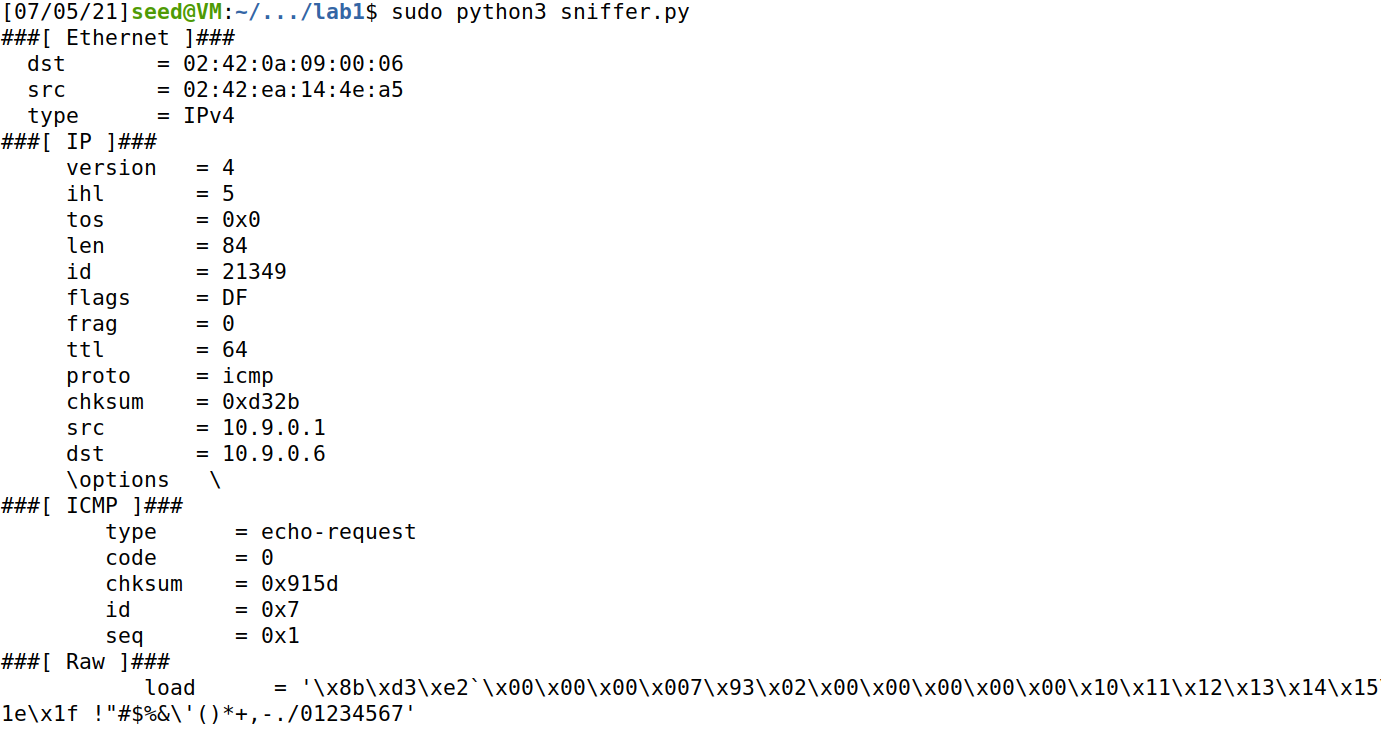
57118133 钟杰

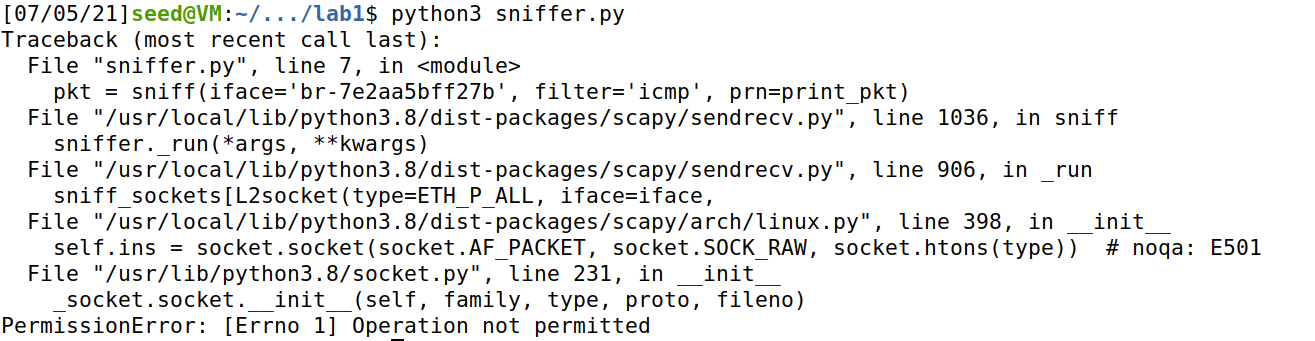
1.1.A

通过设置filter规则为ICMP，可以过滤掉其他的数据包；通过指定prn的参数为显示数据包的函数在终端打印数据包。



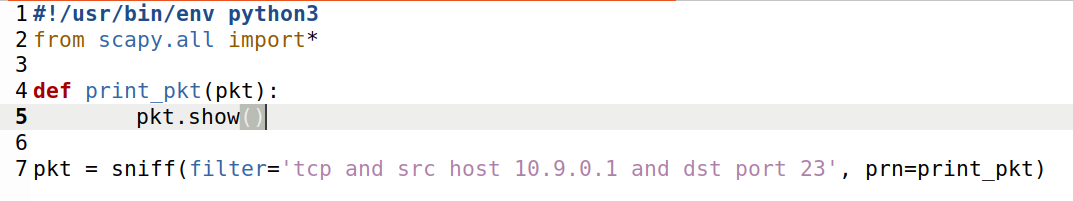
在执行sniffer.py的同时，使用ping发出ICMP数据包。当使用sudo执行脚本可以正常捕获并显示数据包。

当不适用sudo执行脚本时，操作不被允许，根据报错可以看到是因为\_init\_socket.socket.init（）函数的执行需要更高的权限

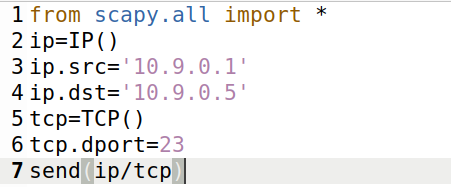


* 1. B (1)与1.1.A相同
  2. B (2)

设置filter的参数为过滤报文为TCP且源地址为10.9.0.1，目的端口为23号

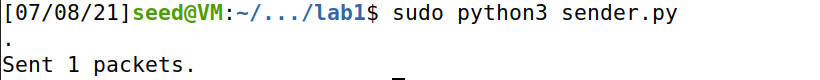


发送一个TCP报文，报文从源地址10.9.0.1发向目的地址10.9.0.5，且目的端口为23号



运行sniffer.py开始嗅探报文

运行sender.py发送报文



Sniffer.py接收到发送的报文并显示

###[ 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 = 40

id = 1

flags =

frag = 0

ttl = 64

proto = tcp

chksum = 0x66b8

src = 10.9.0.1

dst = 10.9.0.5

\options \

###[ TCP ]###

sport = ftp\_data

dport = telnet

seq = 0

ack = 0

dataofs = 5

reserved = 0

flags = S

window = 8192

chksum = 0x7ba0

urgptr = 0

options = []

* 1. B (3)

发送目的子网为10.9.0.0/24的数据包

from scapy.all import \*

send(IP(dst='10.9.0.0/24'))

设置filter参数为目的子网10.9.0.0/24

#!/usr/bin/env python3

from scapy.all import\*

def print\_pkt(pkt):

pkt.show()

pkt = sniff(filter='dst net 10.9.0.0/24',prn=print\_pkt)

