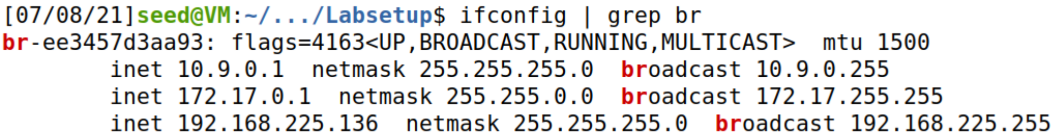
LAB 1

57118116 陈煜

Task 1.1A

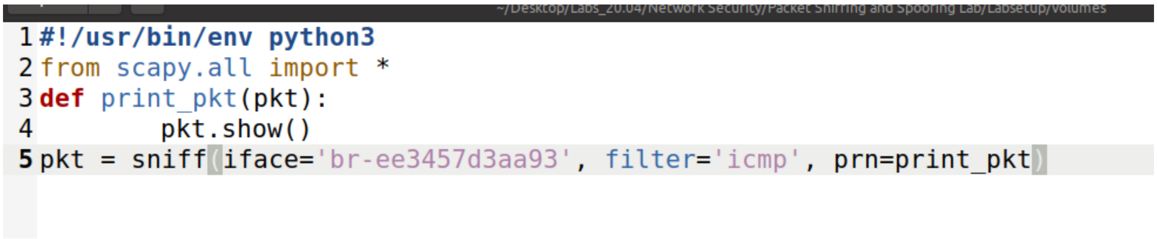
查看网络接口：



```
[07/08/21]seed@VM:~/.../Labsetup$ ifconfig | grep br
br-ee3457d3aa93: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet 192.168.225.136  netmask 255.255.255.0  broadcast 192.168.225.255
```

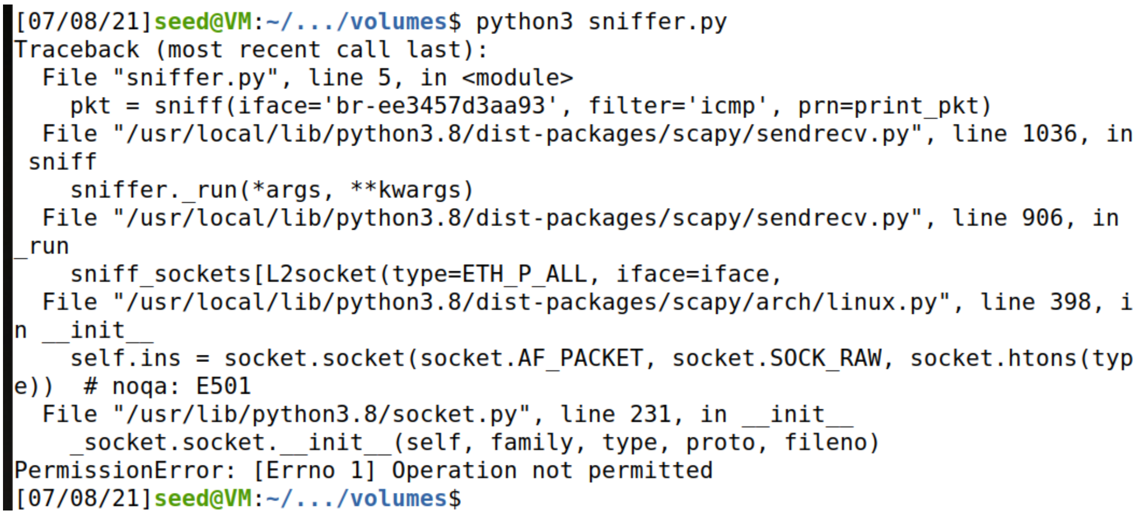
接口为 br-ee3457d3aa93。

进入 volumes 目录下，新建 sniffer.py 文件：


~/Desktop/Labs_20.04/Network Security/Packet Sniffing and Spoofing Lab/Labsetup/volumes

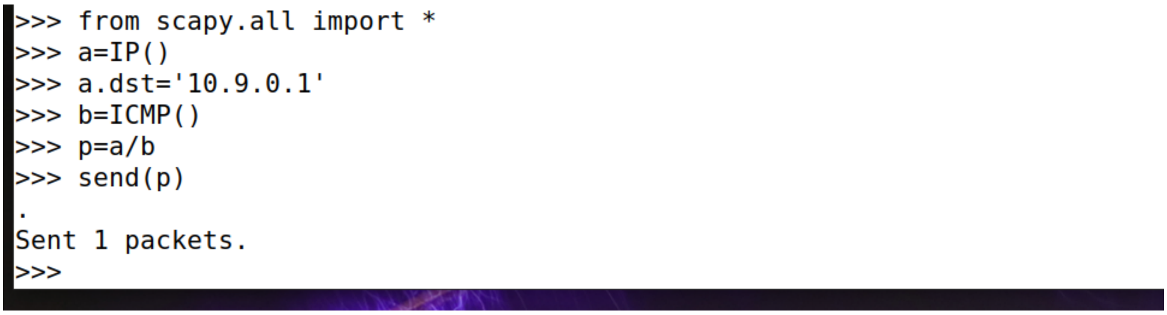
```
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()
pkt = sniff(iface='br-ee3457d3aa93', filter='icmp', prn=print_pkt)
```

