57118231　向颖

Task1.1 A

```
[07/05/21]seed@VM:~/.../volumes$ sudo python3 sniffer.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:98:08:59:52
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 10258
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0xfe7f
     src       = 10.9.0.1
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0x71ef
        id        = 0x2
        seq       = 0x1
```

```
[07/05/21]seed@VM:~/Desktop$ 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=0.169 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=64 time=0.059 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=64 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=64 time=0.053 ms
64 bytes from 10.9.0.5: icmp_seq=6 ttl=64 time=0.056 ms
64 bytes from 10.9.0.5: icmp_seq=7 ttl=64 time=0.100 ms
64 bytes from 10.9.0.5: icmp_seq=8 ttl=64 time=0.045 ms
```

非 root 权限下无法运行

```
[07/05/21]seed@VM:~/.../volumes$ python3 sniffer.py
Traceback (most recent call last):
  File "sniffer.py", line 6, in <module>
    pkt = sniff(iface='br-12f3b39d28b1', filter='icmp', prn=print_pkt)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 1036, in sniff
    sniffer._run(*args, **kwargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 906, in _run
    sniff_sockets[L2socket(type=ETH_P_ALL, iface=iface,
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 398, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(type))  # noqa: E501
  File "/usr/lib/python3.8/socket.py", line 231, in __init__
    _socket.socket.__init__(self, family, type, proto, fileno)
PermissionError: [Errno 1] Operation not permitted
[07/05/21]seed@VM:~/.../volumes$
```

Task1.1B

1.ICMP

同 1.1A

2.TCP

sniffer

```
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()

pkt = sniff(filter='tcp and src host 10.9.0.1 and dst port 23', prn=print_pkt)
```

发包程序

sniffer.py

```
from scapy.all import *

ip=IP()
ip.src='10.9.0.1'
ip.dst='10.9.0.5'
tcp=TCP()
tcp.dport=23
send(ip/tcp)
```

结果

```
^C[07/05/21]seed@VM:~/.../volumes$ sudo python3 sniffer.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:37:52:f1:34
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 40
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x66b8
     src       = 10.9.0.1
     dst       = 10.9.0.5
     \options   \
###[ TCP ]###
        sport     = ftp_data
        dport     = telnet
        seq       = 0
        ack       = 0
        dataofs   = 5
        reserved  = 0
        flags     = S
        window    = 8192
        chksum    = 0x7ba0
        urgptr    = 0
        options   = []
```

3.网段

发包程序

```
Open
1 from scapy.all import *
2
3 send(IP(dst='128.230.0.0/16'))
```

Sniffer

```
Open                                              sniffer.py
                                                  ~/Desktop/lab1/Labsetup/volumes
1 #!/usr/bin/env python3
2 from scapy.all import *
3 def print_pkt(pkt):
4         pkt.show()
5
6 pkt = sniff(filter='dst net 128.230.0.0/16', prn=print_pkt)
```

结果

```
###[ Ethernet ]###
  dst       = 76:7e:b7:63:ad:6f
  src       = 00:0c:29:86:f4:8c
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 20
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = hopopt
     chksum    = 0xde11
     src       = 192.168.43.59
     dst       = 128.230.48.14
     \options   \
```

Task1.2

发包程序

```
~/Desktop/lab1/Labsetup/volumes
1 from scapy.all import *
2
3 send((IP(src='10.9.0.2',dst='10.9.0.5'))/ICMP())
```

Wireshark 查看

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-07-07 10:48:31.068979874	02:42:c9:19:78:aa	Broadcast	ARP	42	Who has 1
2	2021-07-07 10:48:31.068993983	02:42:0a:09:00:05	02:42:c9:19:78:aa	ARP	42	10.9.0.5
3	2021-07-07 10:48:31.084520481	10.9.0.2	10.9.0.5	ICMP	42	Echo (pin
4	2021-07-07 10:48:31.084554897	02:42:0a:09:00:05	Broadcast	ARP	42	Who has 1
5	2021-07-07 10:48:32.093493688	02:42:0a:09:00:05	Broadcast	ARP	42	Who has 1
6	2021-07-07 10:48:33.118636360	02:42:0a:09:00:05	Broadcast	ARP	42	Who has 1

Task1.3

发包程序

```
1 from scapy.all import *
2
3 ans,unans=sr(IP(dst='www.baidu.com', ttl=(4,25))/TCP(flags=0x2))
4 for snd,rcv in ans:
5     print(snd.ttl, rcv.src, isinstance(rcv.payload, TCP))
6
```

