

Lab7 VPN Tunnel

57118219 贾志豪

Task 1

Testing: 验证网络连接的正确性

10.9.0.5 ping 10.9.0.11 结果如下:

```
root@91e8820f18eb:/# 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=0.181 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.081 ms
^C
--- 10.9.0.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.042/0.101/0.181/0.058 ms
```

Router ping 192.168.60.5 结果如下:

```
root@0fbc696795bc:/# 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=64 time=0.261 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=64 time=0.045 ms
^C
--- 192.168.60.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2056ms
rtt min/avg/max/mdev = 0.045/0.119/0.261/0.100 ms
```

10.9.0.5 ping 192.168.60.5 结果如下:

```
root@91e8820f18eb:/# 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 ---
4 packets transmitted, 0 received, 100% packet loss, time 3051ms
```

Router 在 10.9.0.0/24 网段上监听结果如下:

```
root@0fbc696795bc:/# 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
01:19:18.186982 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 1, length
 64
01:19:18.187000 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 1, length 6
 4
01:19:19.217022 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 2, length
 64
01:19:19.217038 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 2, length 6
 4
01:19:20.240594 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 3, length
 64
Router 在 192.168.60.0/24 网段上监听结果如下:
```

```
root@0fbc696795bc:/# tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
01:21:25.003561 IP6 fe80::303d:57ff:fe0c:29a3 > ff02::2: ICMP6, router solicitat
ion, length 16
01:21:34.130154 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 29, seq 1
, length 64
01:21:34.130209 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 29, seq 1,
length 64
01:21:35.152631 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 29, seq 2
, length 64
01:21:35.152669 IP 192.168.60.5 > 192.168.60.11: ICMP echo reply, id 29, seq 2,
length 64
01:21:36.177103 IP 192.168.60.11 > 192.168.60.5: ICMP echo request, id 29, seq 3
```

Task 2.A

更改权限并执行：

```
root@91e8820f18eb:/volumes# chmod a+x tun.py
root@91e8820f18eb:/volumes# tun.py
Interface Name: tun0
```

查看 ip address:

```
root@91e8820f18eb:/# 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
2: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default
    qlen 500
        link/none
8: eth0@if9: <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
```

Task 2.B

设置 interface 相连网段与 IP，并置于开启状态

```
19 # Get the interface name
20 ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
21 print("Interface Name: {}".format(ifname))
22 os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
23 os.system("ip link set dev {} up".format(ifname))
24 while True:
25     time.sleep(10)
```

执行：

```
root@91e8820f18eb:/# ip addr add 192.168.53.99/24 dev tun0
root@91e8820f18eb:/# ip link set dev tun0 up
```

查看 ip address，结果 tun0 中多了一条路由，并默认 IP 为 192.168.53.99：

```
root@91e8820f18eb:/# 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
3: 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
        valid_lft forever preferred_lft forever
8: eth0@if9: <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
```

Task 2.C

10.9.0.5 客户端 Ping 192.168.53.2，设置了 192.168.53.99/24 网段通过 tun0，所以 tun0 能接收到 ICMP request 报文：

```
root@91e8820f18eb:/volumes# tun.py
Interface Name: tun0
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.2 echo-request 0 / Raw
```

10.9.0.5 客户端 Ping 192.168.60.5，无输出，ping 报文不经过 tun0，因为没有设置路由：

```
root@91e8820f18eb:/volumes# tun.py
Interface Name: tun0
[...]
root@91e8820f18eb:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
```

Task 2.D

Tun_client.py 中 spoof icmp reply 报文，并写入 tun0，并且负载原有负载

```
24 while True:  
25     # Get a packet from the tun interface  
26     packet = os.read(tun, 2048)  
27     if packet:  
28         ip = IP(packet)  
29         print(ip.summary())  
30         if ip[ICMP].type==8:  
31             newip = IP(src=ip.dst, dst=ip.src)  
32             icmp = ICMP(type=0, code=0)  
33             ip[ICMP].type=0  
34             newpkt = newip/ip.payload  
35             os.write(tun, bytes(newpkt))
```

