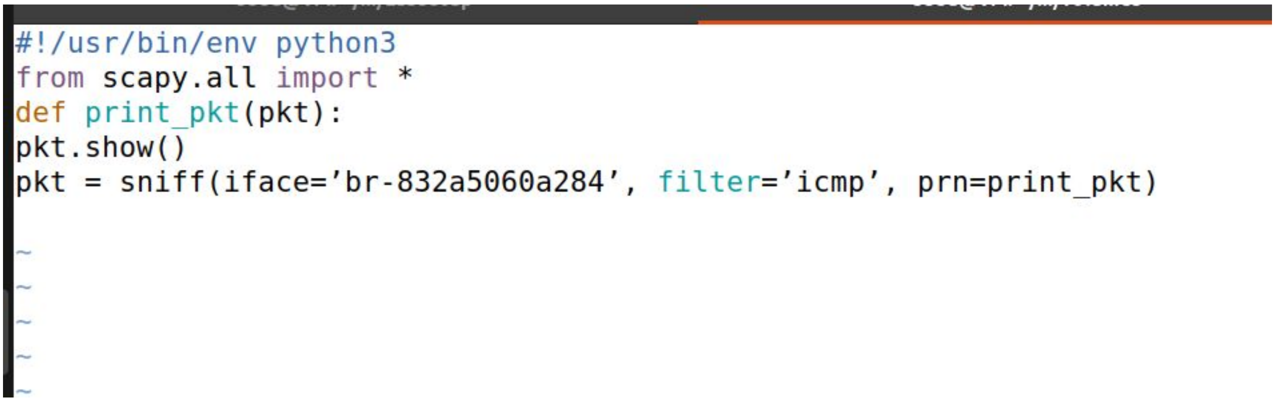
TASK 1.1

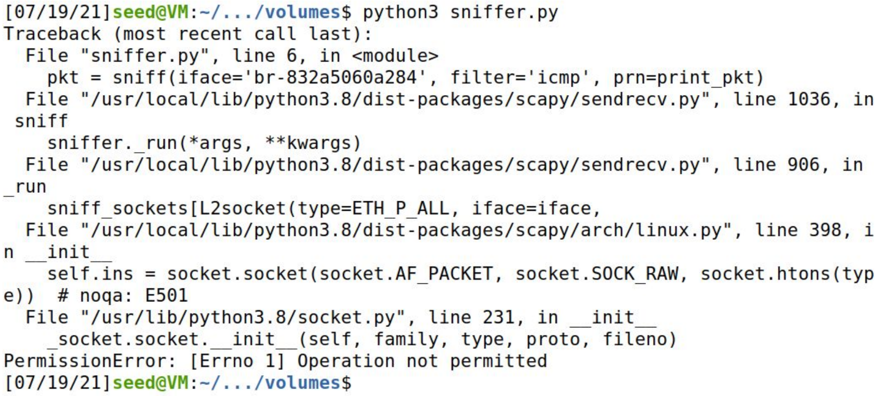
1.1A

先在□□□□□□□□□□□□□□□□□□□□□sniffer.py



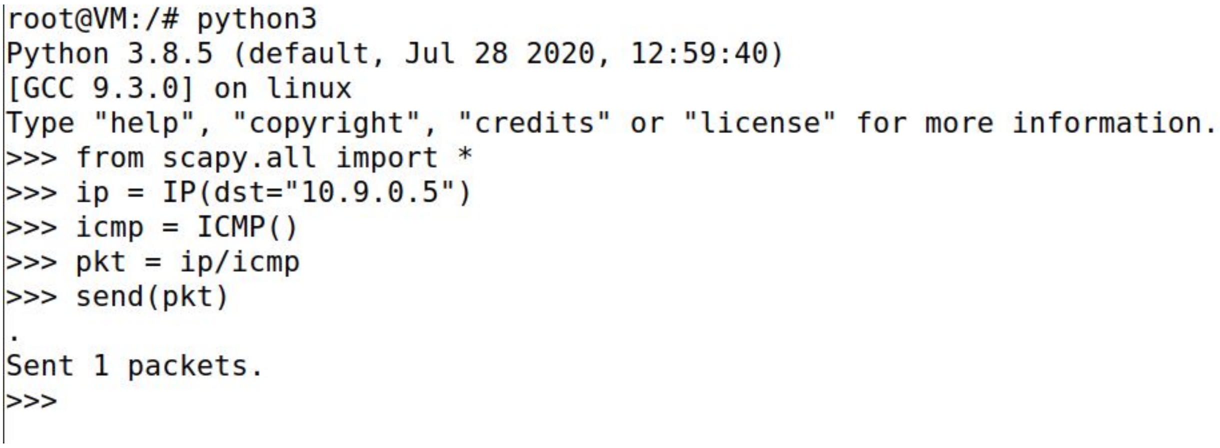
```
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
pkt.show()
pkt = sniff(iface=’br-832a5060a284’, filter=’icmp’, prn=print_pkt)

~
~
~
~
~
```

