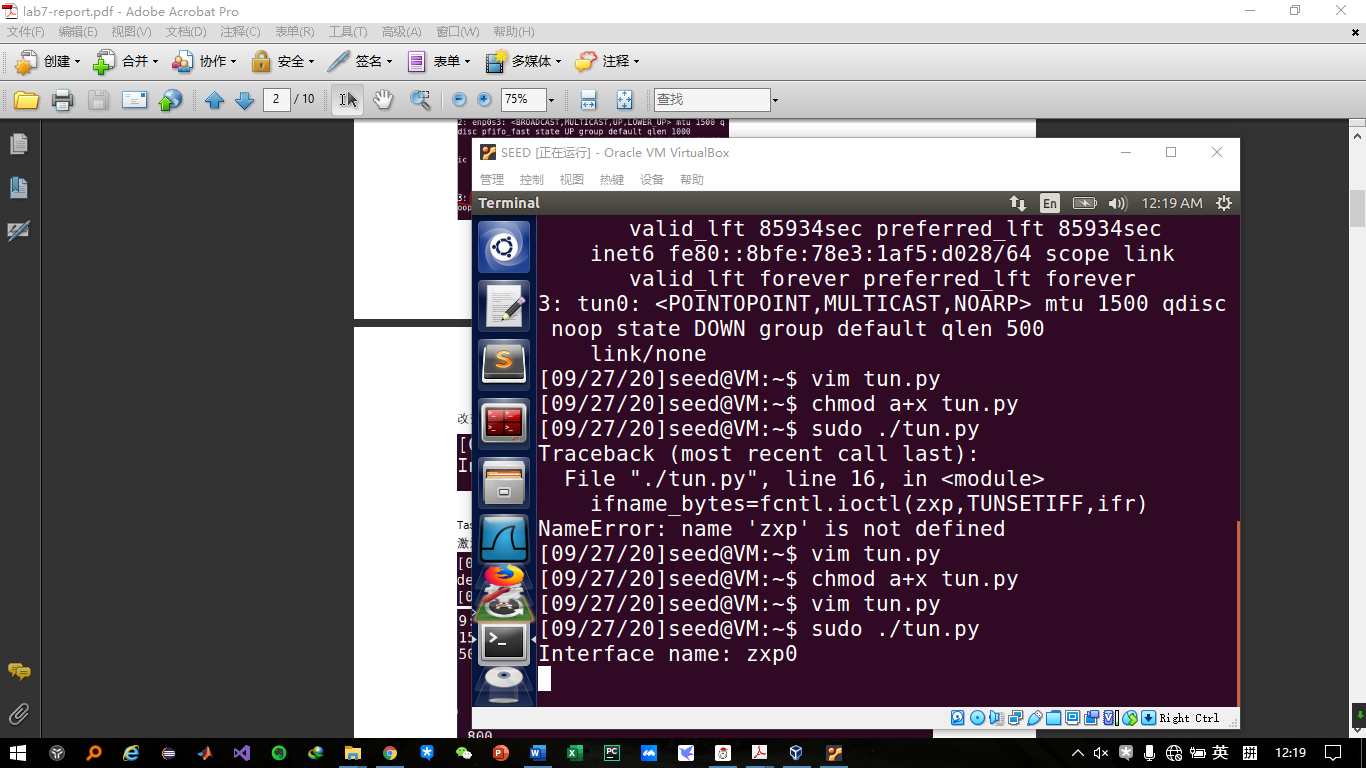
a.

