

Packet Sniffing and Spoofing Lab Report

Task1

Task1.1A

我们使用的scapy包的官方介绍为

Scapy runs natively on Linux, and on most Unixes with libpcap and its python wrappers (see [scapy's installation page](#)). The same code base now runs natively on both Python 2 and Python 3.

libpcap的文档中明确指出

Under Linux:

You must be root or the application capturing packets must be installed setuid to root (unless your distribution has a kernel that supports capability bits such as CAP_NET_RAW and code to allow those capability bits to be given to particular accounts and to cause those bits to be set on a user's initial processes when they log in, in which case you must have CAP_NET_RAW in order to capture and CAP_NET_ADMIN to enumerate network devices with, for example, the -D flag).

因此所有基于scapy的程序都必须在root权限下才可以正常工作,反之则不行.

如果没有sudo权限,则会报错:

```
_sock = _realsocket(family, type, proto)
socket.error: [Errno 1] Operation not permitted
100.128.101.100: / 65534$
```

Task1.1B

要求的三种规则分别为:

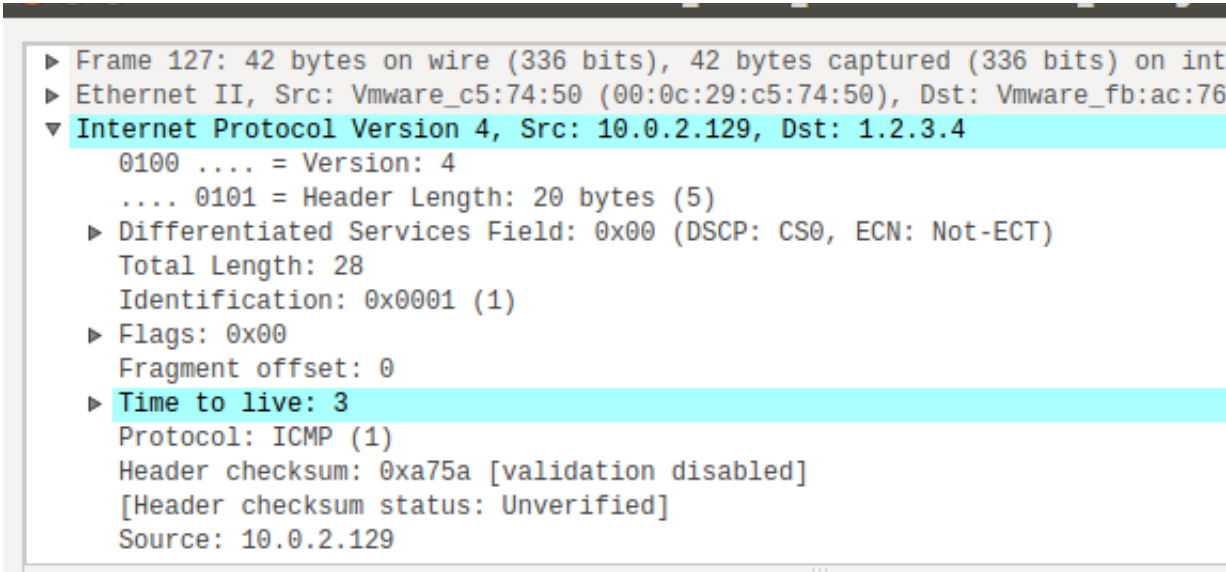
```
pkt=sniff(filter='icmp',prn=print_pkt)
```

```
pkt=sniff(filter='tcp and port 23',prn=print_pkt)
```

```
pkt=sniff(filter='net 128.230',prn=print_pkt)
```

Task1.2

运行指定程序之后用wireshark抓包,发现我们向1.2.3.4发送了ICMP request



Task1.3

由于虚拟机的网络限制,除了前两个包,程序并不能得到结果.尝试了真正的traceroute也得不到结果,通过wireshark观察到了大量UDP的包,而且并没有被识别成DNS流量,怀疑是这个环节出了问题..

程序如下

```
#!/usr/bin/python
from scapy.all import *
for i in range(30):
    a = IP()
    a.dst='180.101.49.11'
    a.ttl=i
    b=ICMP()
    send(a/b)
```

wireshark抓包结果如图

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-09-28 14:14:43.772861896	Vmware_c5:74:50	Broadcast	ARP	42	Who has 10.0.2.2? Tell 10.0.2.129
2	2019-09-28 14:14:43.773048263	Vmware_fb:ac:76	Vmware_c5:74:50	ARP	60	10.0.2.2 is at 00:50:56:fb:ac:76
3	2019-09-28 14:14:43.774240988	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=0 (no response found!)
4	2019-09-28 14:14:43.774375171	10.0.2.2	10.0.2.129	ICMP	78	Time-to-live exceeded (time to live exceeded in transit)
5	2019-09-28 14:14:43.779370189	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (no response found!)
6	2019-09-28 14:14:43.779627191	10.0.2.2	10.0.2.129	ICMP	78	Time-to-live exceeded (time to live exceeded in transit)
7	2019-09-28 14:14:43.783550399	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=2 (no response found!)
8	2019-09-28 14:14:43.790518162	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=3 (no response found!)
9	2019-09-28 14:14:43.793826181	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=4 (no response found!)
10	2019-09-28 14:14:43.797179005	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=5 (no response found!)
11	2019-09-28 14:14:43.807565482	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=6 (no response found!)
12	2019-09-28 14:14:43.812289784	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=7 (no response found!)
13	2019-09-28 14:14:43.819165603	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=8 (no response found!)
14	2019-09-28 14:14:43.825492378	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=9 (no response found!)
15	2019-09-28 14:14:43.829467534	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=10 (no response found!)
16	2019-09-28 14:14:43.833537439	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=11 (no response found!)
17	2019-09-28 14:14:43.836898683	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=12 (no response found!)
18	2019-09-28 14:14:43.841683922	10.0.2.129	180.101.49.11	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=13 (no response found!)