成功伪造：

```
root@91e8820f18eb:/# ping 192.168.53.2  
PING 192.168.53.2 (192.168.53.2) 56(84) bytes of data.  
64 bytes from 192.168.53.2: icmp_seq=444 ttl=64 time=2.40 ms  
64 bytes from 192.168.53.2: icmp_seq=445 ttl=64 time=2.47 ms  
64 bytes from 192.168.53.2: icmp_seq=446 ttl=64 time=2.55 ms  
64 bytes from 192.168.53.2: icmp_seq=447 ttl=64 time=1.87 ms  
64 bytes from 192.168.53.2: icmp_seq=448 ttl=64 time=1.91 ms  
64 bytes from 192.168.53.2: icmp_seq=449 ttl=64 time=2.58 ms  
64 bytes from 192.168.53.2: icmp_seq=450 ttl=64 time=1.94 ms  
64 bytes from 192.168.53.2: icmp_seq=451 ttl=64 time=2.03 ms  
64 bytes from 192.168.53.2: icmp_seq=452 ttl=64 time=2.40 ms  
64 bytes from 192.168.53.2: icmp_seq=453 ttl=64 time=1.81 ms  
64 bytes from 192.168.53.2: icmp_seq=454 ttl=64 time=2.39 ms
```

Task 3

Tun_clinet.py 客户端设置，加上 192.168.60.0/24 路由，并建立 socket 连接：

```
22 os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))  
23 os.system("ip link set dev {} up".format(ifname))  
24 os.system("ip route add 192.168.60.0/24 dev {}".format(ifname))  
25 |  
26 # Create UDP socket  
27 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  
28 while True:  
29     # Get a packet from the tun interface  
30     packet = os.read(tun, 2048)  
31     if packet:  
32         ip = IP(packet)  
33         print(ip.summary())  
34         # Send the packet via the tunnel  
35         sock.sendto(packet, ('10.9.0.11'. 9090))
```

在服务器端 tun_server.py 建立对应的连接：

```
53 #while True:  
54 #    data, (ip, port) = sock.recvfrom(2048)  
55 #    print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))  
56 #    pkt = IP(data)  
57 #    print(" Inside: {} --> {}".format(pkt.src, pkt.dst))  
58 #    os.write(tun,data)  
59 #    print("write")
```

在客户端 Ping 192.168.53.2，结果如下：

客户端：

服务器端：

客户端添加路由后 Ping 192.168.60.5，结果如下：

客户端：

服务器端：确实收到了来自客户端的 ping 192.168.60.5 的报文：

```
root@e88f38135cbc:/volumes# server.py
Interface Name: tun0
10.9.0.5:53907 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
```

流量报文显示了 UDP 连接，确切使用到了 Tunnel：

14 2021-07-26 03:4... 10.9.0.5	10.9.0.11	UDP	126 53662 → 9090 Len=84
15 2021-07-26 03:4... 10.9.0.5	10.9.0.11	UDP	126 53662 → 9090 Len=84
16 2021-07-26 03:4... 10.9.0.5	10.9.0.11	UDP	126 53662 → 9090 Len=84
17 2021-07-26 03:4... 10.9.0.5	10.9.0.11	UDP	126 53662 → 9090 Len=84
18 2021-07-26 03:4... 10.9.0.5	10.9.0.11	UDP	126 53662 → 9090 Len=84
19 2021-07-26 03:4... 10.9.0.5	10.9.0.11	UDP	126 53662 → 9090 Len=84
20 2021-07-26 03:4... 02:42:0a:09:00:05	02:42:0a:09:00:0b	ARP	42 Who has 10.9.0.11? Tell 10.9.0.5
21 2021-07-26 03:4... 02:42:0a:09:00:05	02:42:0a:09:00:05	ARP	42 10.9.0.11 is at 02:42:0a:09:00:0b
22 2021-07-26 03:4... 10.9.0.5	10.9.0.11	UDP	126 53662 → 9090 Len=84
23 2021-07-26 03:4... 10.9.0.5	10.9.0.11	UDP	126 53662 → 9090 Len=84
24 2021-07-26 03:4... 10.9.0.5	10.9.0.11	UDP	126 53662 → 9090 Len=84

Task 4

服务器端 tun_server.py 改动如下：

```
16 tun = os.open("/dev/net/tun", os.O_RDWR)
17 ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
18 ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
19
20 # Get the interface name
21 ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
22 print("Interface Name: {}".format(ifname))
23 os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))
24 #os.system("ip addr add 192.168.60.0/24 dev {}".format(ifname))
25 os.system("ip link set dev {} up".format(ifname))
26
27 IP_A = "0.0.0.0"
28 PORT = 9090
29 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
30 sock.bind((IP_A, PORT))
31 while True:
32     data, (ip, port) = sock.recvfrom(2048)
33     print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
34     pkt = IP(data)
35     print(" Inside: {} --> {}".format(pkt.src, pkt.dst))
36     os.write(tun,data)
37     print("write")
~~
```