创建一个TUN接口 ,tun.py

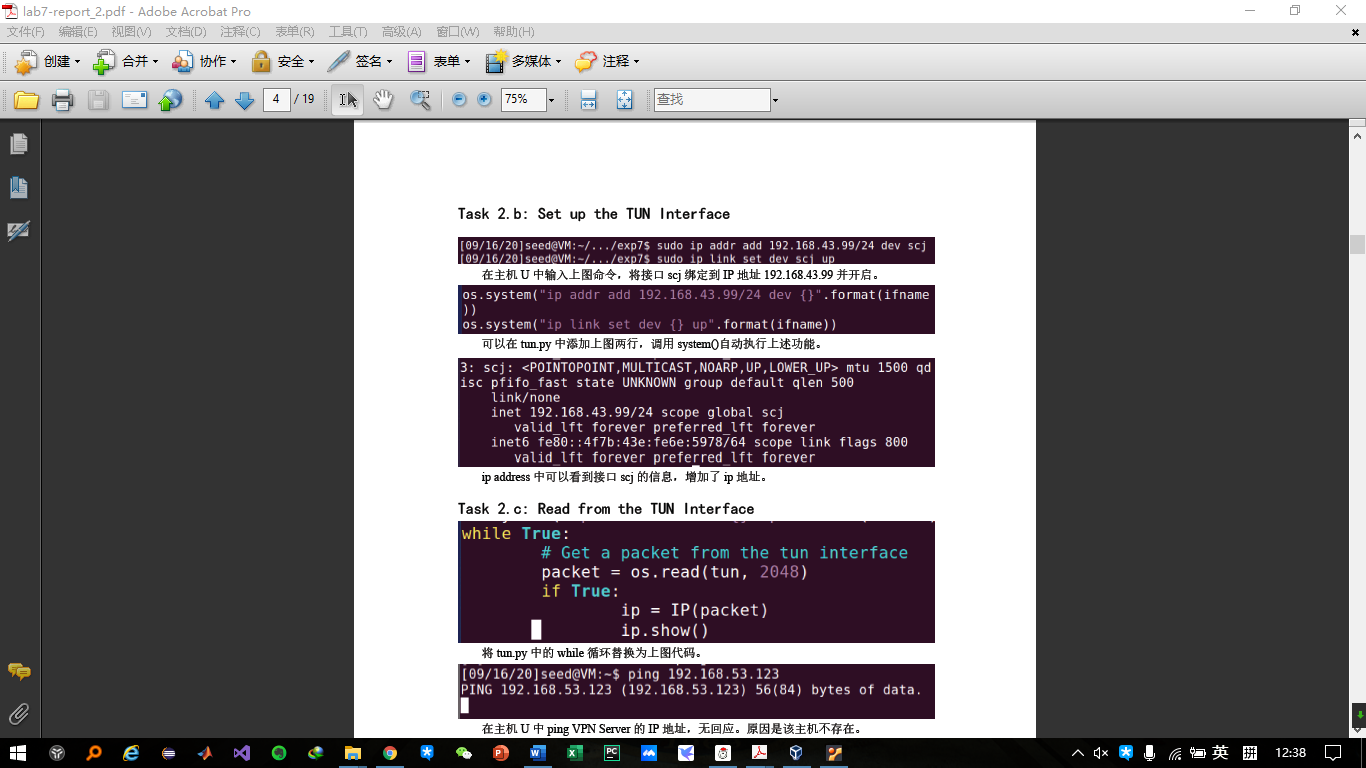
运行tun.py

打开终端接口，可以看到tun0接口