□□□□运行发现运行失败，因为没有相应权限



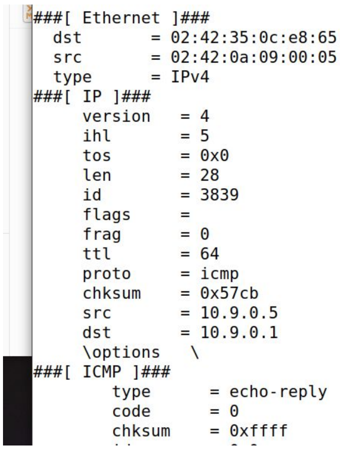
```
[07/19/21]seed@VM:~/.../volumes$ python3 sniffer.py
Traceback (most recent call last):
  File "sniffer.py", line 6, in <module>
    pkt = sniff(iface='br-832a5060a284', filter='icmp', prn=print_pkt)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 1036, in
 sniff
    sniffer._run(*args, **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
[07/19/21]seed@VM:~/.../volumes$
```

root 后运行 sniffer.py，并构造并发送如下报文：



```
root@VM:/# python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from scapy.all import *
>>> ip = IP(dst="10.9.0.5")
>>> icmp = ICMP()
>>> pkt = ip/icmp
>>> send(pkt)
.
Sent 1 packets.
>>>
```

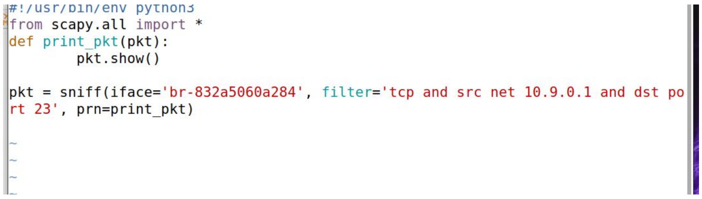
sniffer.py 成功捕获如下信息：



```
###[ Ethernet ]###
  dst       = 02:42:35:0c:e8:65
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 3839
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x57cb
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xffff
```

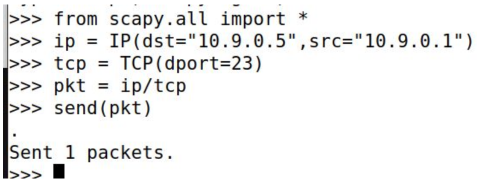
1.1B

捕获特定源地址和目的端口号为 23 的 TCP 报文时，filter为



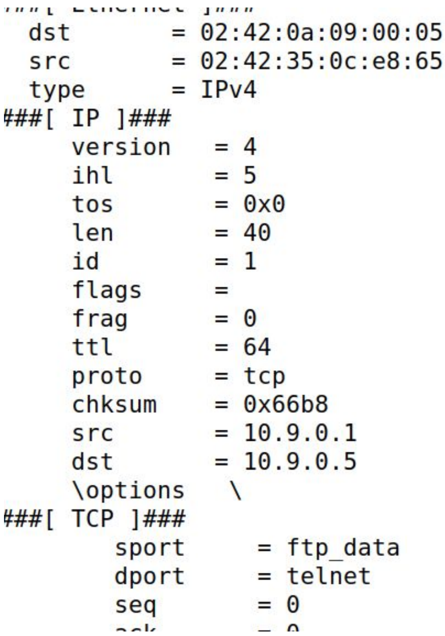
```
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()

pkt = sniff(iface='br-832a5060a284', filter='tcp and src net 10.9.0.1 and dst po
rt 23', prn=print_pkt)

~
~
~
~
```