在普通用户下运行，无法运行：

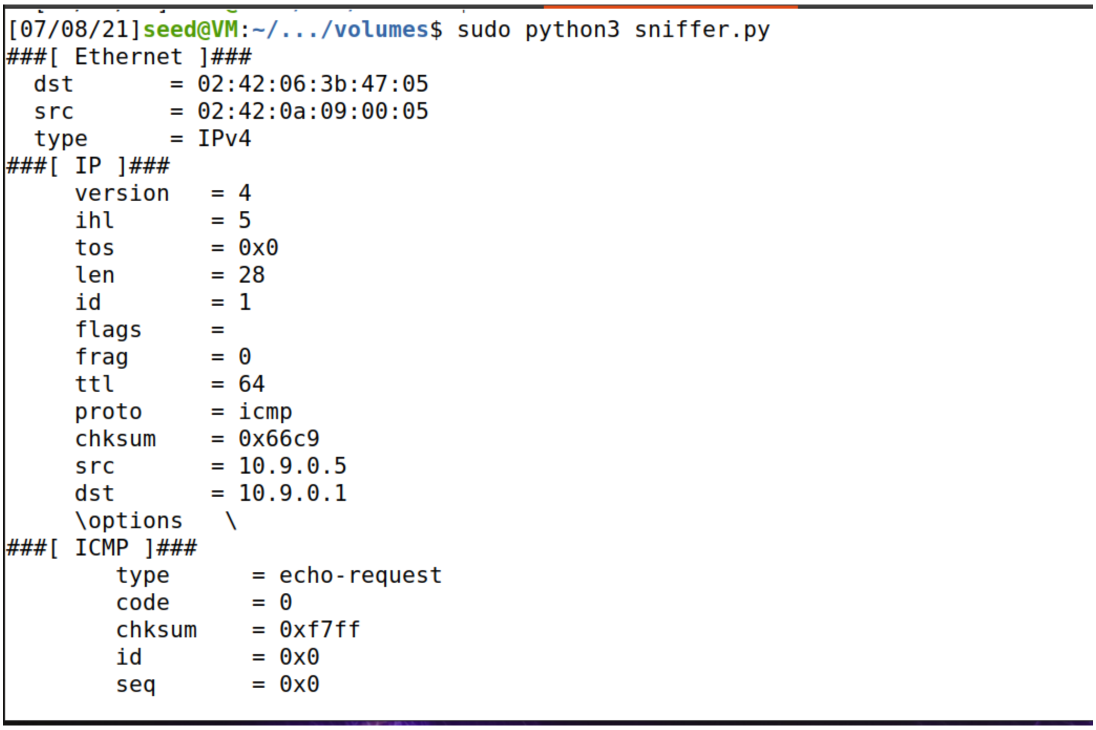


```
[07/08/21]seed@VM:~/.../volumes$ python3 sniffer.py
Traceback (most recent call last):
  File "sniffer.py", line 5, in <module>
    pkt = sniff(iface='br-ee3457d3aa93', filter='icmp', prn=print_pkt)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 1036, in
 sniff
    sniffer._run(*args, **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, i
n __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(typ
e))  # 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
[07/08/21]seed@VM:~/.../volumes$
```

在 root 下运行程序，同时在 docker 上构造报文并发送，发现成功抓到报文：



```
>>> from scapy.all import *
>>> a=IP()
>>> a.dst='10.9.0.1'
>>> b=ICMP()
>>> p=a/b
>>> send(p)
.
Sent 1 packets.
>>>
```



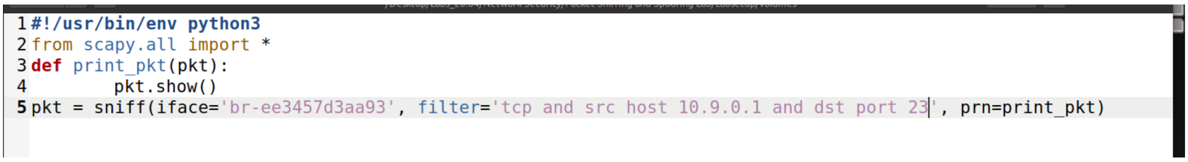
```
[07/08/21]seed@VM:~/.../volumes$ sudo python3 sniffer.py
###[ Ethernet ]###
  dst       = 02:42:06:3b:47:05
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x66c9
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xf7ff
        id        = 0x0
        seq       = 0x0
```

