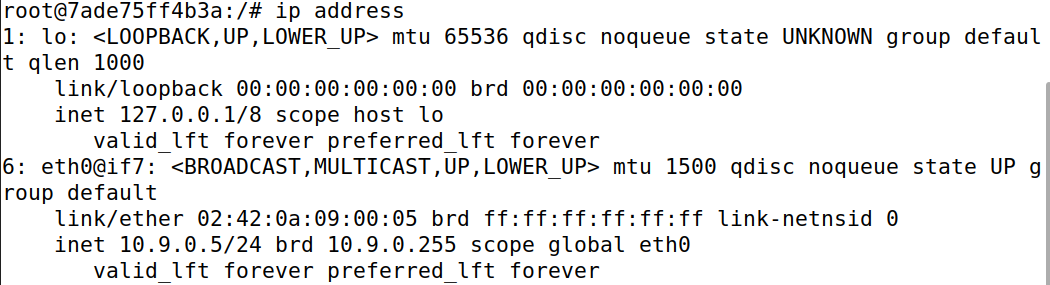
**第七次实验**

57118216 丰思飏

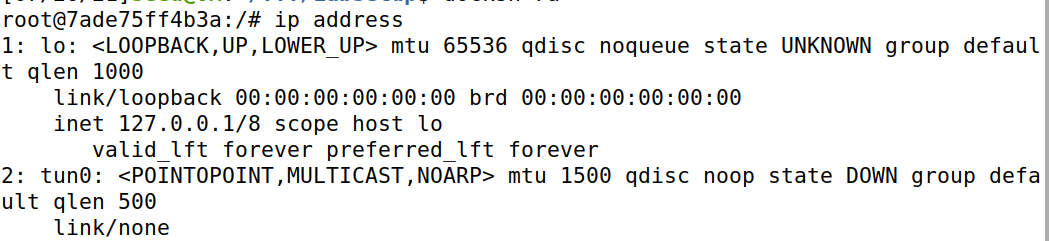
Task2

a

未开启端口前



开启端口后

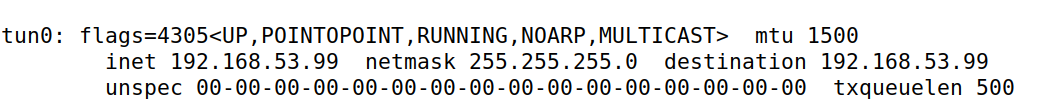


b

在程序中添加以下内容

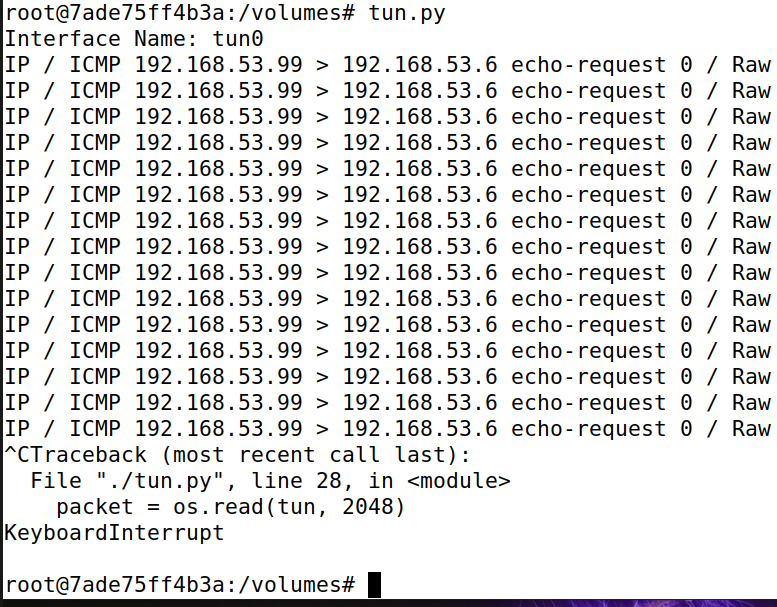
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))

os.system("ip link set dev {} up".format(ifname))