运行 sniffer.py，构造并发送报文



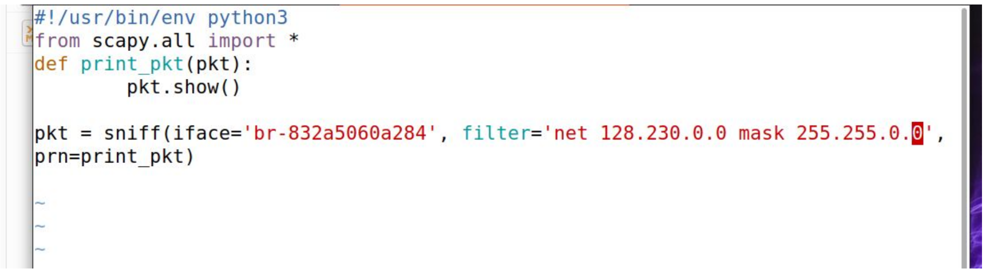
```
>>> from scapy.all import *
>>> ip = IP(dst="10.9.0.5",src="10.9.0.1")
>>> tcp = TCP(dport=23)
>>> pkt = ip/tcp
>>> send(pkt)
.
Sent 1 packets.
>>> 
```

sniffer.py 捕获到的结果如下，其中 dport 端口为 telnet，默认为 23。



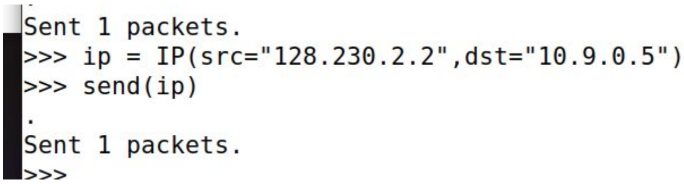
```
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:35:0c:e8:65
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 40
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x66b8
     src       = 10.9.0.1
     dst       = 10.9.0.5
     \options   \
###[ TCP ]###
        sport     = ftp_data
        dport     = telnet
        seq       = 0
```

捕获来自任意子网或去往任意子网的报文，filter为



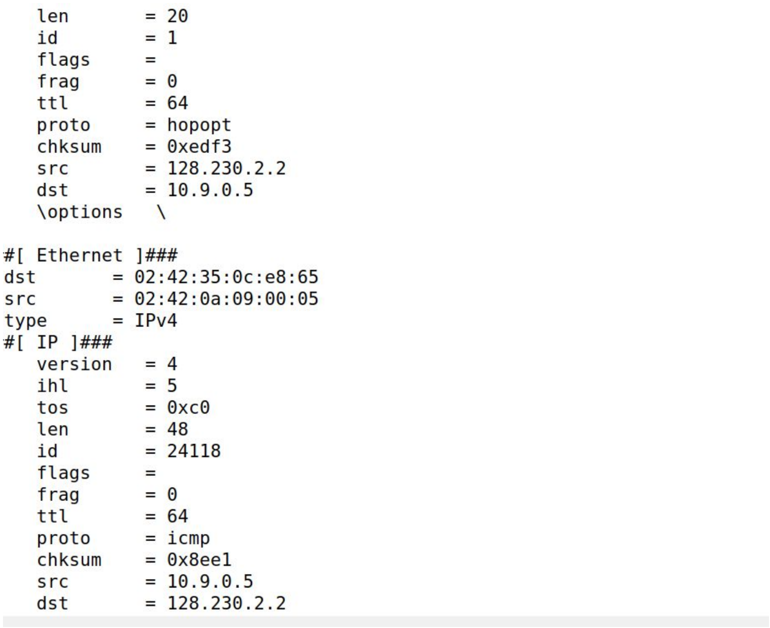
```
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()

pkt = sniff(iface='br-832a5060a284', filter='net 128.230.0.0 mask 255.255.0.0',
prn=print_pkt)

~
~
~
```

