Lab 1

57118228 孙志刚

Task 1.1：Sniffing Packets

Task 1.1A

1. 启动 docker，用 ifconfig 查询自己的 iface，为 br-b622afe74271

```
seed@VM: ~/.../Labsetup
[07/09/21]seed@VM:~/.../Labsetup$ dcup
Creating network "net-10.9.0.0" with the default driver
Creating seed-attacker ... done
Creating host-10.9.0.5 ... done
Attaching to seed-attacker, host-10.9.0.5
```

```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ dockps
833f47c4115b  host-10.9.0.5
41d432cd126f  seed-attacker
[07/09/21]seed@VM:~/.../volumes$ ifconfig
br-b622afe74271: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
        inet6 fe80::42:5ff:fec2:8431  prefixlen 64  scopeid 0x20<link>
        ether 02:42:05:c2:84:31  txqueuelen 0  (Ethernet)
```

sniffer.py 如下

```python
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
    pkt.show()
pkt = sniff(iface='br-b622afe74271',filter='icmp',prn=print_pkt)
```

2. 以 root 权限运行 sniffer.py，对主机 IP 进行 ping 命令

```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=64 time=0.051 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=64 time=0.215 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=64 time=0.041 ms
```

```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ chmod a+x sniffer.py
[07/09/21]seed@VM:~/.../volumes$ sudo python3 sniffer.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:05:c2:84:31
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 47430
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x6d4b
     src       = 10.9.0.1
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xbae2
```

3. 以 seed 用户运行 sniffer.py 时，系统会报错

```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ sniffer.py
Traceback (most recent call last):
  File "./sniffer.py", line 5, in <module>
    pkt = sniff(iface='br-b622afe74271',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, i
n __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(typ
e))  # 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
[07/09/21]seed@VM:~/.../volumes$
```

Task 1.1B

1. 只抓取 ICMP 报文，见 Task 1.1A 所示。

2. 捕获任何来自特定 IP，目的端口为 23 的 TCP 数据包。

```python
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
    pkt.show()
pkt = sniff(iface='br-b622afe74271',filter='tcp port 23 and host 10.9.0.5',prn=print_pkt)
```

利用 docksh 获取 host 的 shell，telnet 任意一个 IP 地址建立连接。

```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ dockps
833f47c4115b  host-10.9.0.5
41d432cd126f  seed-attacker
[07/09/21]seed@VM:~/.../volumes$ docksh 83
root@833f47c4115b:/# telnet 1.1.1.1
Trying 1.1.1.1...
telnet: Unable to connect to remote host: Network is unreachable
```

```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ sudo python3 sniffer.py
###[ Ethernet ]###
  dst       = 02:42:05:c2:84:31
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 60
     id        = 12930
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0xfc1a
     src       = 10.9.0.5
     dst       = 1.1.1.1
     \options   \
###[ TCP ]###
        sport     = 36288
        dport     = telnet
        seq       = 624449
        ack       = 0
        dataofs   = 10
        reserved  = 0
        flags     = S
        window    = 64240
        chksum    = 0xc3e
        urgptr    = 0
        options   = [('MSS', 1460), ('SAckOK', b''), ('Timestamp', (2929156180,
0)), ('NOP', None), ('WScale', 7)]
```

3. 捕获发送或接收的子网的报文，这里子网选用 128.230.0.0/16. 代码如下：

```python
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
    pkt.show()
pkt = sniff(filter='net 128.230.0.0/16',prn=print_pkt)
```

```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ ping 128.230.0.1
PING 128.230.0.1 (128.230.0.1) 56(84) bytes of data.
64 bytes from 128.230.0.1: icmp_seq=1 ttl=49 time=317 ms
64 bytes from 128.230.0.1: icmp_seq=2 ttl=49 time=334 ms
64 bytes from 128.230.0.1: icmp_seq=3 ttl=49 time=332 ms
64 bytes from 128.230.0.1: icmp_seq=4 ttl=49 time=337 ms
64 bytes from 128.230.0.1: icmp_seq=5 ttl=49 time=340 ms
64 bytes from 128.230.0.1: icmp_seq=6 ttl=49 time=324 ms
64 bytes from 128.230.0.1: icmp_seq=7 ttl=49 time=322 ms
64 bytes from 128.230.0.1: icmp_seq=8 ttl=49 time=324 ms
64 bytes from 128.230.0.1: icmp_seq=9 ttl=49 time=333 ms
64 bytes from 128.230.0.1: icmp_seq=10 ttl=49 time=249 ms
```