端口处于running状态，并显示地址

C



Ping192.168.53.6，程序输出ICMP请求信息，因为在循环中输出报文信息

Ping192.168.60.6，程序不输出，因为该子网无法连接

#!/usr/bin/env python3

import fcntl

import struct

import os

import time

from scapy.all import \*

TUNSETIFF = 0x400454ca

IFF\_TUN = 0x0001

IFF\_TAP = 0x0002

IFF\_NO\_PI = 0x1000

# Create the tun interface

tun = os.open("/dev/net/tun", os.O\_RDWR)

ifr = struct.pack('16sH', b'tun%d', IFF\_TUN | IFF\_NO\_PI)

ifname\_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name

ifname = ifname\_bytes.decode('UTF-8')[:16].strip("\x00")

print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))

os.system("ip link set dev {} up".format(ifname))

while True:

# Get a packet from the tun interface

packet = os.read(tun, 2048)

if packet:

ip = IP(packet)

print(ip.summary())

d

#!/usr/bin/env python3

import fcntl

import struct

import os

import time

from scapy.all import \*

TUNSETIFF = 0x400454ca

IFF\_TUN = 0x0001

IFF\_TAP = 0x0002

IFF\_NO\_PI = 0x1000

# Create the tun interface

tun = os.open("/dev/net/tun", os.O\_RDWR)

ifr = struct.pack('16sH', b'tun%d', IFF\_TUN | IFF\_NO\_PI)

ifname\_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name

ifname = ifname\_bytes.decode('UTF-8')[:16].strip("\x00")

print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))

os.system("ip link set dev {} up".format(ifname))

while True:

# Get a packet from the tun interface

packet = os.read(tun, 2048)

if True:

pkt = IP(packet)

print(pkt.summary())

if ICMP in pkt:

newip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)

newip.ttl = 216

newicmp =ICMP(type=0, id=pkt[ICMP].id,seq=pkt[ICMP].seq)

if pkt.haslayer(Raw):

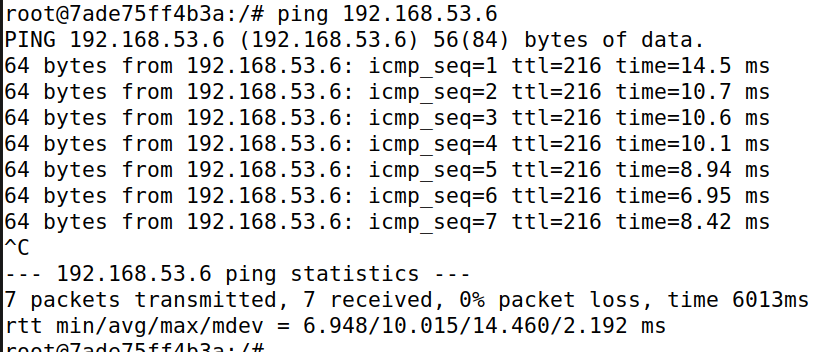
data = pkt[Raw].load

newpkt = newip/newicmp/data

else:

newpkt = newip/newicmp

os.write(tun,bytes(newpkt))



ping通

Task 3

服务器端

#!/usr/bin/env python3

import fcntl

import struct

import os

import time

from scapy.all import \*

TUNSETIFF = 0x400454ca

IFF\_TUN = 0x0001

IFF\_TAP = 0x0002

IFF\_NO\_PI = 0x1000

# Create the tun interface

tun = os.open("/dev/net/tun", os.O\_RDWR)

ifr = struct.pack('16sH', b'tun%d', IFF\_TUN | IFF\_NO\_PI)

ifname\_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name

ifname = ifname\_bytes.decode('UTF-8')[:16].strip("\x00")

print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))

os.system("ip link set dev {} up".format(ifname))

server = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

SERVER\_IP="0.0.0.0"

SERVER\_PORT=9090

server.bind((SERVER\_IP,SERVER\_PORT))

while True:

data,(ip,port) = server.recvfrom(2048)

print("{}:{}-->{}:{}".format(ip,port,SERVER\_IP,SERVER\_PORT))

pkt=IP(data)

print("Inside : {}:{}".format(pkt.src,pkt.dst))

os.write(tun,data)

用户端

#!/usr/bin/env python3

import fcntl

import struct

import os

import time

from scapy.all import \*

TUNSETIFF = 0x400454ca

IFF\_TUN = 0x0001

IFF\_TAP = 0x0002

IFF\_NO\_PI = 0x1000

# Create the tun interface

tun = os.open("/dev/net/tun", os.O\_RDWR)

ifr = struct.pack('16sH', b'tun%d', IFF\_TUN | IFF\_NO\_PI)

ifname\_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name

ifname = ifname\_bytes.decode('UTF-8')[:16].strip("\x00")

print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))

os.system("ip link set dev {} up".format(ifname))

os.system("ip addr add 192.168.60.0/24 dev {}".format(ifname))