客户端发出 ping 报文结果如下：

客户端没有接收到 reply：

```
Interface Name: tun0
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
```

Server 数据显示，收到 10.9.0.5 发的 ping 报文

```
root@e88f38135cbc:/volumes# server.py
Interface Name: tun0
10.9.0.5:46600 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.60.5
write
10.9.0.5:46600 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.60.5
write
10.9.0.5:46600 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.60.5
write
```

Server 接收到报文后发出，192.168.60.5 接收到了 ICMP request，并且发送了 ICMP reply

7 2021-07-26 02:5...	192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request id=0x002b, seq=25/6400, ttl=63 (reply in...
8 2021-07-26 02:5...	192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply id=0x002b, seq=25/6400, ttl=64 (request ...
9 2021-07-26 02:5...	192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request id=0x002b, seq=26/6656, ttl=63 (reply in...
10 2021-07-26 02:5...	192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply id=0x002b, seq=26/6656, ttl=64 (request ...
11 2021-07-26 02:5...	192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request id=0x002b, seq=27/6912, ttl=63 (reply in...
12 2021-07-26 02:5...	192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply id=0x002b, seq=27/6912, ttl=64 (request ...
13 2021-07-26 02:5...	192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request id=0x002b, seq=28/7168, ttl=63 (reply in...
14 2021-07-26 02:5...	192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply id=0x002b, seq=28/7168, ttl=64 (request ...
15 2021-07-26 02:5...	02:42:c0:a8:3c:0b	02:42:c0:a8:3c:05	ARP	42 Who has 192.168.60.5? Tell 192.168.60.11
16 2021-07-26 02:5...	02:42:c0:a8:3c:05	02:42:c0:a8:3c:0b	ARP	42 192.168.60.5 is at 02:42:c0:a8:3c:05
17 2021-07-26 02:5...	192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request id=0x002b, seq=29/7424, ttl=63 (reply in...
18 2021-07-26 02:5...	192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply id=0x002b, seq=29/7424, ttl=64 (request ...
19 2021-07-26 02:5...	192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request id=0x002b, seq=30/7680, ttl=63 (reply in...
20 2021-07-26 02:5...	192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply id=0x002b, seq=30/7680, ttl=64 (request ...
21 2021-07-26 02:5...	192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request id=0x002b, seq=31/7936, ttl=63 (reply in...
22 2021-07-26 02:5...	192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply id=0x002b, seq=31/7936, ttl=64 (request ...
23 2021-07-26 02:5...	192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request id=0x002b, seq=32/8192, ttl=63 (reply in...
24 2021-07-26 02:5...	192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply id=0x002b, seq=32/8192, ttl=64 (request ...

Task 5

tun_client.py 代码如下：

```
26 # Create UDP socket
27 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
28 fds= [sock,tun]
29 while True:
30     # this will block until at least one interface is ready
31     ready, _, _ = select.select([sock, tun], [], [])
32     for fd in ready:
33         if fd is sock:
34             data, (ip, port) = sock.recvfrom(2048)
35             pkt = IP(data)
36             print("From socket <==: {} --> {}".format(pkt.src, pkt.dst))
37             os.write(tun,data)
38
39         if fd is tun:
40             packet = os.read(tun, 2048)
41             if packet:
42                 ip = IP(packet)
43                 print(ip.summary())
44                 # Send the packet via the tunnel
45                 sock.sendto(packet, ('10.9.0.11', 9090)) |
```

tun_server.py 代码如下：

```
34 fds= [sock,tun]
35 while True:
36     # this will block until at least one interface is ready
37     ready, _, _ = select.select([sock, tun], [], [])
38     for fd in ready:
39         if fd is sock:
40             data, (ip, port) = sock.recvfrom(2048)
41             print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
42             pkt = IP(data)
43             print(" Inside: {} --> {}".format(pkt.src, pkt.dst))
44             os.write(tun,data)
45
46         if fd is tun:
47             print("tun...")
48             packet = os.read(tun, 2048)
49             pkt = IP(packet)
50             print("Return : {} --> {}".format(pkt.src,pkt.dst))
51             sock.sendto(packet, (ip, port))
```

客户端结果如下：

```
root@d89c722cd80c:/volumes# tun.py
Interface Name: tun0
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
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
```

