**5718133钟杰**

**Task2.A**

root@b481c133d961:/volumes# chmod a+x tun.py

root@b481c133d961:/volumes# tun.py

Interface Name: tun0

root@b481c133d961:~# ip address

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

5: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500

link/none

13: eth0@if14: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP group default

link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0

inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0

valid\_lft forever preferred\_lft forever

**Task2.B**

执行两个指令后查看ip address，如图，可以看到tun0有了ip地址

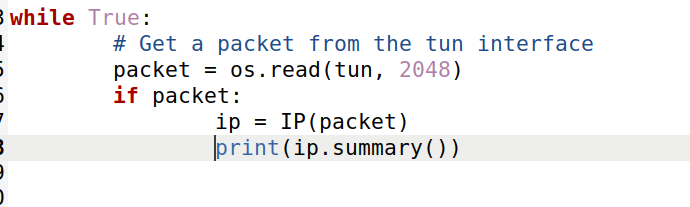
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UNKNOWN group default qlen 500

link/none

inet 192.168.53.99/24 scope global tun0

**Task2.C**

修改代码



重新执行tun.py，配置接口地址并开启接口，ping192.168.53.0/24网段，如图所示

root@b481c133d961:/# ping 192.168.53.1

PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.

^C

--- 192.168.53.1 ping statistics ---

6 packets transmitted, 0 received, 100% packet loss, time 5100ms

root@b481c133d961:/volumes# tun.py

Interface Name: tun0

IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw

IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw

IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw

IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw

IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw

IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw

^CTraceback (most recent call last):

File "./tun.py", line 27, in <module>

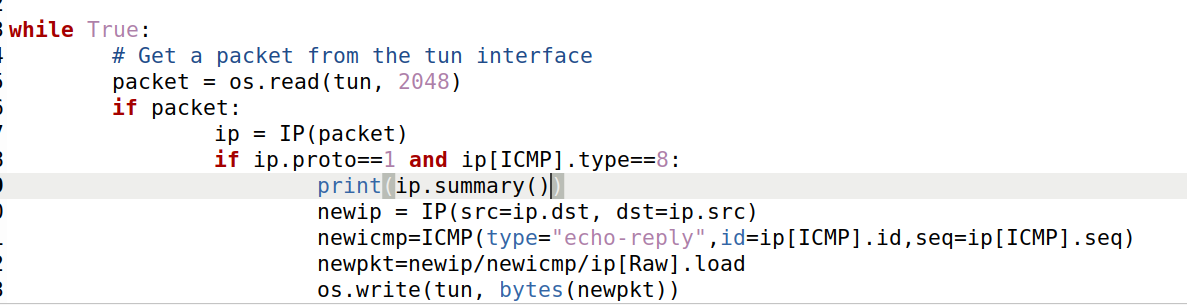
packet = os.read(tun, 2048)

KeyboardInterrupt

Ping192.168.60.0/24网段，无法ping通且没有输出，因为目前没有到达192.168.60.0/24网段的路由

Task2.D

代码如下



未执行tun.py时无法Ping通 192.168.53.1，执行后如图所示，能够ping通

root@b481c133d961:/# ping 192.168.53.1

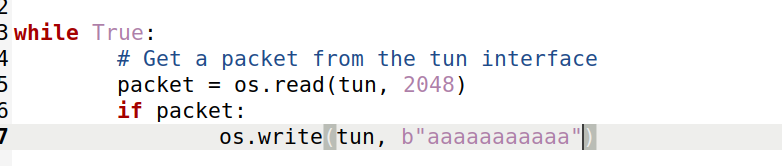
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.