Task1.1B

（1） 只捕获 ICMP 报文。结果如 Task1.1A 所示。

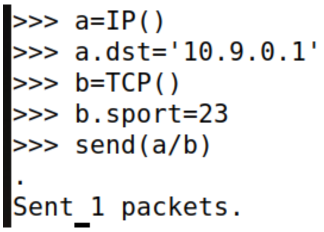
（2） 只捕获来自特定 IP、目的端口 23 的数据包。

修改 sniffer.py：



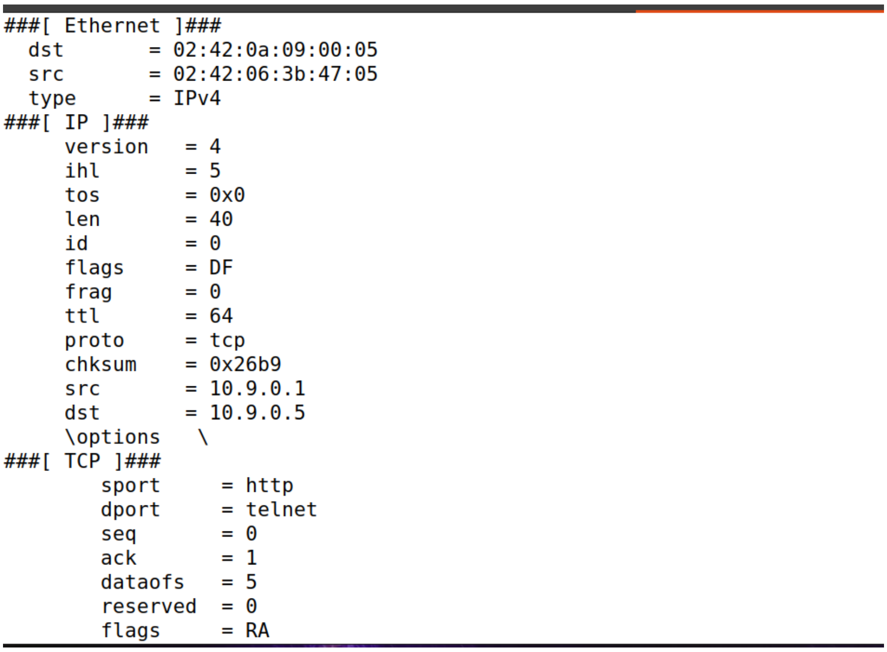
```
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()
pkt = sniff(iface='br-ee3457d3aa93', filter='tcp and src host 10.9.0.1 and dst port 23', prn=print_pkt)
```

构造响应数据包并发送：



```
>>> a=IP()
>>> a.dst='10.9.0.1'
>>> b=TCP()
>>> b.sport=23
>>> send(a/b)
.
Sent 1 packets.
```

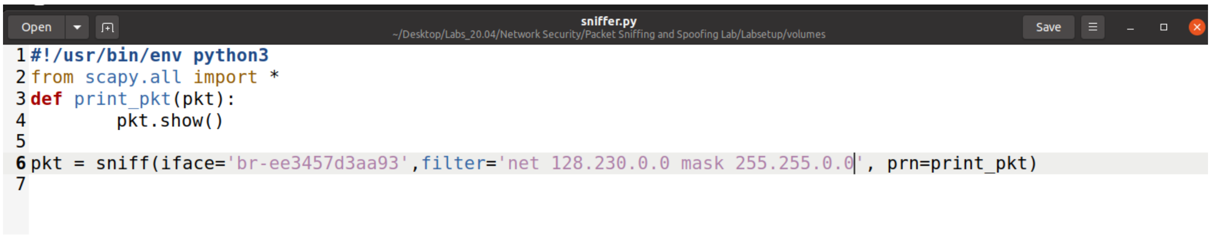
捕获的结果：



```
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:06:3b:47:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 40
     id        = 0
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x26b9
     src       = 10.9.0.1
     dst       = 10.9.0.5
     \options   \
###[ TCP ]###
        sport     = http
        dport     = telnet
        seq       = 0
        ack       = 1
        dataofs   = 5
        reserved  = 0
        flags     = RA
```