构造的报文为：



```
.
Sent 1 packets.
>>> ip = IP(src="128.230.2.2",dst="10.9.0.5")
>>> send(ip)
.
Sent 1 packets.
>>>
```

捕获的结果为：

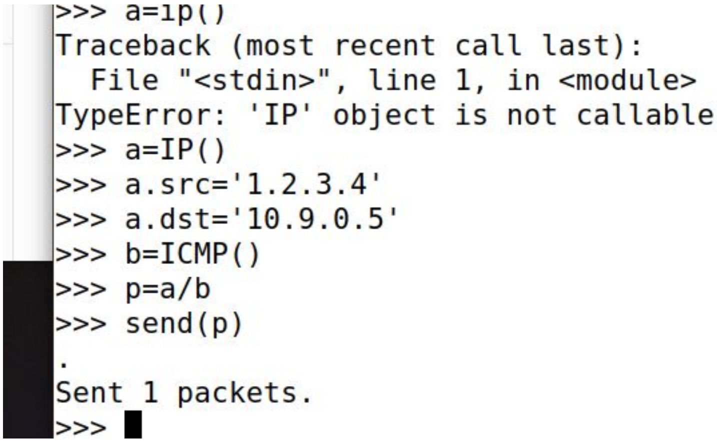


```
   len       = 20
   id        = 1
   flags     =
   frag      = 0
   ttl       = 64
   proto     = hopopt
   chksum    = 0xedf3
   src       = 128.230.2.2
   dst       = 10.9.0.5
   \options   \

#[ Ethernet ]###
dst       = 02:42:35:0c:e8:65
src       = 02:42:0a:09:00:05
type      = IPv4
#[ IP ]###
   version   = 4
   ihl       = 5
   tos       = 0xc0
   len       = 48
   id        = 24118
   flags     =
   frag      = 0
   ttl       = 64
   proto     = icmp
   chksum    = 0x8ee1
   src       = 10.9.0.5
   dst       = 128.230.2.2
```

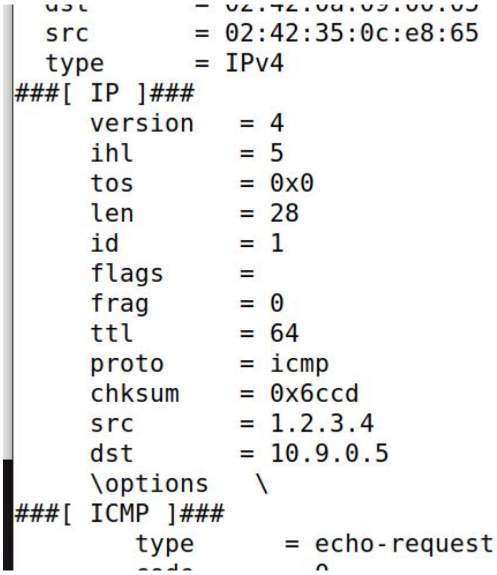
无论是 src 为 128.230.1.1 还是 dst 为 128.230.1.1，都能成功捕获到。

TASK1.2

运行程□ □ □sniffer□ □ □ □ □



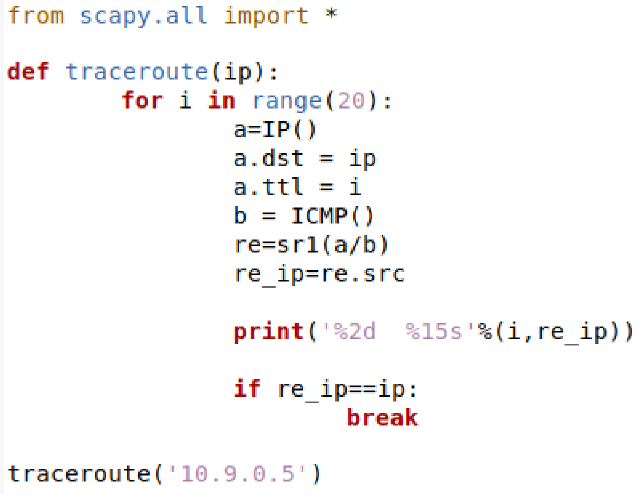
```
>>> a=ip()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'IP' object is not callable
>>> a=IP()
>>> a.src='1.2.3.4'
>>> a.dst='10.9.0.5'
>>> b=ICMP()
>>> p=a/b
>>> send(p)
.
Sent 1 packets.
>>> 
```



```
  dst       = 02:42:0a:09:00:05
  src       = 02:42:35:0c:e8:65
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     = 
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x6ccd
     src       = 1.2.3.4
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
```