服务器端结果如下：

```
root@e88f38135cbc:/volumes# server.py
Interface Name: tun0
10.9.0.5:60695 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return : 192.168.60.5 --> 192.168.53.99
10.9.0.5:60695 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return : 192.168.60.5 --> 192.168.53.99
10.9.0.5:60695 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return : 192.168.60.5 --> 192.168.53.99
10.9.0.5:60695 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return : 192.168.60.5 --> 192.168.53.99
```

如下是 ping 的结果，得到了 192.168.60.5 返回的 ICMP reply 报文

```
root@d89c722cd80c:/# 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=3.85 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=2.56 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=2.73 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=2.92 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=2.52 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=4.04 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=3.68 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=2.99 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=3.00 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=2.97 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=3.08 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=2.47 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=3.10 ms
^C
--- 192.168.60.5 ping statistics ---
14 packets transmitted, 13 received, 7.14286% packet loss, time 13027ms
rtt min/avg/max/mdev = 2.471/3.071/4.039/0.479 ms
```

以下为 ping 过程中抓取的流量数据：

在 10.9.0.0/24 网段中，有 udp 连接，通过 tunnel 来进行报文传输：

1 2021-07-26 03:1.. 10.9.0.5	10.9.0.11	UDP	126 55479 → 9090 Len=84
2 2021-07-26 03:1.. 10.9.0.11	10.9.0.5	UDP	126 9090 → 55479 Len=84
3 2021-07-26 03:1.. 10.9.0.5	10.9.0.11	UDP	126 55479 → 9090 Len=84
4 2021-07-26 03:1.. 10.9.0.11	10.9.0.5	UDP	126 9090 → 55479 Len=84
5 2021-07-26 03:1.. 10.9.0.5	10.9.0.11	UDP	126 55479 → 9090 Len=84
6 2021-07-26 03:1.. 10.9.0.11	10.9.0.5	UDP	126 9090 → 55479 Len=84
7 2021-07-26 03:1.. 10.9.0.5	10.9.0.11	UDP	126 55479 → 9090 Len=84
8 2021-07-26 03:1.. 10.9.0.11	10.9.0.5	UDP	126 9090 → 55479 Len=84
9 2021-07-26 03:1.. 10.9.0.5	10.9.0.11	UDP	126 55479 → 9090 Len=84
10 2021-07-26 03:1.. 10.9.0.11	10.9.0.5	UDP	126 9090 → 55479 Len=84
11 2021-07-26 03:1.. 10.9.0.5	10.9.0.11	UDP	126 55479 → 9090 Len=84
12 2021-07-26 03:1.. 10.9.0.11	10.9.0.5	UDP	126 9090 → 55479 Len=84

在 192.168.60.0/24 网段中，192.168.60.5 确切收到了 ICMP request 报文，并返回了 reply：

1 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request
2 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply
3 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request
4 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply
5 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request
6 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply
7 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request
8 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply
9 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request
10 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply
11 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	ICMP	98 Echo (ping) request
12 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	ICMP	98 Echo (ping) reply

使用 10.9.0.5 Telnet 192.168.60.5 结果如下：

```
root@d89c722cd80c:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f8aa0344912f login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

This system has been minimized by removing packages and content that are not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*copyright.

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

客户端结果如下：

```
root@d89c722cd80c:/volumes# tun.py
Interface Name: tun0
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet S
From socket <==: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet PA / Raw
From socket <==: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet PA / Raw
From socket <==: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet PA / Raw
From socket <==: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet A
From socket <==: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet PA / Raw
From socket <==: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet A
From socket <==: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:53886 > 192.168.60.5:telnet PA / Raw
```

服务器端结果如下：