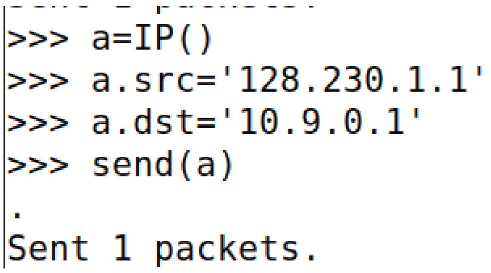
（3） 来自特定子网的报文。

修改 sniffer.py：


Open
sniffer.py
~/Desktop/Labs_20.04/Network Security/Packet Sniffing and Spoofing Lab/Labsetup/volumes
Save

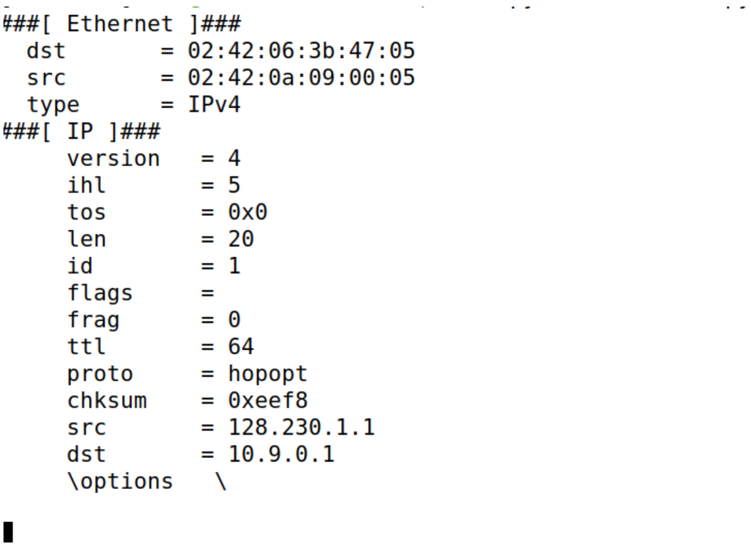
```
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()

pkt = sniff(iface='br-ee3457d3aa93',filter='net 128.230.0.0 mask 255.255.0.0', prn=print_pkt)

```

构造数据包并发送：



```
>>> a=IP()
>>> a.src='128.230.1.1'
>>> a.dst='10.9.0.1'
>>> send(a)
.
Sent 1 packets.
```

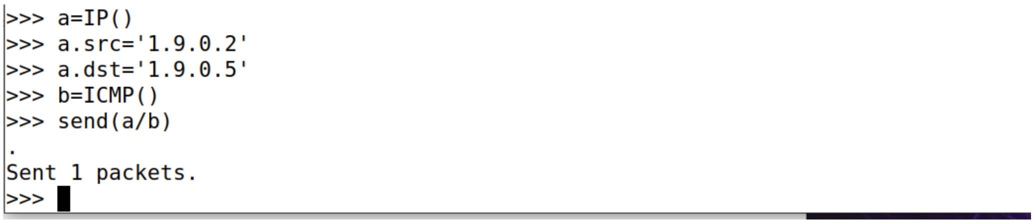
捕获的结果：



```
###[ Ethernet ]###
  dst       = 02:42:06:3b:47:05
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 20
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = hopopt
     chksum    = 0xeef8
     src       = 128.230.1.1
     dst       = 10.9.0.1
     \options   \
```

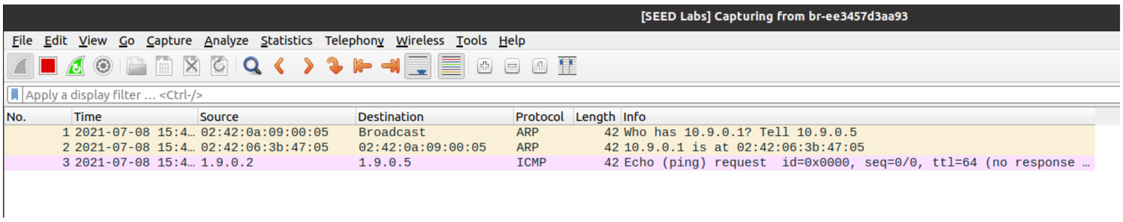
Task 1.2

打开 Wireshark。构造报文并发送：



```
>>> a=IP()
>>> a.src='1.9.0.2'
>>> a.dst='1.9.0.5'
>>> b=ICMP()
>>> send(a/b)
.
Sent 1 packets.
>>> 
```