更改接口名为zxp0，可以看到接口修改成功

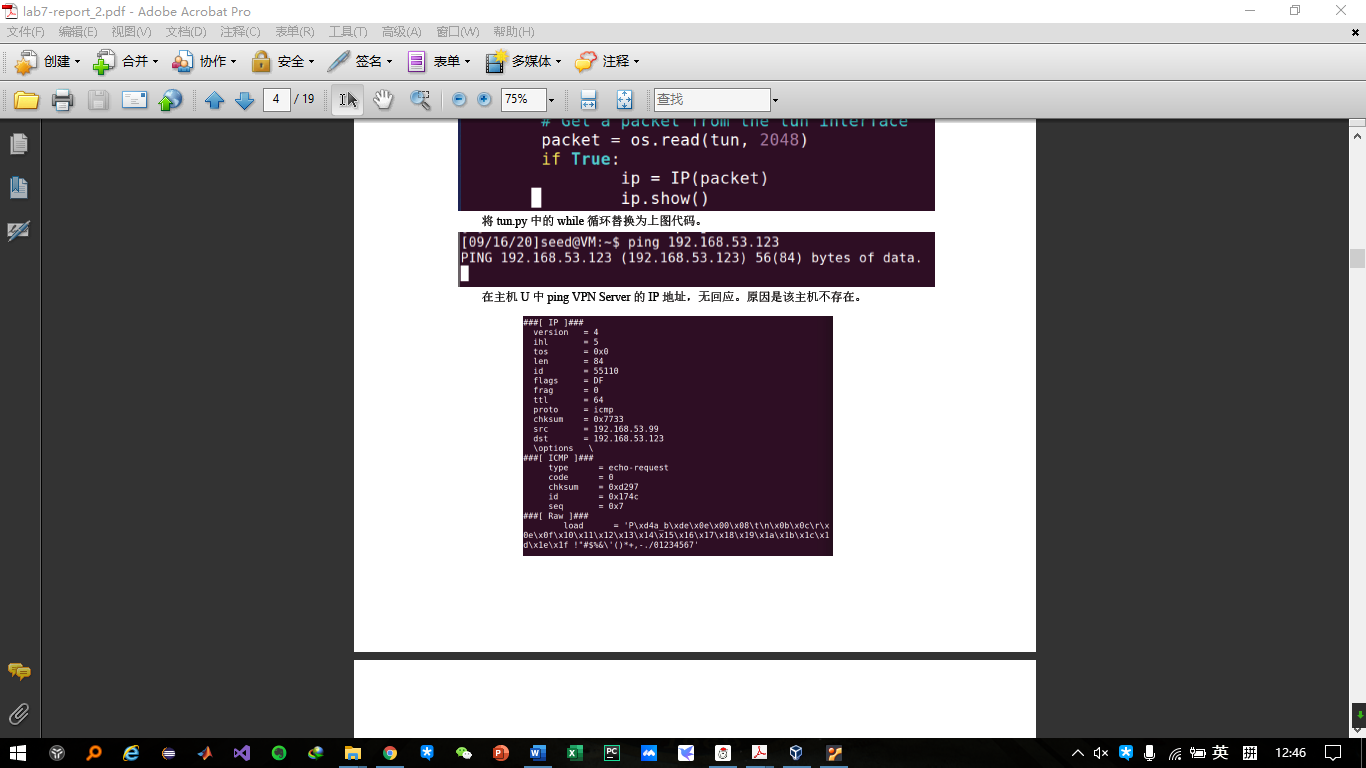


b.激活接口，发现接口已被绑定ip



C

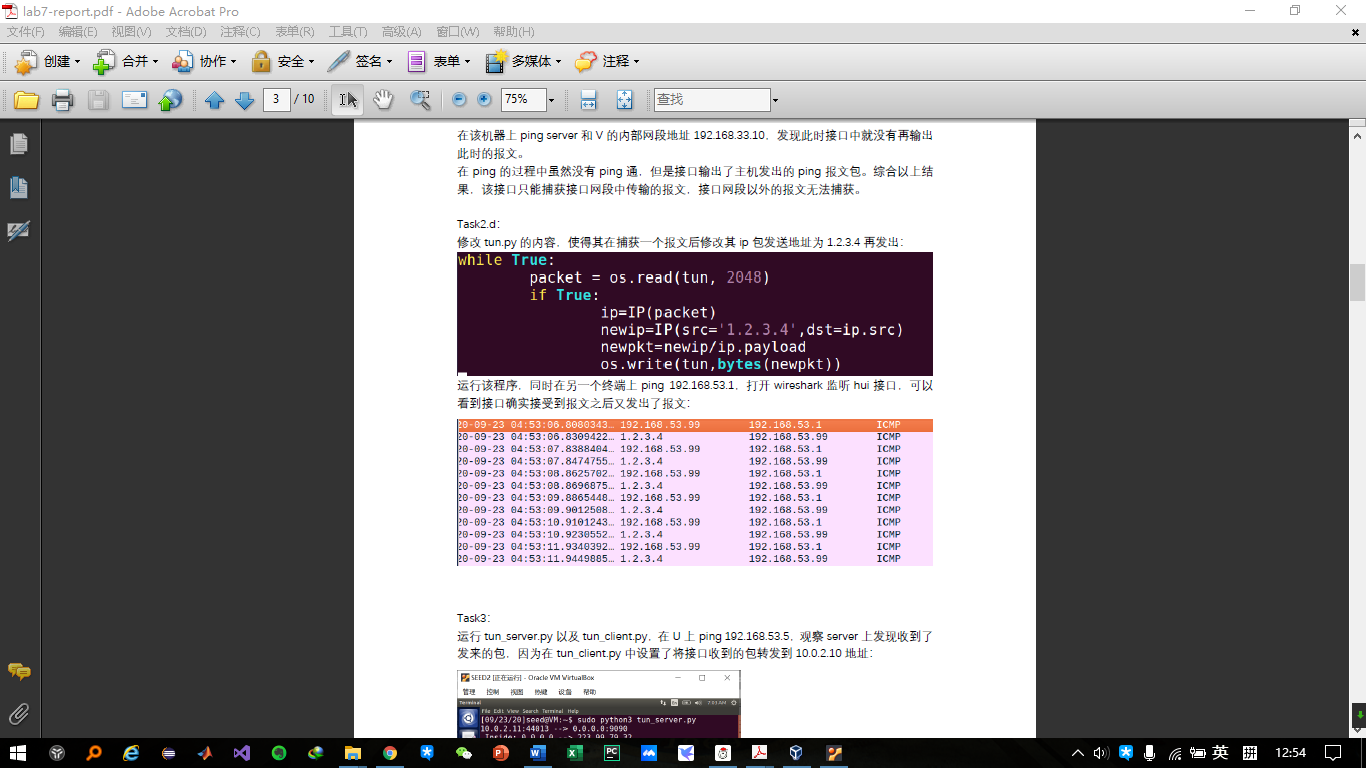
从该主机ping通192.168.33.4

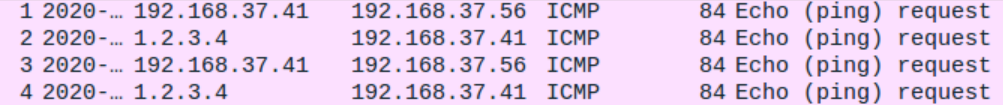


打印出ping报告

d.