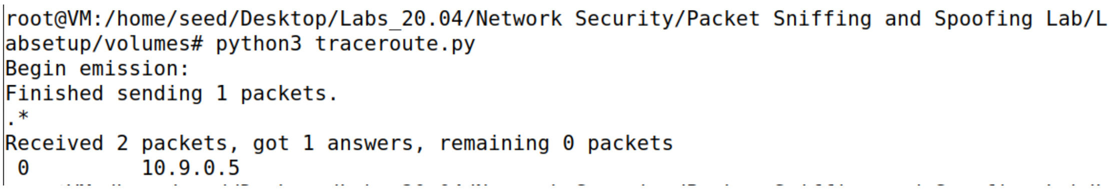
TASK1.3

traceroute.py 代码如下



```
from scapy.all import *

def traceroute(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

traceroute('10.9.0.5')
```

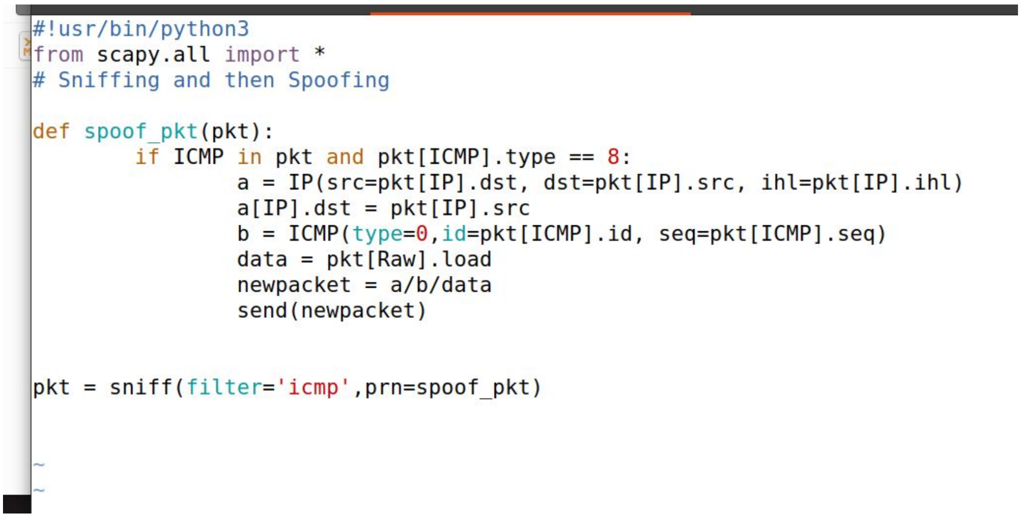
经过一跳到达了目的地址。



```
root@VM:/home/seed/Desktop/Labs_20.04/Network Security/Packet Sniffing and Spoofing Lab/L
absetup/volumes# python3 traceroute.py
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
 0         10.9.0.5
```

