Lab7

57118232 谢隆文

Task 1：Network Setup

验证主机 U 可以与 VPN Server 通信以及在路由器上 tcpdump

```
root@ff649d991563:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
64 bytes from 192.168.60.11: icmp_seq=1 ttl=64 time=0.073 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.113 ms
64 bytes from 192.168.60.11: icmp_seq=3 ttl=64 time=0.070 ms
64 bytes from 192.168.60.11: icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from 192.168.60.11: icmp_seq=5 ttl=64 time=0.068 ms
^C
--- 192.168.60.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4074ms
rtt min/avg/max/mdev = 0.058/0.076/0.113/0.019 ms
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
12:36:55.269731 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 1, length 64
12:36:55.269749 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 1, length 64
12:36:56.271637 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 2, length 64
12:36:56.271666 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 2, length 64
12:36:57.294817 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 3, length 64
12:36:57.294838 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 3, length 64
12:36:58.318859 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 4, length 64
12:36:58.318875 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 4, length 64
12:36:59.343899 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 5, length 64
12:36:59.343919 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 5, length 64
12:37:00.498824 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, lengt
```

验证主机 V 可以与 VPN Server 通信以及在路由器上 tcpdump 捕获的报文

```
root@ff649d991563:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
64 bytes from 192.168.60.11: icmp_seq=1 ttl=64 time=0.073 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.113 ms
64 bytes from 192.168.60.11: icmp_seq=3 ttl=64 time=0.070 ms
64 bytes from 192.168.60.11: icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from 192.168.60.11: icmp_seq=5 ttl=64 time=0.068 ms
^C
--- 192.168.60.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4074ms
rtt min/avg/max/mdev = 0.058/0.076/0.113/0.019 ms
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
12:36:55.269731 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 1, length 64
12:36:55.269749 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 1, length 64
12:36:56.271637 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 2, length 64
12:36:56.271666 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 2, length 64
12:36:57.294817 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 3, length 64
12:36:57.294838 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 3, length 64
12:36:58.318859 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 4, length 64
12:36:58.318875 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 4, length 64
12:36:59.343899 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 5, length 64
12:36:59.343919 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 5, length 64
12:37:00.498824 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, lengt
```

验证 U 和 V 不通。

```
root@70a884552c63:/# 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 ---
7 packets transmitted, 0 received, 100% packet loss, time 6148ms
```

Task 2：Create and Configure TUN Interface

Task 2.A：Name of the Interface

在代码此处将 tun 修改成自己名字简拼 xlw 。

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

```
root@70a884552c63:/volume# chmod a+x tun.py
root@70a884552c63:/volume# chmod tun.py
Interface Name: xlw0
```

Task 2.B：Set up the TUN Interface

在 tun.py 文件中添加以下两行代码，编译运行后主机 U(10.9.0.5) 上运行 ifconfig 查看所有接口， 可观察到绑定 IP 地址。

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

Task 2.C：Read from the TUN Interface

ping 192.168.53.1 ，可以看到程序有输出，但是请求无响应，因为实际主机不存在。

```
root@70a884552c63:/# 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 ---
12 packets transmitted, 0 received, 100% packet loss, time 11293ms
```

U 上运行代码并 ping 192.168.60.1，可以 ping 通，但程序没有打印任何结果，因为没有添 加路由。

```
PING 192.168.60.1 (192.168.60.1) 56(84) bytes of data.
64 bytes from 192.168.60.1: icmp_seq=1 ttl=64 time=0.058 ms
64 bytes from 192.168.60.1: icmp_seq=2 ttl=64 time=0.074 ms
64 bytes from 192.168.60.1: icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from 192.168.60.1: icmp_seq=4 ttl=64 time=0.095 ms
64 bytes from 192.168.60.1: icmp_seq=5 ttl=64 time=0.062 ms
64 bytes from 192.168.60.1: icmp_seq=6 ttl=64 time=0.056 ms
64 bytes from 192.168.60.1: icmp_seq=7 ttl=64 time=0.058 ms
64 bytes from 192.168.60.1: icmp_seq=8 ttl=64 time=0.086 ms
64 bytes from 192.168.60.1: icmp_seq=9 ttl=64 time=0.078 ms
64 bytes from 192.168.60.1: icmp_seq=10 ttl=64 time=0.079 ms
64 bytes from 192.168.60.1: icmp_seq=11 ttl=64 time=0.044 ms
64 bytes from 192.168.60.1: icmp_seq=12 ttl=64 time=0.046 ms
^C
--- 192.168.60.1 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11263ms
rtt min/avg/max/mdev = 0.044/0.067/0.095/0.015 ms
```

Task 2.D：Write to the TUN Interface

代码如下：

```
#!/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'xlwl%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))
```

```
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
^CTraceback (most recent call last):
  File "./tun.py", line 28, in <module>
    packet = os.read(tun, 2048)
KeyboardInterrupt
```