64 bytes from 192.168.53.1: icmp\_seq=1 ttl=64 time=1.44 ms

64 bytes from 192.168.53.1: icmp\_seq=2 ttl=64 time=1.68 ms

64 bytes from 192.168.53.1: icmp\_seq=3 ttl=64 time=1.49 ms

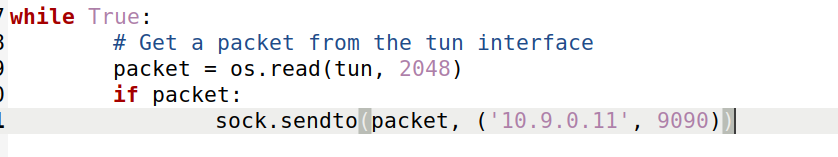
修改tun.py，如图所示，无法ping通，写入的任意字符串会被当做伪造的IP报文



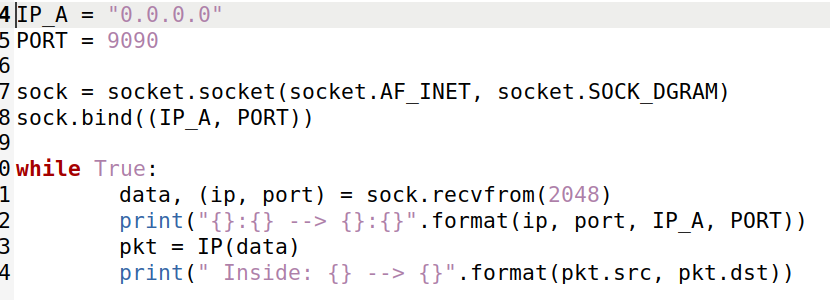
**task3**

代码如下

Tun\_client:



Tun\_server:



Ping 192.168.53.5

root@b481c133d961:/# ping 192.168.53.5

PING 192.168.53.5 (192.168.53.5) 56(84) bytes of data.

^Z

[3]+ Stopped ping 192.168.53.5

在vpn server上有如下输出

root@bb5bf6a5acc2:/volumes# python3 tunserver.py

10.9.0.5:50729 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.53.5

10.9.0.5:50729 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.53.5

10.9.0.5:50729 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.53.5

10.9.0.5:50729 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.53.5

10.9.0.5:50729 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.53.5

10.9.0.5:50729 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.53.5

10.9.0.5:50729 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.53.5

10.9.0.5:50729 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.53.5

10.9.0.5:50729 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.53.5

现在ping 192.168.60.0/24网段不通，因为没有路由

修改tun\_client.py，添加如下代码



现在ping 192.168.60.5，在vpn server上有如下输出

root@bb5bf6a5acc2:/volumes# python3 tunserver.py

10.9.0.5:40525 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.60.5

10.9.0.5:40525 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.60.5

10.9.0.5:40525 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.60.5

10.9.0.5:40525 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.60.5

10.9.0.5:40525 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.60.5

10.9.0.5:40525 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.60.5

10.9.0.5:40525 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.60.5

10.9.0.5:40525 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.60.5

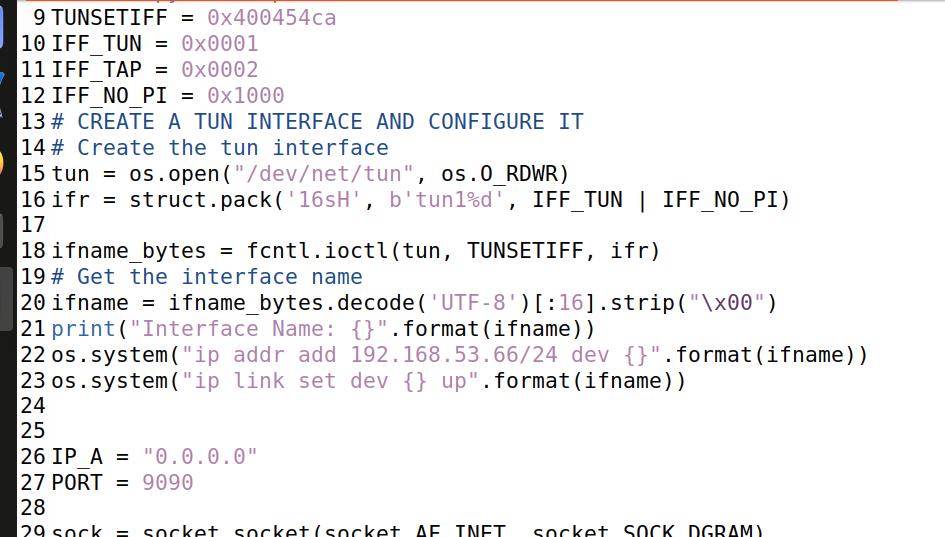
10.9.0.5:40525 --> 0.0.0.0:9090

Inside: 192.168.53.99 --> 192.168.60.5

**Task4**

程序如下

Client\_server:



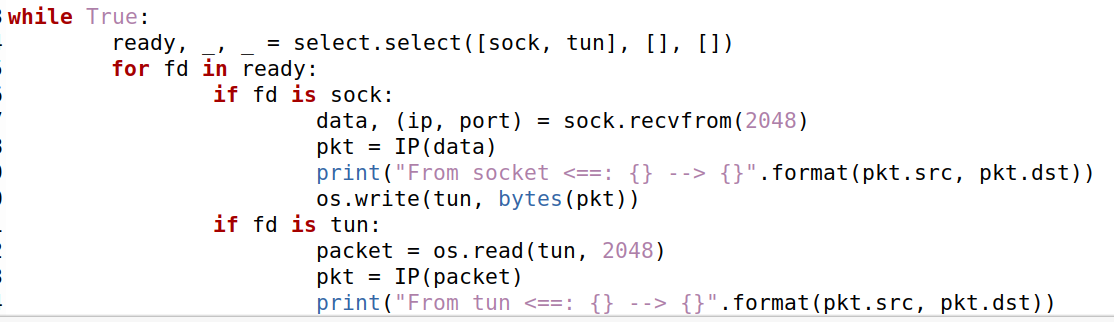
开启服务端和客户端后在客户端ping 192.168.60.5，查看wireshark

通过wireshark监听192.168.60.5，可以看到收到了icmp request并返回了reply，但并没有ping通，说明reply没有返回到10.9.0.5，因为没有配置好路由

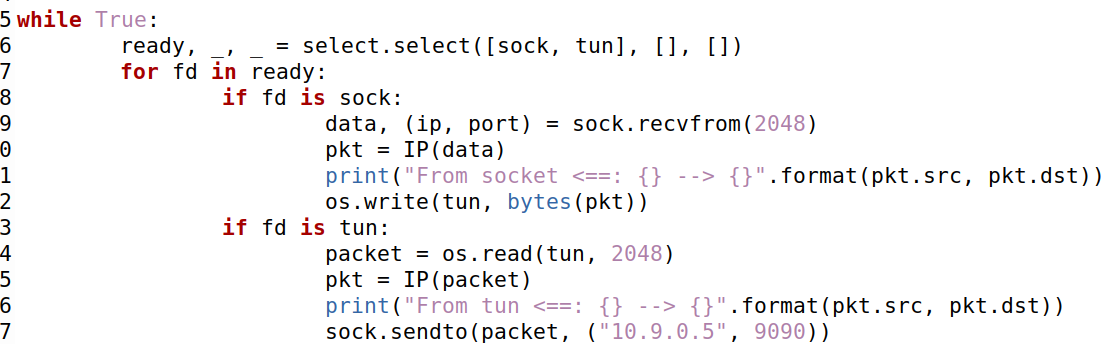
**Task5**

程序如下

Tun\_client



Tun\_server



10.9.0.5ping192.168.60.5,如图所示，ping通

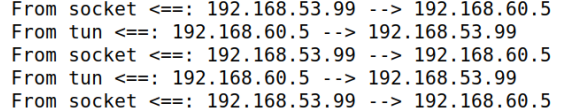
telnet连接成功

报文流向：

从10.9.0.5发向192.169.60.5的报文经过路由配置由网卡tun1发送到10.9.0.11，然后由于10.9.0.11开启了路由转发，报文被发送到192.168.60.5，应答报文发送到路由器后由于目的地址192.168.53.99与网卡tun10地址192.168.53.66处于同一网段，报文被转发到网卡tun10，再由tun10将其发送到10.9.0.5，由网卡tun0解析，整个过程结束。

**Task6**

Telnet连接后终止server，会无法输入内容，这时候重新启动server仍能显示刚才键入内容，

查看wireshark，发现tcp会持续重发包含第一个字符的报文，当该报文被收到后直接将后续所有键入内容放在一个报文中发送，如图所示