Wireshark 捕获的结果：

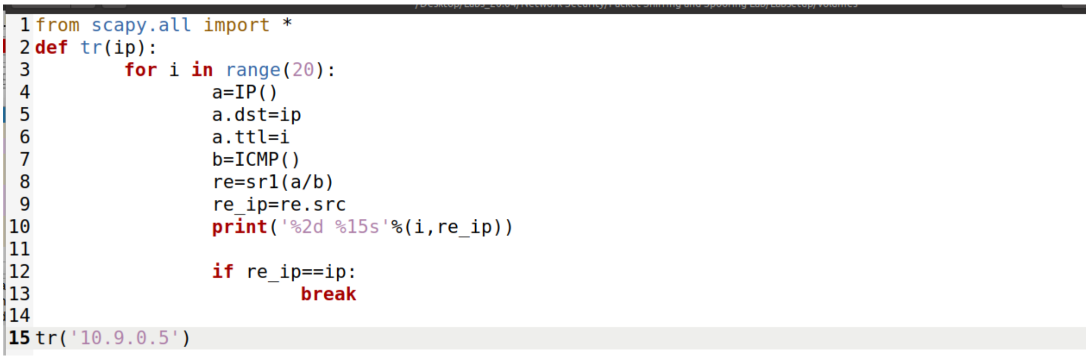


[SEED Labs] Capturing from br-ee3457d3aa93
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-07-08 15:4…	02:42:0a:09:00:05	Broadcast	ARP	42	Who has 10.9.0.1? Tell 10.9.0.5
2	2021-07-08 15:4…	02:42:06:3b:47:05	02:42:0a:09:00:05	ARP	42	10.9.0.1 is at 02:42:06:3b:47:05
3	2021-07-08 15:4…	1.9.0.2	1.9.0.5	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response …

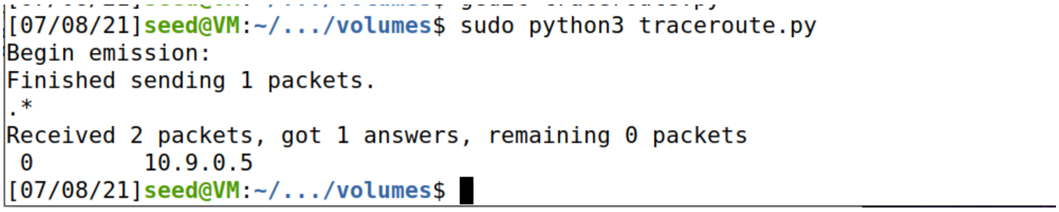
Task 1.3

编写 traceroute.py 代码：



```python
from scapy.all import *
def tr(ip):
        for i in range(20):
                a=IP()
                a.dst=ip
                a.ttl=i
                b=ICMP()
                re=sr1(a/b)
                re_ip=re.src
                print('%2d %15s'%(i,re_ip))

                if re_ip==ip:
                        break

tr('10.9.0.5')
```

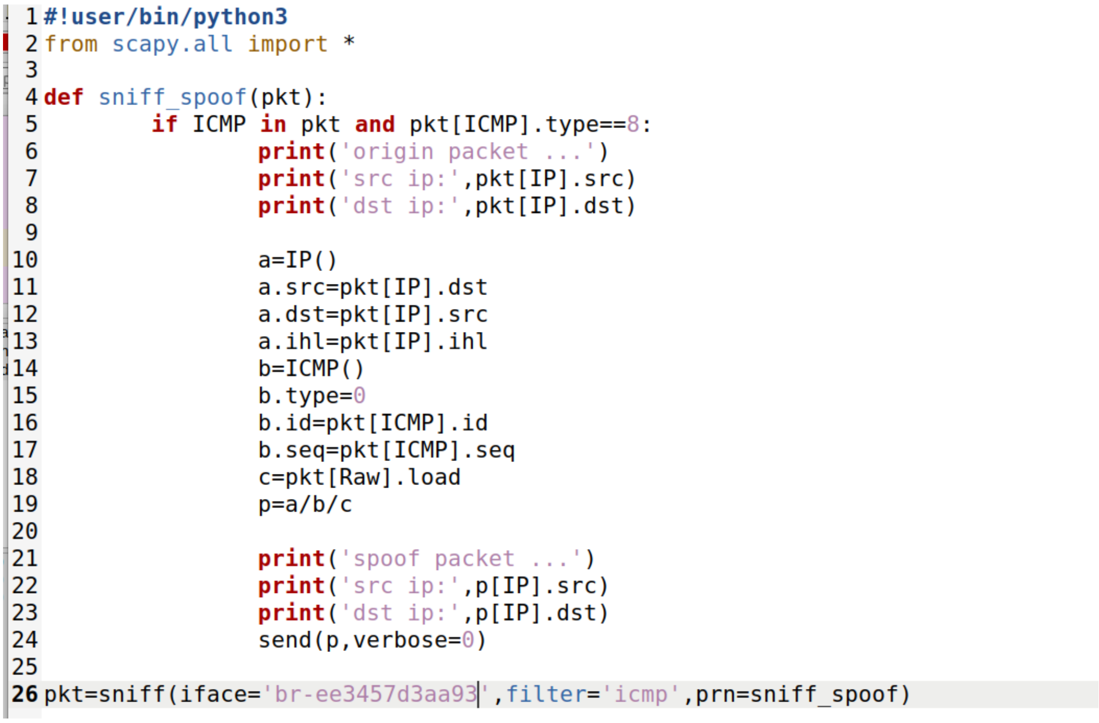
运行程序：



```
[07/08/21]seed@VM:~/.../volumes$ sudo python3 traceroute.py
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
 0       10.9.0.5
[07/08/21]seed@VM:~/.../volumes$ 
```