结果

```
[07/05/21]seed@VM:~/.../volumes$ sudo python3 tracert.py
Begin emission:
Finished sending 22 packets.
.******************.^C
Received 21 packets, got 19 answers, remaining 3 packets
4 112.80.4.129 False
5 221.6.5.29 False
6 182.61.216.0 False
7 58.240.60.170 False
8 221.6.1.250 False
9 112.80.248.76 True
10 112.80.248.76 True
11 112.80.248.76 True
12 112.80.248.76 True
13 112.80.248.76 True
14 112.80.248.76 True
15 112.80.248.76 True
16 112.80.248.76 True
17 112.80.248.76 True
18 112.80.248.76 True
19 112.80.248.76 True
20 112.80.248.76 True
21 112.80.248.76 True
22 112.80.248.76 True
[07/05/21]seed@VM:~/.../volumes$
```

Task1.4

Sniff then spoof 程序

```python
#user/bin/python3
from scapy.all import *

def spoof_pkt(pkt):
        if ICMP in pkt and pkt[ICMP].type==8:
                print("origin packet ....")
                print("src ip:",pkt[IP].src)
                print("dst ip:",pkt[IP].dst)

                ip=IP(src=pkt[IP].dst,dst=pkt[IP].src,ihl=pkt[IP].ihl)
                icmp=ICMP(type=0,id=pkt[ICMP].id,seq=pkt[ICMP].seq)
                data=pkt[Raw].load
                newpkt=ip/icmp/data

                print("spoof packet ....")
                print("src ip:",newpkt[IP].src)
                print("dst ip:",newpkt[IP].dst)
                send(newpkt,verbose=0)

pkt=sniff(filter='icmp',iface='br-12f3b39d28b1',prn=spoof_pkt)

```

Ping 1.2.3.4

```
root@97d35a215f5d:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=64 time=66.4 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=16.6 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=19.6 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=20.2 ms
^C
--- 1.2.3.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3010ms
```

```
origin packet ....
src ip: 10.9.0.5
dst ip: 1.2.3.4
spoof packet ....
src ip: 1.2.3.4
dst ip: 10.9.0.5
origin packet ....
src ip: 10.9.0.5
dst ip: 1.2.3.4
spoof packet ....
src ip: 1.2.3.4
dst ip: 10.9.0.5
origin packet ....
src ip: 10.9.0.5
dst ip: 1.2.3.4
spoof packet ....
src ip: 1.2.3.4
dst ip: 10.9.0.5
origin packet ....
src ip: 10.9.0.5
dst ip: 1.2.3.4
spoof packet ....
src ip: 1.2.3.4
dst ip: 10.9.0.5
origin packet
```

ping 10.9.0.99

```
root@97d35a215f5d:/# ping 10.9.0.99
PING 10.9.0.99 (10.9.0.99) 56(84) bytes of data.
From 10.9.0.5 icmp_seq=1 Destination Host Unreachable
From 10.9.0.5 icmp_seq=2 Destination Host Unreachable
From 10.9.0.5 icmp_seq=3 Destination Host Unreachable
From 10.9.0.5 icmp_seq=4 Destination Host Unreachable
From 10.9.0.5 icmp_seq=5 Destination Host Unreachable
From 10.9.0.5 icmp_seq=6 Destination Host Unreachable
^C
--- 10.9.0.99 ping statistics ---
7 packets transmitted, 0 received, +6 errors, 100% packet loss, time 6146ms
```

```
seed@VM: ~/.../volumes
root@VM:/volumes# python3 sniff_spoof.py
```

Ping 8.8.8.8

```
root@97d35a215f5d:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=64 time=70.9 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=52 time=71.5 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=16.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=52 time=30.4 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=64 time=19.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=52 time=96.7 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=4 ttl=64 time=20.2 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=52 time=42.9 ms (DUP!)
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, +4 duplicates, 0% packet loss, time 3017ms
```

```
root@VM:/volumes# python3 sniff_spoof.py
origin packet ....
src ip: 10.9.0.5
dst ip: 8.8.8.8
spoof packet ....
src ip: 8.8.8.8
dst ip: 10.9.0.5
origin packet ....
src ip: 10.9.0.5
dst ip: 8.8.8.8
spoof packet ....
src ip: 8.8.8.8
dst ip: 10.9.0.5
origin packet ....
src ip: 10.9.0.5
dst ip: 8.8.8.8
spoof packet ....
src ip: 8.8.8.8
dst ip: 10.9.0.5
```

结论：由于 1.2.3.4 和 8.8.8.8 和 host 10.9.0.5 不在一个子网下，报文需要通过网关 10.9.0.1，就被劫持并伪造了假的 icmp reply。而 10.9.0.99 在同一个子网下，通过 ARP 广播获得 MAC 地址来发送，而没有收到 ARP 响应就 ping 不通。