```
root@e88f38135cbc:/volumes# server.py
Interface Name: tun0
10.9.0.5:58656 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return : 192.168.60.5 --> 192.168.53.99
10.9.0.5:58656 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:58656 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return : 192.168.60.5 --> 192.168.53.99
tun...
Return : 192.168.60.5 --> 192.168.53.99
10.9.0.5:58656 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:58656 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return : 192.168.60.5 --> 192.168.53.99
tun...
```

以下为 telnet 过程中截取到的报文：

在 10.9.0.0/24 网段中：

1 2021-07-26 03:1.. 10.9.0.5	10.9.0.11	UDP	102 58656 → 9090 Len=60
2 2021-07-26 03:1.. 10.9.0.11	10.9.0.5	UDP	102 9090 → 58656 Len=60
3 2021-07-26 03:1.. 10.9.0.5	10.9.0.11	UDP	94 58656 → 9090 Len=52
4 2021-07-26 03:1.. 10.9.0.5	10.9.0.11	UDP	118 58656 → 9090 Len=76
5 2021-07-26 03:1.. 10.9.0.11	10.9.0.5	UDP	94 9090 → 58656 Len=52
6 2021-07-26 03:1.. 02:42:0a:09:00:0b	Broadcast	ARP	42 Who has 10.9.0.17 Tell 10.9.0.11
7 2021-07-26 03:1.. 02:42:5d:79:88:da	02:42:0a:09:00:0b	ARP	42 10.9.0.1 is at 02:42:5d:79:88:da
8 2021-07-26 03:1.. 192.168.60.5	114.114.114.114	DNS	86 Standard query 0x269e PTR 99.53.168.192.in-addr.arpa
9 2021-07-26 03:1.. 192.168.60.5	8.8.8.8	DNS	86 Standard query 0x269e PTR 99.53.168.192.in-addr.arpa
10 2021-07-26 03:1.. 02:42:0a:09:00:0b	02:42:0a:09:00:05	ARP	42 Who has 10.9.0.5 Tell 10.9.0.11
11 2021-07-26 03:1.. 02:42:0a:09:00:05	02:42:0a:09:00:0b	ARP	42 10.9.0.5 is at 02:42:0a:09:00:05
12 2021-07-26 03:1.. 02:42:0a:09:00:05	02:42:0a:09:00:0b	ARP	42 Who has 10.9.0.11? Tell 10.9.0.5
13 2021-07-26 03:1.. 02:42:0a:09:00:0b	02:42:0a:09:00:05	ARP	42 10.9.0.11 is at 02:42:0a:09:00:0b
14 2021-07-26 03:1.. 192.168.60.5	114.114.114.114	DNS	86 Standard query 0x269e PTR 99.53.168.192.in-addr.arpa

在 192.168.60.0/24 网段中，显示确切建立了 telnet 连接：

1 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	TCP	74 53886 → 23 [SYN] Seq=3827690171 Win=64240 Len=0 MSS=1460
2 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	TCP	74 23 → 53886 [SYN, ACK] Seq=214075036 Ack=3827690172 Win=64256
3 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	TCP	66 53886 → 23 [ACK] Seq=3827690172 Ack=214075037 Win=64256
4 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	TELNET	90 Telnet Data ...
5 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	TCP	66 23 → 53886 [ACK] Seq=214075037 Ack=3827690196 Win=65152
6 2021-07-26 03:1.. 192.168.60.5	114.114.114.114	DNS	86 Standard query 0x269e PTR 99.53.168.192.in-addr.arpa
7 2021-07-26 03:1.. 192.168.60.5	8.8.8.8	DNS	86 Standard query 0x269e PTR 99.53.168.192.in-addr.arpa
8 2021-07-26 03:1.. 02:42:c0:a8:3c:0b	02:42:c0:a8:3c:05	ARP	42 Who has 192.168.60.5? Tell 192.168.60.11
9 2021-07-26 03:1.. 02:42:c0:a8:3c:05	02:42:c0:a8:3c:0b	ARP	42 192.168.60.5 is at 02:42:c0:a8:3c:05
10 2021-07-26 03:1.. 192.168.60.5	114.114.114.114	DNS	86 Standard query 0x269e PTR 99.53.168.192.in-addr.arpa
11 2021-07-26 03:1.. 192.168.60.5	8.8.8.8	DNS	86 Standard query 0x269e PTR 99.53.168.192.in-addr.arpa
12 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	TELNET	78 Telnet Data ...
13 2021-07-26 03:1.. 192.168.53.99	192.168.60.5	TCP	66 53886 → 23 [ACK] Seq=3827690196 Ack=214075049 Win=64256
14 2021-07-26 03:1.. 192.168.60.5	192.168.53.99	TELNET	81 Telnet Data ...
15 2021-07-26 03:1.. 192.168.60.5	192.168.60.5	TELNET	80 Telnet Data ...

Task 6

通过观察和抓取流量数据得出：

当 telnet 连接时，一旦中断 tun_client.py 或者 tun_server.py 其一，telnet 客户端输入无反应，再次开启 tun_client.py 和 tun_server.py，不用重新 telnet 申请，可以继续使用中断前的 telnet，查看流量包文显示，tcp 连接并没有断开。

因为 tcp 一开始已经建立了 telnet 连接，当中断 tun_client.py 或者 tun_server.py 时并没有发出任何报文取消此次 tcp 连接，192.168.60.5 以为仍然在与 192.168.53.99 建立 telnet 连接，但对于 10.9.0.5 来说，tunnel 不服存在，故而无法发送消息，无法连接上 telnet。