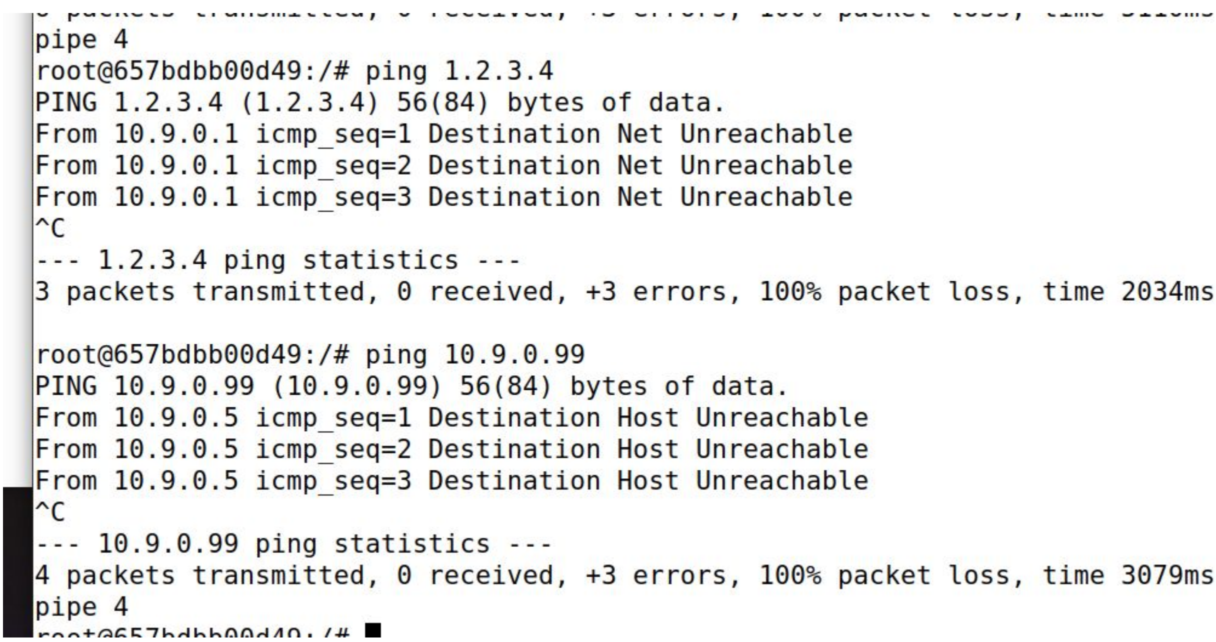
TASK1.4

实验4的代码如下



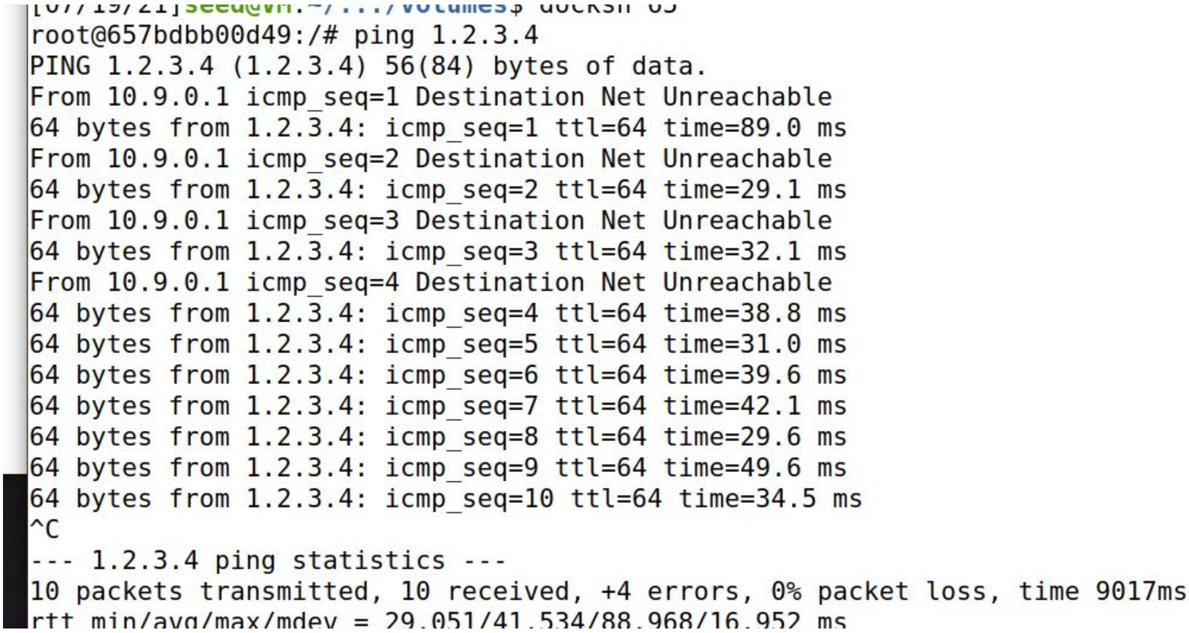
```
#!usr/bin/python3
from scapy.all import *
# Sniffing and then Spoofing

def spoof_pkt(pkt):
        if ICMP in pkt and pkt[ICMP].type == 8:
                a = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
                a[IP].dst = pkt[IP].src
                b = ICMP(type=0,id=pkt[ICMP].id, seq=pkt[ICMP].seq)
                data = pkt[Raw].load
                newpacket = a/b/data
                send(newpacket)


pkt = sniff(filter='icmp',prn=spoof_pkt)


~
~
```

