Lab 7

57118104 郭雅琪

1

client可以和VPN服务器连接。

```
[07/27/21]seed@VM:~/.../Labsetup$ docksh 89
root@890c445efefe:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.187 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.051 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.078 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.081 ms
^C
```

server-router可以和VPN服务器连接。

```
root@06767248c7f1:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=64 time=0.065 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=64 time=0.075 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=64 time=0.350 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=64 time=0.077 ms
64 bytes from 10.9.0.5: icmp_seq=6 ttl=64 time=0.060 ms
^C
```

client和server-router间不能连接。

```
root@890c445efefe:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
22 packets transmitted, 0 received, 100% packet loss, time 21488ms
```

```
root@890c445efefe:/# ping 192.168.60.6
PING 192.168.60.6 (192.168.60.6) 56(84) bytes of data.
^C
--- 192.168.60.6 ping statistics ---
20 packets transmitted, 0 received, 100% packet loss, time 19454ms
```

在server上ping 10.9.0.11时，在client-router上运行tcpdump，可以捕捉到数据包。

```
root@890c445efefe:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.114 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.098 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.057 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.071 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.074 ms
64 bytes from 10.9.0.11: icmp_seq=6 ttl=64 time=0.099 ms
64 bytes from 10.9.0.11: icmp_seq=7 ttl=64 time=0.072 ms
^C
```

```
root@06767248c7f1:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
09:13:08.729470 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 1, length 64
09:13:08.729508 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 1, length 64
09:13:09.743441 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 2, length 64
09:13:09.743468 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 2, length 64
09:13:10.767777 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 3, length 64
09:13:10.767793 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 3, length 64
09:13:11.791327 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 4, length 64
09:13:11.791347 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 4, length 64
09:13:12.815743 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 5, length 64
09:13:12.815762 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 5, length 64
09:13:13.744035 ARP, Request who-has 10.9.0.5 tell 10.9.0.11, length 28
09:13:13.744226 ARP, Request who-has 10.9.0.11 tell 10.9.0.5, length 28
09:13:13.744251 ARP, Reply 10.9.0.11 is-at 02:42:0a:09:00:0b, length 28
09:13:13.744255 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
```

2

A

在tun.py中做如下修改，将tun修改成gyq。

```
ifr = struct.pack('16sH', b'gyq%d', IFF_TUN | IFF_NO_PI)
```

在client上运行tun.py，可以看到修改的名字。

```
root@890c445efefe:/volumes# tun.py
Interface Name: gyq0
```

运行ip address命令也可以看到更改后的名字。

```
       valid_lft forever preferred_lft forever
3: gyq0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qle
n 500
    link/none
8: eth0@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group de
fault
    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
root@890c445efefe:/#
```

B

在tun.py中增加如下代码：

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

运行tun.py后，使用ifconfig查看信息。

```
gyq0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 192.168.53.99  netmask 255.255.255.0  destination 192.168.53.99
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC
)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

C

将tun.py进行如下修改：

```
while True:
    packet=os.read(tun,2048)
    if packet:
        ip=IP(packet)
        print(ip.summary())
```

在client上ping 192.168.53.0/24网段内的主机，tun.py程序有如下输出：

```
root@890c445efefe:/# 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 ---
16 packets transmitted, 0 received, 100% packet loss, time 15355ms
```

```
root@890c445efefe:/volumes# tun.py
Interface Name: gyq0
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
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
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
```

ping 192.168.60.0/24网段内的主机，tun.py无输出。

```
root@890c445efefe:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
17 packets transmitted, 0 received, 100% packet loss, time 16380ms
```

```
root@890c445efefe:/volumes# tun.py
Interface Name: gyq0
```

D

修改tun.py程序如下：

```
#!/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'gyq%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 = 99
            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))
```

在client上ping 192.168.53.0/24网段上的主机，可以发现有输出。看似可以ping通，但实际没有ping通。

```
root@890c445efefe:/# ping 192.168.53.66
PING 192.168.53.66 (192.168.53.66) 56(84) bytes of data.
64 bytes from 192.168.53.66: icmp_seq=1 ttl=99 time=1.89 ms
64 bytes from 192.168.53.66: icmp_seq=2 ttl=99 time=2.47 ms
64 bytes from 192.168.53.66: icmp_seq=3 ttl=99 time=1.14 ms
64 bytes from 192.168.53.66: icmp_seq=4 ttl=99 time=1.21 ms
64 bytes from 192.168.53.66: icmp_seq=5 ttl=99 time=3.77 ms
64 bytes from 192.168.53.66: icmp_seq=6 ttl=99 time=2.14 ms
64 bytes from 192.168.53.66: icmp_seq=7 ttl=99 time=2.38 ms
64 bytes from 192.168.53.66: icmp_seq=8 ttl=99 time=2.77 ms
```

tun.py脚本有如下输出：

```
root@890c445efefe:/volumes# tun.py
Interface Name: gyq0
IP / ICMP 192.168.53.99 > 192.168.53.66 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.66 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.66 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.66 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.66 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.66 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.66 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.66 echo-request 0 / Raw
```

在接口处输入字符串，无反应。