Task 1.4

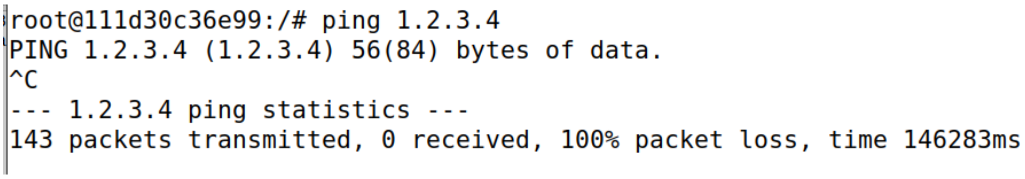
编写代码 sniff_spoof.py：



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

def sniff_spoof(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)

                a=IP()
                a.src=pkt[IP].dst
                a.dst=pkt[IP].src
                a.ihl=pkt[IP].ihl
                b=ICMP()
                b.type=0
                b.id=pkt[ICMP].id
                b.seq=pkt[ICMP].seq
                c=pkt[Raw].load
                p=a/b/c

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

pkt=sniff(iface='br-ee3457d3aa93',filter='icmp',prn=sniff_spoof)
```

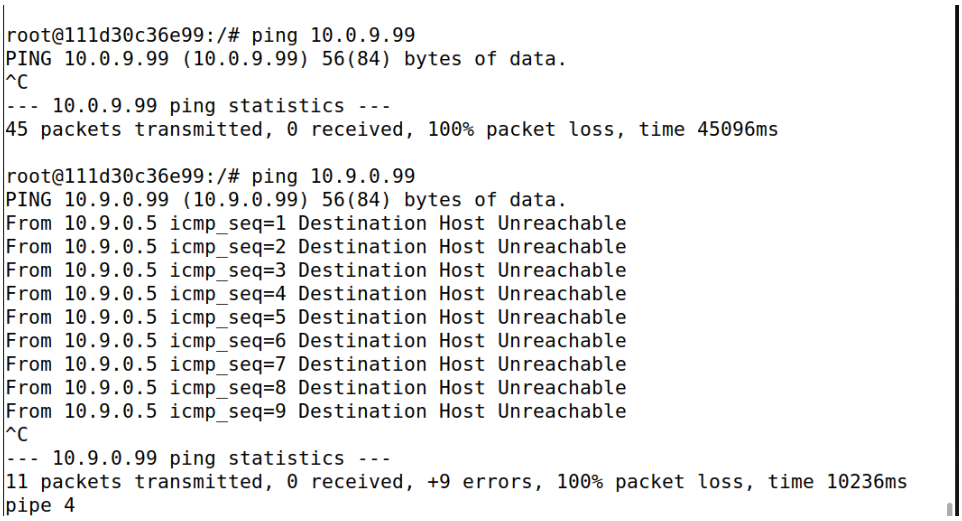
在运行程序前，分别 ping 三个地址：

1.2.3.4：



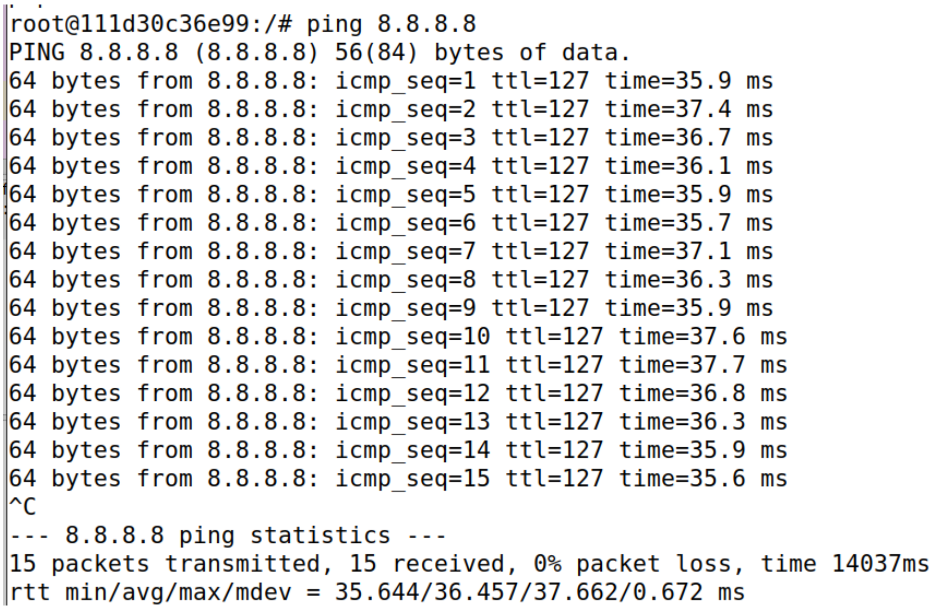
```
root@111d30c36e99:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
^C
--- 1.2.3.4 ping statistics ---
143 packets transmitted, 0 received, 100% packet loss, time 146283ms
```

10.0.9.99：