在未运行 4.py 时，ping 三个地址都是不可达

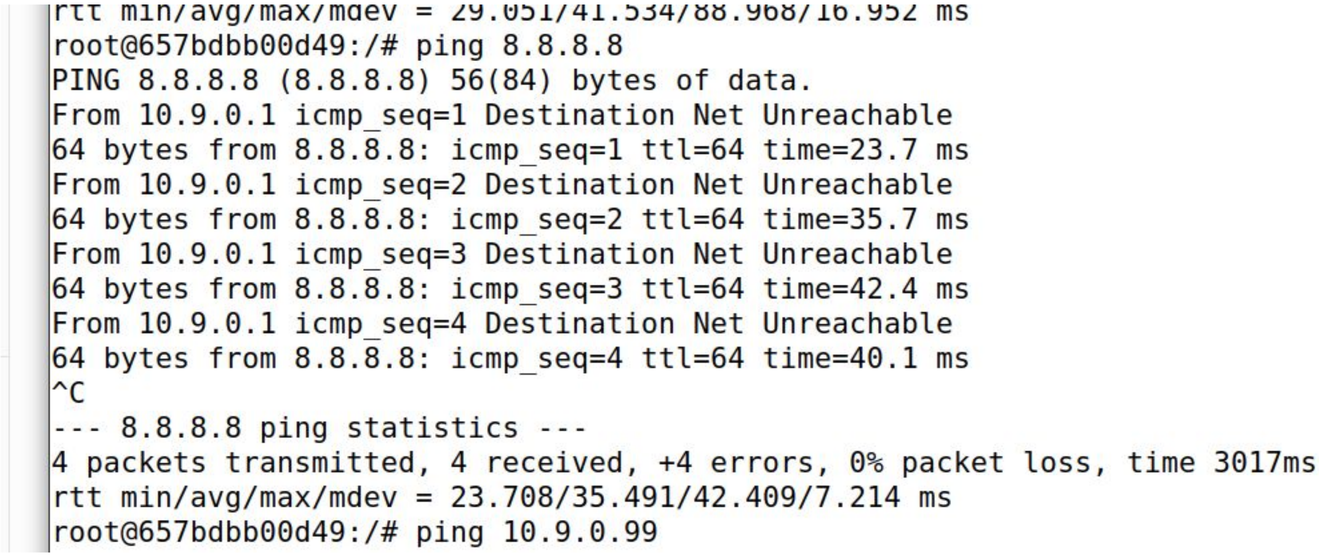


```
pipe 4
root@657bdbb00d49:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
From 10.9.0.1 icmp_seq=1 Destination Net Unreachable
From 10.9.0.1 icmp_seq=2 Destination Net Unreachable
From 10.9.0.1 icmp_seq=3 Destination Net Unreachable
^C
--- 1.2.3.4 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2034ms

root@657bdbb00d49:/# 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
^C
--- 10.9.0.99 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3079ms
pipe 4
root@657bdbb00d49:/# 
```