```
root@890c445efefe:/volumes# tun.py
Interface Name: gyq0
test
```

3

编写tun_server.py， tun_client.py脚本如下：

tun_server.py

```
#!/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'gyq%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 route add 192.168.60.0/24 dev {}".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))
```

tun_client.py

```
#!/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'gyq%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 route add 192.168.60.0/24 dev {}".format(ifname))

# Create UDP socket
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))
```

在ping 192.168.53.0/24网段上的主机时，无输出。

```
root@b9b8f5f1f9ef:/# ping 192.168.53.67
PING 192.168.53.67 (192.168.53.67) 56(84) bytes of data.
^C
--- 192.168.53.67 ping statistics ---
27 packets transmitted, 0 received, 100% packet loss, time 26572ms
```

tun_client.py输出：

```
root@b9b8f5f1f9ef:/volumes# python3 tun_client.py
Interface Name: gyq0
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.67 echo-request 0 / Raw
```

tun_server.py输出：

```
root@568e75678cdb:/volumes# python3 tun_server.py
Interface Name: gyq0
RTNETLINK answers: File exists
10.9.0.5:55990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.67
10.9.0.5:55990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.67
10.9.0.5:55990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.67
10.9.0.5:55990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.67
10.9.0.5:55990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.67
10.9.0.5:55990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.67
10.9.0.5:55990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.67
10.9.0.5:55990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.67
```

4

首先确保配置文件中的net.ipv4.ip_forward已经被设定为1。

```
Router:
    image: handsonsecurity/seed-ubuntu:large
    container_name: server-router
    tty: true
    cap_add:
            - ALL
    devices:
            - "/dev/net/tun:/dev/net/tun"
    sysctls:
            - net.ipv4.ip_forward=1
```

查看router各个接口的ip地址。

```
root@568e75678cdb:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.11  netmask 255.255.255.0  broadcast 10.9.0.255
        ether 02:42:0a:09:00:0b  txqueuelen 0  (Ethernet)
        RX packets 67  bytes 8070 (8.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.60.11  netmask 255.255.255.0  broadcast 192.168.60.255
        ether 02:42:c0:a8:3c:0b  txqueuelen 0  (Ethernet)
        RX packets 69  bytes 8310 (8.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

ping 192.168.60.11时，router上的eth1端口捕捉不到报文。

```
root@b9b8f5f1f9ef:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
```

```
root@568e75678cdb:/# tcpdump -nni eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
```

修改tun_server.py如下：

```python
#!/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'gyq%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))
os.system("ip route add 192.168.60.0/24 dev {}".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)
    print("write")
```

运行tun_server.py，tun_client.py，同时ping 192.168.60.5。

```
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
13 packets transmitted, 0 received, 100% packet loss, time 12288ms
```

```
root@568e75678cdb:/volumes# python3 tun_server.py
Interface Name: gyq0
RTNETLINK answers: File exists
10.9.0.5:42166 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
write
10.9.0.5:42166 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
write
10.9.0.5:42166 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
write
10.9.0.5:42166 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
write
10.9.0.5:42166 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
write
10.9.0.5:42166 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
write
```

```
root@b9b8f5f1f9ef:/volumes# python3 tun_client.py
Interface Name: gyq0
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
```

tcpdump中出现信息，说明ICMP报文到达了目的主机。

```
root@568e75678cdb:/# tcpdump -nni eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
03:10:03.926661 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
03:10:03.926701 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
03:10:03.926705 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 87, seq 1
, length 64
03:10:03.926763 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 87, seq 1,
length 64
03:10:04.949312 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 87, seq 2
, length 64
03:10:04.949366 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 87, seq 2,
length 64
03:10:05.971900 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 87, seq 3
, length 64
03:10:05.972009 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 87, seq 3,
length 64
03:10:06.995917 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 87, seq 4
```

5

修改tun_server.py如下：

```python
#!/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'gyq%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))
os.system("ip route add 192.168.60.0/24 dev {}".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))
```

修改tun_client.py如下：

```
#!/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'gyq%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 route 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:
```

```python
            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))
```

此时ping 192.168.60.5可以ping通。

```
root@b9b8f5f1f9ef:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=2.83 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=1.94 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=1.62 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=3.12 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=4.03 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=3.57 ms
```

tun_server.py输出如下：

```
root@568e75678cdb:/volumes# python3 tun_server.py
Interface Name: gyq0
RTNETLINK answers: File exists
sock...
10.9.0.5:47779 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:47779 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:47779 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:47779 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
```

tun_client.py输出如下：

```
root@b9b8f5f1f9ef:/volumes# python3 tun_client.py
Interface Name: gyq0
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
```

telnet操作也成功。

```
root@b9b8f5f1f9ef:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
```

6

在telnet连接之后，停止运行tun_server.py。

```
Inside: 192.168.53.99 --> 192.168.60.5
^CTraceback (most recent call last):
  File "tun_server.py", line 34, in <module>
    ready,_,_=select.select(fds,[],[])
KeyboardInterrupt
```

此时在远程登录端无法输入信息。

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@a719f0983bce:~$ ls
seed@a719f0983bce:~$
```

重新运行tun_server.py后，可以输入信息。

```
seed@a719f0983bce:~$ ls
seed@a719f0983bce:~$ 1231234llsls
```