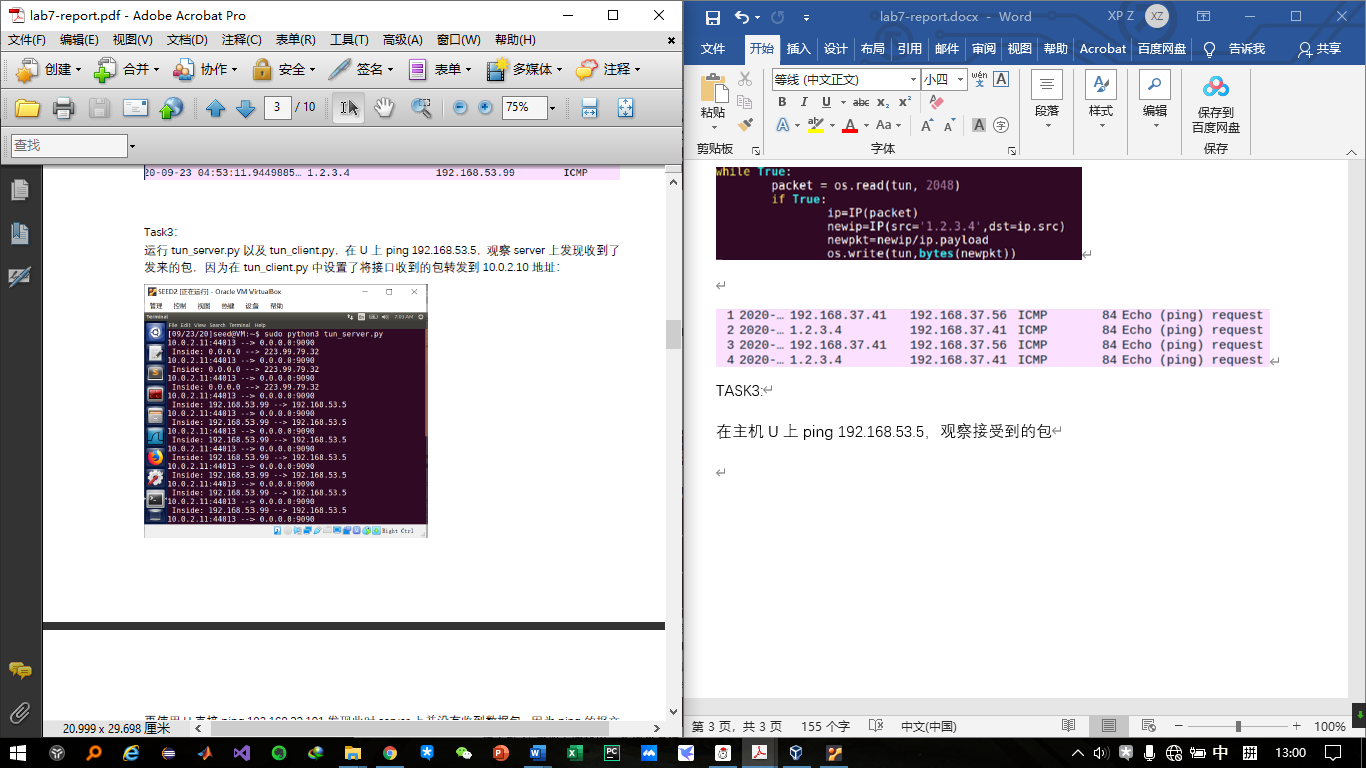
修改 tun.py的内容，使得其在捕获一个报文后并成功发出





TASK3:

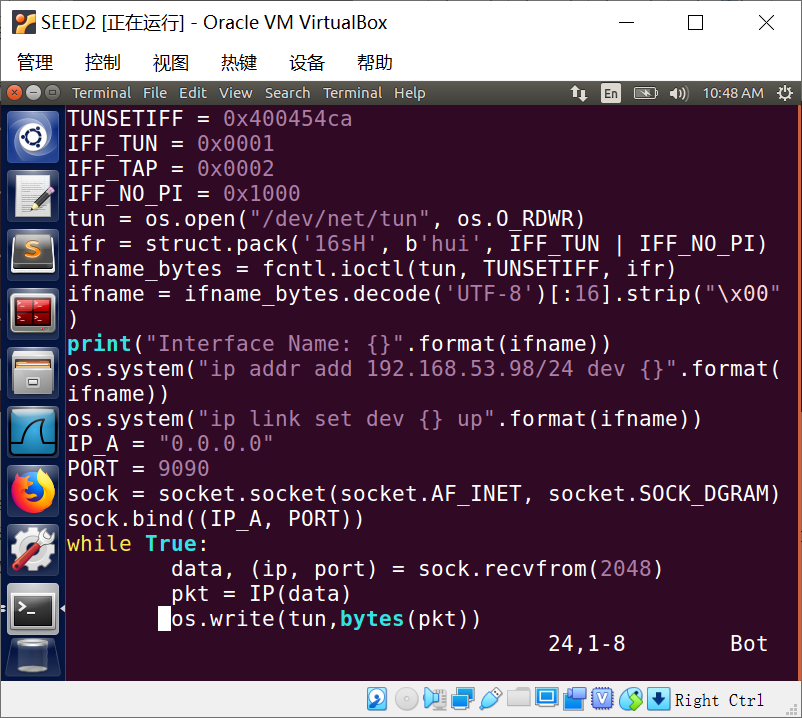
在主机U上ping 192.168.53.5，观察接受到的包



客户端的TUN接口捕获到ping报文后，将报文作为数据部分封装在新的IP头里，即源地址为10.0.2.4，宿地址为10.0.2.5，将新的报文发送到内核，内核通过路由设置选择合适的接口发送给服务器。服务器收到后，将IP头去除，取出原始报文，得到原始报文的源地址192.168.53.99，宿地址为192.168.53.5。服务器收到报文但未进行转发.