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

SERVER\_IP="10.9.0.11"

SERVER\_PORT=9090

while True:

# Get a packet from the tun interface

packet = os.read(tun, 2048)

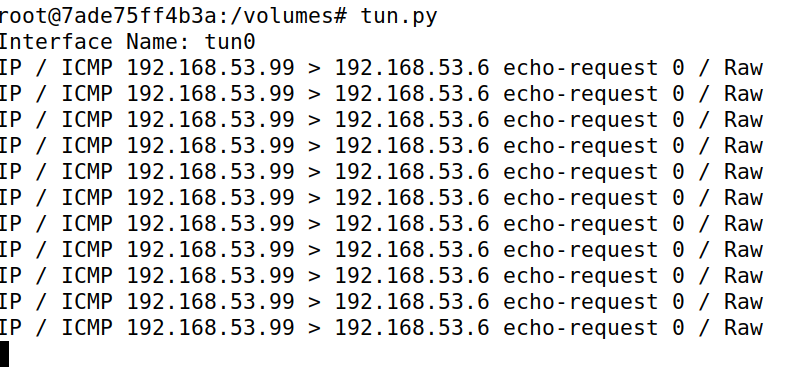
if packet:

pkt = IP(packet)

print(pkt.summary())

sock.sendto(packet, (SERVER\_IP, SERVER\_PORT))

通过隧道发送成功



剩余task4-task6内容

客户端

#!/usr/bin/env python3

import fcntl

import struct

import os

import time

from scapy.all import \*

TUNSETIFF = 0x400454ca

IFF\_TUN = 0x0001

IFF\_TAP = 0x0002

IFF\_NO\_PI = 0x1000

# Create the tun interface

tun = os.open("/dev/net/tun", os.O\_RDWR)

ifr = struct.pack('16sH', b'tun%d', IFF\_TUN | IFF\_NO\_PI)

ifname\_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name

ifname = ifname\_bytes.decode('UTF-8')[:16].strip("\x00")

print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))

os.system("ip link set dev {} up".format(ifname))

os.system("ip addr add 192.168.60.0/24 dev {}".format(ifname))

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

SERVER\_IP="10.9.0.11"

SERVER\_PORT=9090

fds=[sock,tun]

while True:

ready,\_,\_=select.select(fds,[],[])

for fd in ready:

if fd is sock:

data,(ip,port)=sock.recvfrom(2048)

pkt=IP(data)

print("From socket : {} --> {}".format(pkt.src,pkt.dst))

os.write(tun,data)

if fd is tun:

packet = os.read(tun,2048)

if packet:

pkt=IP(packet)

print(pkt.summary())

sock.sendto(packet,(SERVER\_IP,SERVER\_PORT))

服务器

#!/usr/bin/env python3

import fcntl

import struct

import os

import time

from scapy.all import \*

TUNSETIFF = 0x400454ca

IFF\_TUN = 0x0001

IFF\_TAP = 0x0002

IFF\_NO\_PI = 0x1000

# Create the tun interface

tun = os.open("/dev/net/tun", os.O\_RDWR)

ifr = struct.pack('16sH', b'tun%d', IFF\_TUN | IFF\_NO\_PI)

ifname\_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name

ifname = ifname\_bytes.decode('UTF-8')[:16].strip("\x00")

print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))

os.system("ip link set dev {} up".format(ifname))

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

SERVER\_IP="0.0.0.0"

SERVER\_PORT=9090

ip = "10.9.0.5"

port = 10000

sock.bind((SERVER\_IP,SERVER\_PORT))

fds=[sock,tun]

while True:

ready,\_,\_=select.select(fds,[],[])

for fd in ready:

if fd is sock:

print("sock...")

data,(ip,port)=sock.recvfrom(2048)

print("{}:{}-->{}:{}".format(ip,port,SERVER\_IP,SERVER\_PORT))

pkt=IP(data)

print("Inside : {}:{}".format(pkt.src,pkt.dst))

os.write(tun,data)

if fd is tun:

print("tun...")

packet = os.read(tun,2048)

pkt=IP(packet)

print("Return : {}:{}".format(pkt.src,pkt.dst))

sock.sendto(packet,(ip,port))