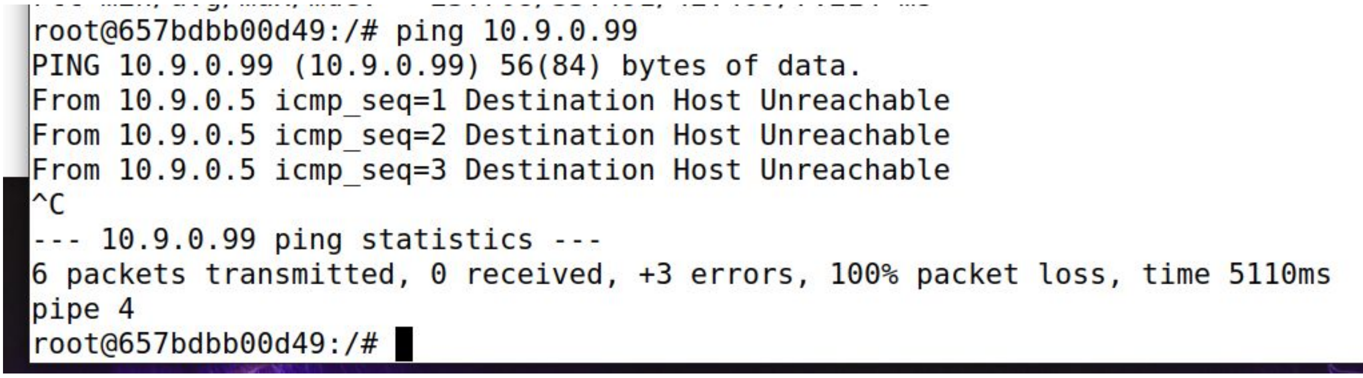
运行 4.py 后再次 ping 这三个地址，其中1.2.3.4，8.8.8.8可达，但是10.9.0.99失败了，这是因为我们在ping一个LAN□□□□□□□□□□□□□□□□□□□□ARP□□□□□□ICMP echo request□□□□□□□□□□



```
[07/19/21]seed@VM:~/.../Volumes$ docksh 65
root@657bdbb00d49:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
From 10.9.0.1 icmp_seq=1 Destination Net Unreachable
64 bytes from 1.2.3.4: icmp_seq=1 ttl=64 time=89.0 ms
From 10.9.0.1 icmp_seq=2 Destination Net Unreachable
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=29.1 ms
From 10.9.0.1 icmp_seq=3 Destination Net Unreachable
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=32.1 ms
From 10.9.0.1 icmp_seq=4 Destination Net Unreachable
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=38.8 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=31.0 ms
64 bytes from 1.2.3.4: icmp_seq=6 ttl=64 time=39.6 ms
64 bytes from 1.2.3.4: icmp_seq=7 ttl=64 time=42.1 ms
64 bytes from 1.2.3.4: icmp_seq=8 ttl=64 time=29.6 ms
64 bytes from 1.2.3.4: icmp_seq=9 ttl=64 time=49.6 ms
64 bytes from 1.2.3.4: icmp_seq=10 ttl=64 time=34.5 ms
^C
--- 1.2.3.4 ping statistics ---
10 packets transmitted, 10 received, +4 errors, 0% packet loss, time 9017ms
rtt min/avg/max/mdev = 29.051/41.534/88.968/16.952 ms
```



```
rtt min/avg/max/mdev = 29.051/41.534/88.968/16.952 ms
root@657bdbb00d49:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
From 10.9.0.1 icmp_seq=1 Destination Net Unreachable
64 bytes from 8.8.8.8: icmp_seq=1 ttl=64 time=23.7 ms
From 10.9.0.1 icmp_seq=2 Destination Net Unreachable
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=35.7 ms
From 10.9.0.1 icmp_seq=3 Destination Net Unreachable
64 bytes from 8.8.8.8: icmp_seq=3 ttl=64 time=42.4 ms
From 10.9.0.1 icmp_seq=4 Destination Net Unreachable
64 bytes from 8.8.8.8: icmp_seq=4 ttl=64 time=40.1 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, +4 errors, 0% packet loss, time 3017ms
rtt min/avg/max/mdev = 23.708/35.491/42.409/7.214 ms
root@657bdbb00d49:/# ping 10.9.0.99
```



```
rtt min/avg/max/mdev = 23.708/35.491/42.409/7.214 ms
root@657bdbb00d49:/# 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
^C
--- 10.9.0.99 ping statistics ---
6 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5110ms
pipe 4
root@657bdbb00d49:/# 
```