TASK4:

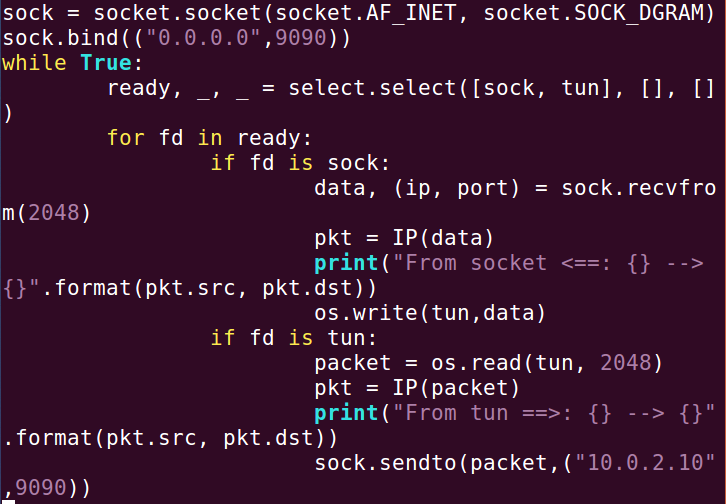
修改tun.py,绑定地址，将ping报文传入zxp0接口：

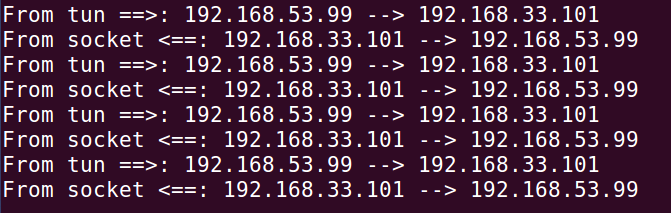


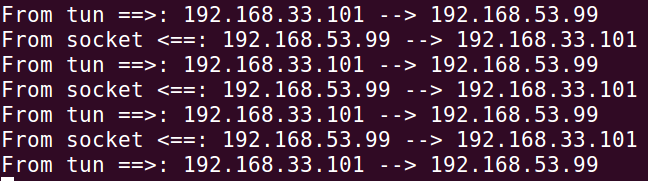
抓取icmp报文

TASK5:

修改tun.py片段

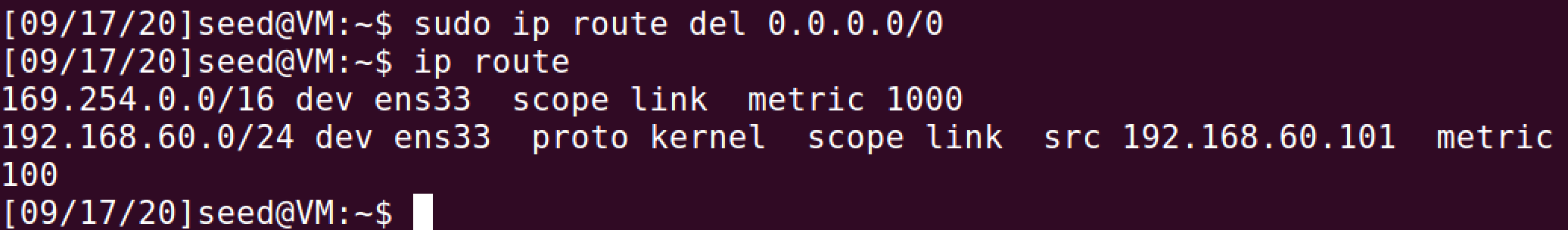


查看hostU上tun\_client的输出可以发现，ping报文通过tun接口发送给了出去，并且最终通过socket包取回了响应报文：

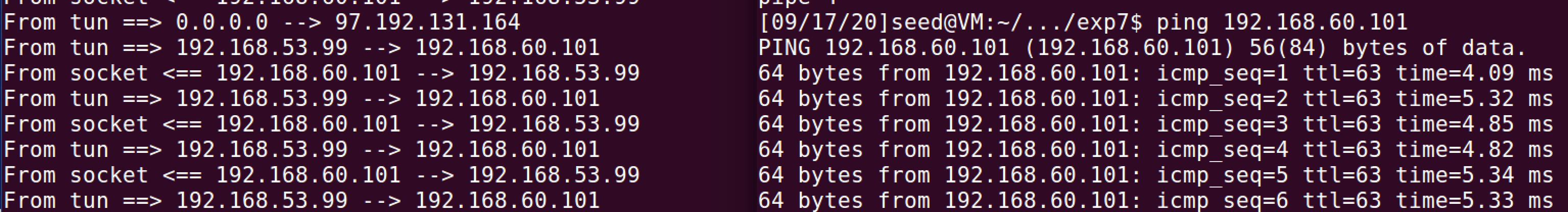
查看server上tun\_server的输出以及wireshark也能发现其做了U和V通信过程中转发的角色：

TASK6:

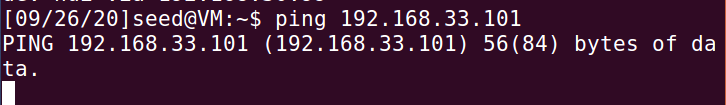
在telnet建立时两个机器经过了tcp三报文握手建立起连接，断开tun\_client后只是阻断了U和V交流的报文通道，实际上tcp还没有经过四报文握手还在等待中，只要时间不长tcp连接就不会断开。当恢复tun之后隧道又变得畅通两个机器可以继续交流。

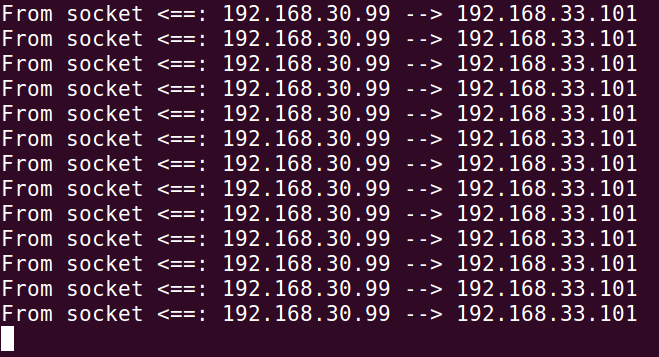
TASK7: 

清楚主机V默认路由

添加信息，显示ping通

TASK8:

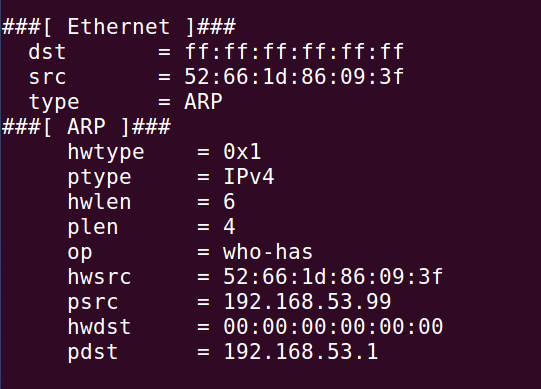
修改tun接口为192.168.30.99后，再次pingV的主机地址发现没有收到响应：



当Host U修改自己TUN接口的IP地址后，仍可以将报文封装到隧道中发送出去。但是报文传送在服务器端中断，服务器收到报文后不进行转发，从而使报文无法传递到Host V处。

内核反向查找路由发现该报文的源地址属于192.168.26.0/24这个网段，将其作为目的地址查找路由并没有找到该网段的接口，故将其丢弃。

TASK9:



接口tap0经过了许多ARP报文，。ARP报文到达协议栈二层时被TAP接口捕获，由于该网络是直连网络且是虚拟网络，该报文滞留在TAP接口上而没有被转发出去。