```
root@111d30c36e99:/# ping 10.0.9.99
PING 10.0.9.99 (10.0.9.99) 56(84) bytes of data.
^C
--- 10.0.9.99 ping statistics ---
45 packets transmitted, 0 received, 100% packet loss, time 45096ms

root@111d30c36e99:/# 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
From 10.9.0.5 icmp_seq=7 Destination Host Unreachable
From 10.9.0.5 icmp_seq=8 Destination Host Unreachable
From 10.9.0.5 icmp_seq=9 Destination Host Unreachable
^C
--- 10.9.0.99 ping statistics ---
11 packets transmitted, 0 received, +9 errors, 100% packet loss, time 10236ms
pipe 4
```

8.8.8.8：

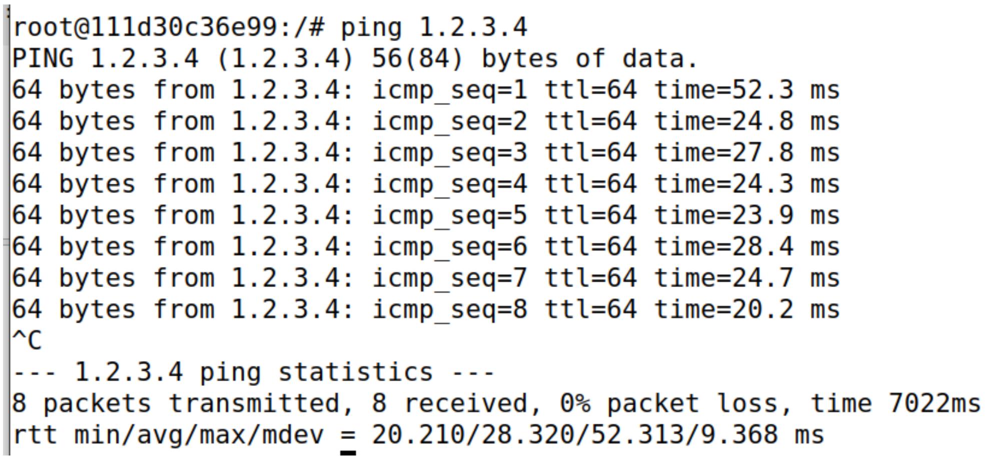