接收并显示报文

###[ Ethernet ]###

dst = ff:ff:ff:ff:ff:ff

src = 02:42:91:c6:7c:5a

type = ARP

###[ ARP ]###

hwtype = 0x1

ptype = IPv4

hwlen = 6

plen = 4

op = who-has

hwsrc = 02:42:91:c6:7c:5a

psrc = 10.9.0.1

hwdst = 00:00:00:00:00:00

pdst = 10.9.0.0

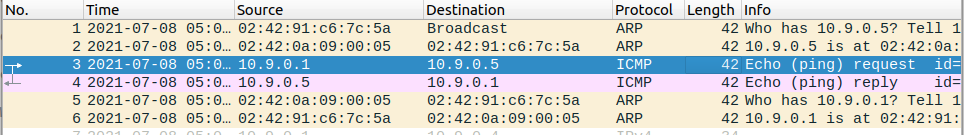
1.2

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 \*

send(IP(dst='10.9.0.5')/ICMP())

用wireshark捕捉



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，并将p发送

from scapy.all import \*

a=IP()

a.dst='10.9.0.5'

b=ICMP()

p=a/b

send(p)

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

1.3

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))

运行结果

$ sudo python3 traceroute.py

Begin emission:

Finished sending 47 packets.

.\*\*\*\*\*\*\*\*\*.^C

Received 11 packets, got 9 answers, remaining 38 packets

4 172.18.0.6 False

5 180.101.49.12 True

6 180.101.49.12 True

7 180.101.49.12 True

8 180.101.49.12 True

9 180.101.49.12 True

10 180.101.49.12 True

11 180.101.49.12 True

12 180.101.49.12 True

1.4

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)

[07/08/21]seed@VM:~/Desktop$ 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=31.5 ms

64 bytes from 1.2.3.4: icmp\_seq=2 ttl=64 time=25.1 ms

64 bytes from 1.2.3.4: icmp\_seq=3 ttl=64 time=25.2 ms

64 bytes from 1.2.3.4: icmp\_seq=4 ttl=64 time=34.0 ms

64 bytes from 1.2.3.4: icmp\_seq=5 ttl=64 time=22.8 ms

^Z

[1]+ Stopped ping 1.2.3.4

[07/08/21]seed@VM:~/Desktop$ ping 8.8.8.8

Ping 8.8.8.8 除了伪造的回复报文，还会接收到真实返回的报文，所以显示报文重复

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=64 time=25.2 ms

64 bytes from 8.8.8.8: icmp\_seq=2 ttl=64 time=22.8 ms

64 bytes from 8.8.8.8: icmp\_seq=2 ttl=112 time=86.8 ms (DUP!)

64 bytes from 8.8.8.8: icmp\_seq=3 ttl=64 time=31.3 ms

64 bytes from 8.8.8.8: icmp\_seq=3 ttl=112 time=83.0 ms (DUP!)

64 bytes from 8.8.8.8: icmp\_seq=4 ttl=64 time=28.3 ms

64 bytes from 8.8.8.8: icmp\_seq=5 ttl=64 time=23.3 ms

64 bytes from 8.8.8.8: icmp\_seq=5 ttl=112 time=124 ms (DUP!)

64 bytes from 8.8.8.8: icmp\_seq=6 ttl=64 time=24.1 ms

^Z

1. + Stopped ping 8.8.8.8

Ping 10.9.0.99 该地址在该网段中，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

[07/08/21]seed@VM:~/Desktop$ ping 10.9.0.99

PING 10.9.0.99 (10.9.0.99) 56(84) bytes of data.

From 10.9.0.1 icmp\_seq=1 Destination Host Unreachable

From 10.9.0.1 icmp\_seq=2 Destination Host Unreachable

From 10.9.0.1 icmp\_seq=3 Destination Host Unreachable

From 10.9.0.1 icmp\_seq=4 Destination Host Unreachable

From 10.9.0.1 icmp\_seq=5 Destination Host Unreachable

From 10.9.0.1 icmp\_seq=6 Destination Host Unreachable

^Z

[3]+ Stopped ping 10.9.0.99