捕获报文如下

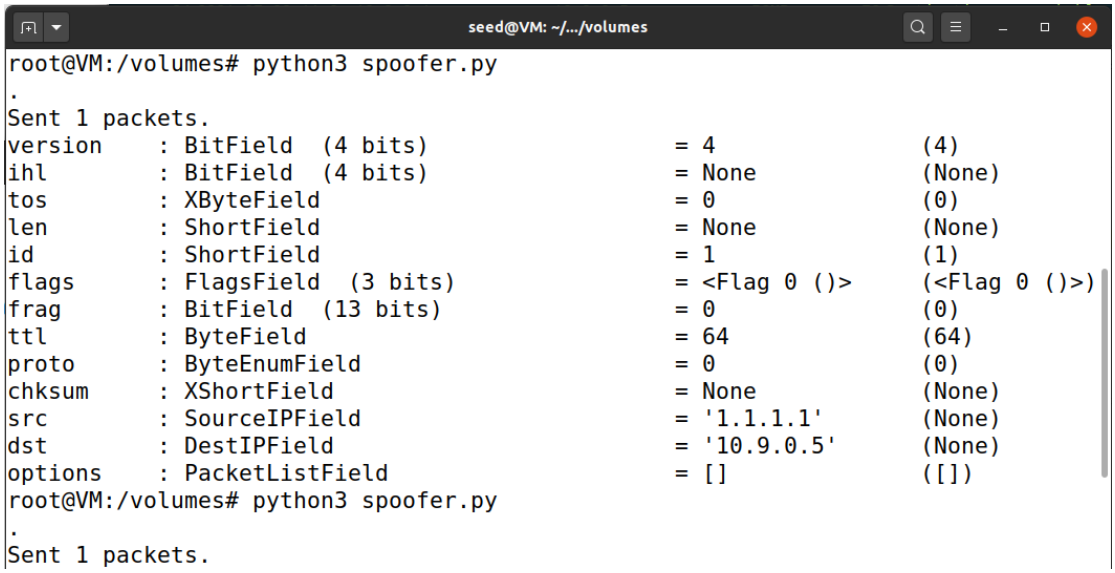
```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ sudo python3 sniffer.py
###[ Ethernet ]###
  dst       = 1a:dc:f0:8b:be:4a
  src       = 08:00:27:08:ba:f1
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 9620
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x43a3
     src       = 192.168.143.226
     dst       = 128.230.0.1
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0x22f2
        id        = 0x3
        seq       = 0x1
###[ Raw ]###
           load      = '\n\\\xe8`\x00\x00\x00\x00\x17z\x0c\x00\x00\x00\x00\x00\x
10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./0
1234567'
```

Task 1.2：Spoofing ICMP Packets

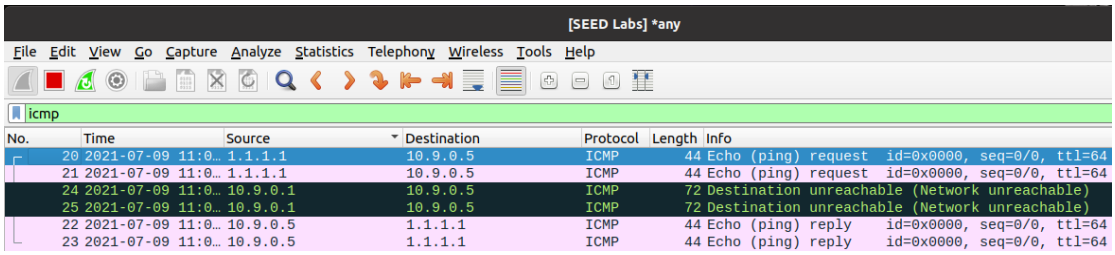
1. 向子网内的一个 IP 发送数据包。代码如下

```
#!usr/bin/evn python3
from scapy.all import *
a =IP()
a.src = '1.1.1.1'
a.dst = '10.9.0.5'
b = ICMP()
p = a/b
send(p)
ls(a)
```