```
root@111d30c36e99:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=35.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=37.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=127 time=36.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=127 time=36.1 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=127 time=35.9 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=127 time=35.7 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=127 time=37.1 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=127 time=36.3 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=127 time=35.9 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=127 time=37.6 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=127 time=37.7 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=127 time=36.8 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=127 time=36.3 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=127 time=35.9 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=127 time=35.6 ms
^C
--- 8.8.8.8 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14037ms
rtt min/avg/max/mdev = 35.644/36.457/37.662/0.672 ms
```

可以发现，在运行 spiff_spoof.py 前，无法 ping 通 1.2.3.4 和 10.0.9.99，但能 ping 通 8.8.8.8。

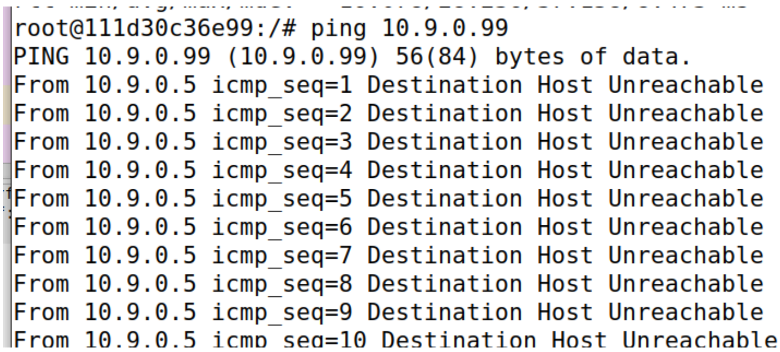
运行 sniff_spoof.py，再次 ping 三个地址：

Ping 1.2.3.4：



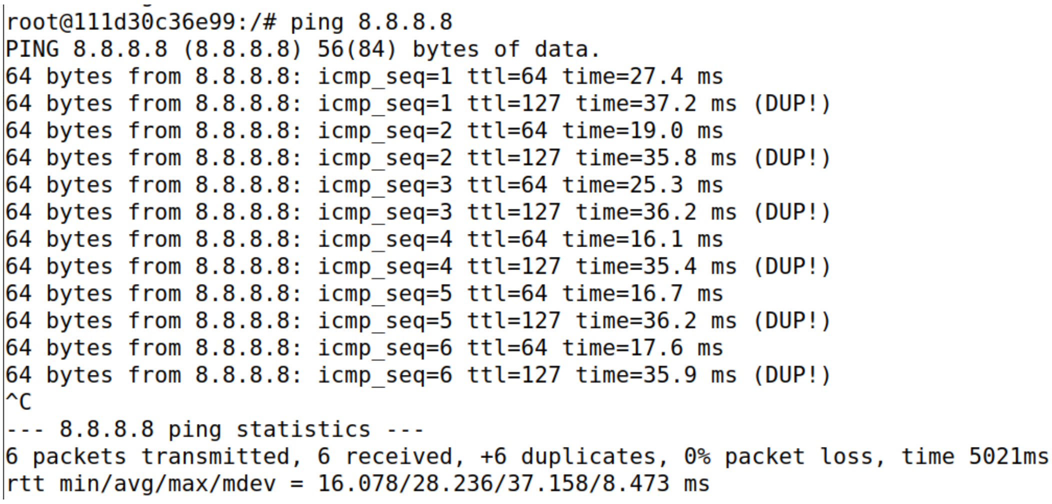
```
root@111d30c36e99:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=64 time=52.3 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=24.8 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=27.8 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=24.3 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=23.9 ms
64 bytes from 1.2.3.4: icmp_seq=6 ttl=64 time=28.4 ms
64 bytes from 1.2.3.4: icmp_seq=7 ttl=64 time=24.7 ms
64 bytes from 1.2.3.4: icmp_seq=8 ttl=64 time=20.2 ms
^C
--- 1.2.3.4 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7022ms
rtt min/avg/max/mdev = 20.210/28.320/52.313/9.368 ms
```

Ping 10.9.0.99：



```
root@111d30c36e99:/# 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
From 10.9.0.5 icmp_seq=7 Destination Host Unreachable
From 10.9.0.5 icmp_seq=8 Destination Host Unreachable
From 10.9.0.5 icmp_seq=9 Destination Host Unreachable
From 10.9.0.5 icmp_seq=10 Destination Host Unreachable
```

Ping 8.8.8.8：



```
root@111d30c36e99:/# 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=27.4 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=37.2 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=19.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=35.8 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=64 time=25.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=127 time=36.2 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=4 ttl=64 time=16.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=127 time=35.4 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=5 ttl=64 time=16.7 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=127 time=36.2 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=6 ttl=64 time=17.6 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=127 time=35.9 ms (DUP!)
^C
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, +6 duplicates, 0% packet loss, time 5021ms
rtt min/avg/max/mdev = 16.078/28.236/37.158/8.473 ms
```

可以发现，在运行 sniff_spoof.py 之后，10.9.0.99 仍然无法 ping 通，但 1.2.3.4 能 ping 通。

因为在运行程序之前，网关 10.9.0.5 告知主机无法通过 ARP 协议找到 1.2.3.4 和 10.0.9.99 对应的 MAC 地址，因此无法 ping 通；而 8.8.8.8 在互联网上存在，因此可以 ping 通。

在运行程序之后，ping 1.2.3.4 需要经过网关 10.9.0.5，网关拦截 ICMP 报文并欺骗主机可以 ping 通 1.2.3.4。而 10.9.0.99 和主机在同一个局域网内，

通过广播 ARP 寻找相应的 MAC 地址，不需要经过网关，因此网关无法欺骗主机，10.9.0.99 仍然 ping 不通。