输出如下：



2. 使用 Wireshark 捕获数据包，可发现发送数据包和响应数据包均被捕获。



Task 1.3：Traceroute

向目标 IP 发送 ICMP 数据包，一开始设置 TTL (Time-To-Live)值 为 1，那么发出的 ICMP 数据包在经历一个路由结点后，就会失活被抛弃,我们利用循环，不断增加 TTL 的值，最终使得数据包到达目的地。

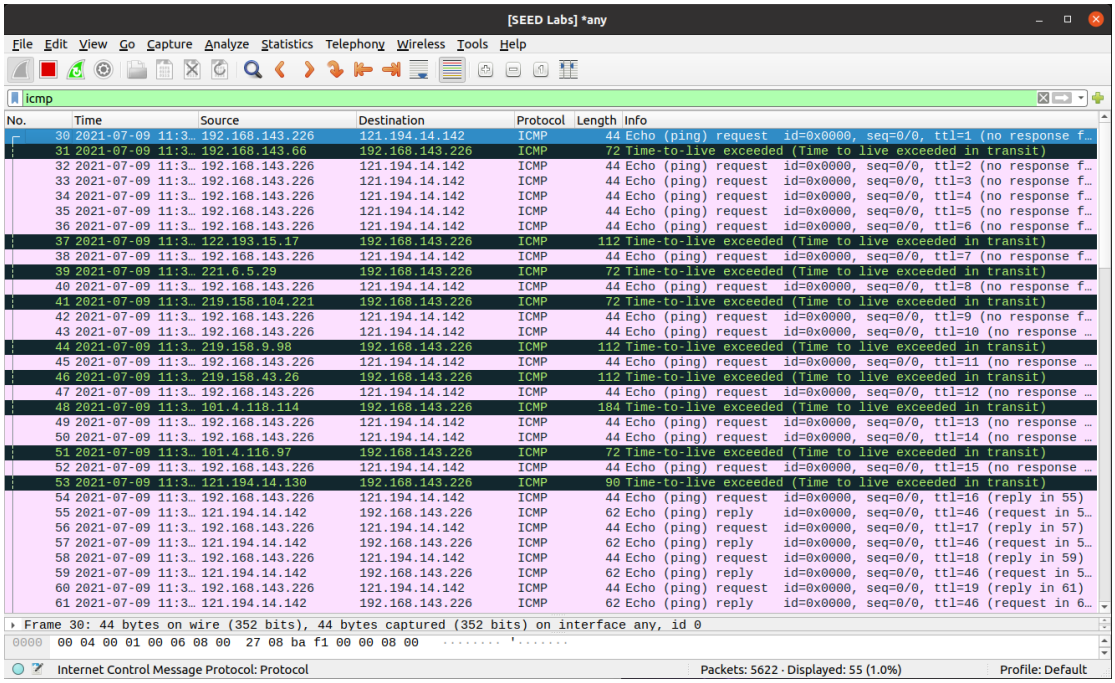
1. 以东南大学官网（www.seu.edu.cn）121.194.14.142 为例，代码如下：

```
#!usr/bin/evn python3
from scapy.all import *
a = IP()
b = ICMP()
a.dst = '121.194.14.142'
for i in range(30):
        a.ttl = i + 1
        p = a / b
        send(p)
```

输出如下：

```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ vi trace.py
[07/09/21]seed@VM:~/.../volumes$ sudo python3 trace.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

2. 使用 Wireshark 捕获数据包，结果如下：



可以看出来，路由为：

192.168.143.226->192.168.143.66->122.193.15.17->……->101.4.116.97

->121.194.14.142

Task 1.4：Sniffing and-then Spoofing

```python
from scapy.all import*

def spoof(pkt):
    a = IP()
    a.src = pkt[IP].dst
    a.dst = '10.9.0.5'
    b = ICMP()
    b.type = 'echo-reply'
    b.code = 0
    b.id = pkt[ICMP].id
    b.seq = pkt[ICMP].seq
    c = pkt[Raw].load
    send(a/b/c)

pkt = sniff(iface='br-b622afe74271',filter='icmp and src host 10.9.0.5',prn=spoof)
```

在未运行 ss.py 时，三个地址都是不可到达的

在虚拟机中运行代码后

```
seed@VM: ~/.../volumes
[07/09/21]seed@VM:~/.../volumes$ sudo python3 ss.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
```

1. ping1.2.3.4 # a non-existing host on the Internet

可以看到当我们 ping 一个网络上不存在的 IP 时，由于伪造报文，我们仍可以接收到响应。

```
seed@VM: ~/.../volumes
root@833f47c4115b:/# 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=52.0 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=37.1 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=17.4 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=27.9 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=17.8 ms
64 bytes from 1.2.3.4: icmp_seq=6 ttl=64 time=23.4 ms
64 bytes from 1.2.3.4: icmp_seq=7 ttl=64 time=18.3 ms
64 bytes from 1.2.3.4: icmp_seq=8 ttl=64 time=16.2 ms
64 bytes from 1.2.3.4: icmp_seq=9 ttl=64 time=19.1 ms
64 bytes from 1.2.3.4: icmp_seq=10 ttl=64 time=25.3 ms
64 bytes from 1.2.3.4: icmp_seq=11 ttl=64 time=17.8 ms
64 bytes from 1.2.3.4: icmp_seq=12 ttl=64 time=16.4 ms
64 bytes from 1.2.3.4: icmp_seq=13 ttl=64 time=14.1 ms
64 bytes from 1.2.3.4: icmp_seq=14 ttl=64 time=13.3 ms
64 bytes from 1.2.3.4: icmp_seq=15 ttl=64 time=15.5 ms
64 bytes from 1.2.3.4: icmp_seq=16 ttl=64 time=17.7 ms
64 bytes from 1.2.3.4: icmp_seq=17 ttl=64 time=18.9 ms
64 bytes from 1.2.3.4: icmp_seq=18 ttl=64 time=21.0 ms
64 bytes from 1.2.3.4: icmp_seq=19 ttl=64 time=17.6 ms
64 bytes from 1.2.3.4: icmp_seq=20 ttl=64 time=25.4 ms
64 bytes from 1.2.3.4: icmp_seq=21 ttl=64 time=28.3 ms
64 bytes from 1.2.3.4: icmp_seq=22 ttl=64 time=19.6 ms
```

2. ping 10.9.0.99 # a non-existing host on the LAN

对于局域网内不存在的主机，先利用 ARP 进行 MAC 地址询问，由于一直得不到结果，所以没有 ICMP 报文，也就不存在报文欺骗。

```
seed@VM: ~/.../volumes
root@833f47c4115b:/# 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
From 10.9.0.5 icmp_seq=8 Destination Host Unreachable
From 10.9.0.5 icmp_seq=9 Destination Host Unreachable
From 10.9.0.5 icmp_seq=10 Destination Host Unreachable
From 10.9.0.5 icmp_seq=11 Destination Host Unreachable
From 10.9.0.5 icmp_seq=12 Destination Host Unreachable
From 10.9.0.5 icmp_seq=13 Destination Host Unreachable
From 10.9.0.5 icmp_seq=14 Destination Host Unreachable
From 10.9.0.5 icmp_seq=15 Destination Host Unreachable
From 10.9.0.5 icmp_seq=16 Destination Host Unreachable
From 10.9.0.5 icmp_seq=17 Destination Host Unreachable
From 10.9.0.5 icmp_seq=18 Destination Host Unreachable
From 10.9.0.5 icmp_seq=19 Destination Host Unreachable
From 10.9.0.5 icmp_seq=20 Destination Host Unreachable
From 10.9.0.5 icmp_seq=21 Destination Host Unreachable
From 10.9.0.5 icmp_seq=22 Destination Host Unreachable
```

3. ping 8.8.8.8 # an existing host on the Internet

对于网络上存在的主机，我们可以看到每个序列号的报文都存在一个重复报文。

由于 8.8.8.8 是网络上存在的主机，故会正常向本机发送报文（TTL=64）,时间较短(TTL=53)的那个报文是伪造的报文

```
seed@VM: ~/.../volumes
root@833f47c4115b:/# 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=53 time=52.8 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=64 time=66.7 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=21.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=29.8 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=64 time=24.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=53 time=36.9 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=4 ttl=64 time=27.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=53 time=27.3 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=5 ttl=64 time=19.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=53 time=30.8 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=6 ttl=64 time=26.3 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=53 time=43.8 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=7 ttl=64 time=18.9 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=53 time=35.6 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=8 ttl=64 time=17.2 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=53 time=39.0 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=9 ttl=64 time=24.3 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=53 time=29.3 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=10 ttl=64 time=20.7 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=53 time=38.3 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=11 ttl=64 time=13.5 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=53 time=31.5 ms (DUP!)
```