正版traceroute结果如图

```
Terminal
tracert to 36.25.241.250 (36.25.241.250), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.138 ms  0.207 ms  0.206 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
```

如果是在宿主机执行此程序,则可以得到比较好的效果.明显可以看见在得到目标地址回复之前,依次收到了ttl exceeded的包.

5	4.371790	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=0 (no response found!)
6	4.372970	192.168.31.1	192.168.31.72	ICMP	78 Time-to-live exceeded (Time to live exceeded in transit)
7	4.379602	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=1 (no response found!)
8	4.380870	192.168.31.1	192.168.31.72	ICMP	78 Time-to-live exceeded (Time to live exceeded in transit)
9	4.387554	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=2 (no response found!)
10	4.396302	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=3 (no response found!)
11	4.402061	61.152.12.117	192.168.31.72	ICMP	118 Time-to-live exceeded (Time to live exceeded in transit)
12	4.403948	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=4 (no response found!)
13	4.407931	124.74.209.121	192.168.31.72	ICMP	78 Time-to-live exceeded (Time to live exceeded in transit)
14	4.413203	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=5 (no response found!)
15	4.421636	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=6 (no response found!)
16	4.430997	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=7 (no response found!)
17	4.431703	202.97.66.206	192.168.31.72	ICMP	78 Time-to-live exceeded (Time to live exceeded in transit)
18	4.439360	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=8 (no response found!)
19	4.440641	101.95.39.14	192.168.31.72	ICMP	78 Time-to-live exceeded (Time to live exceeded in transit)
20	4.440772	58.213.95.98	192.168.31.72	ICMP	78 Time-to-live exceeded (Time to live exceeded in transit)
21	4.447740	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=9 (no response found!)
22	4.455400	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=10 (no response found!)
23	4.462322	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=11 (no response found!)
24	4.463978	58.213.96.78	192.168.31.72	ICMP	78 Time-to-live exceeded (Time to live exceeded in transit)
25	4.469109	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=12 (no response found!)
26	4.476555	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=13 (no response found!)
27	4.483858	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=14 (no response found!)
28	4.485690	180.101.49.11	192.168.31.72	ICMP	50 Echo (ping) reply id=0x0000, seq=0/0, ttl=51
29	4.491371	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=15 (no response found!)
30	4.492973	180.101.49.11	192.168.31.72	ICMP	50 Echo (ping) reply id=0x0000, seq=0/0, ttl=51
31	4.498407	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=16 (no response found!)
32	4.500499	180.101.49.11	192.168.31.72	ICMP	50 Echo (ping) reply id=0x0000, seq=0/0, ttl=51
33	4.505529	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=17 (no response found!)
34	4.507677	180.101.49.11	192.168.31.72	ICMP	50 Echo (ping) reply id=0x0000, seq=0/0, ttl=51
35	4.513028	192.168.31.72	180.101.49.11	ICMP	42 Echo (ping) request id=0x0000, seq=0/0, ttl=18 (no response found!)

Task1.4

伪造程序如图.原理是嗅探到包之后直接将原地址和目的地址互换,并更改icmp类型为reply即可

```
#!/usr/bin/python
from scapy.all import *
a = IP()
a.show()
def print_pkt(pkt):
    if pkt[IP].src=='10.0.2.132':
        fake=IP()
        fake.dst=pkt[IP].src
```

```

fake.src=pkt[IP].dst
fakeicmp=pkt[ICMP]
fakeicmp.type=0
send(fake/fakeicmp)

#
pkt=sniff(filter='icmp',prn=print_pkt)##1.1A

```

我们知道在中国大陆,www.google.com 是不可能ping通的.但是被攻击之后,目标机器会收到回复.

```

[09/28/19]seed@VM:~$ ping www.google.com
PING www.google.com (74.86.142.55) 56(84) bytes of data.
64 bytes from 37.8e.564a.ip4.static.sl-reverse.com (74.86.142.55): icmp_seq=1 ttl=64 time=12.4 ms
64 bytes from 37.8e.564a.ip4.static.sl-reverse.com (74.86.142.55): icmp_seq=2 ttl=64 time=7.98 ms
64 bytes from 37.8e.564a.ip4.static.sl-reverse.com (74.86.142.55): icmp_seq=3 ttl=64 time=8.56 ms
64 bytes from 37.8e.564a.ip4.static.sl-reverse.com (74.86.142.55): icmp_seq=4 ttl=64 time=5.96 ms
64 bytes from 37.8e.564a.ip4.static.sl-reverse.com (74.86.142.55): icmp_seq=5 ttl=64 time=7.04 ms
^C

```

Task2

Task2.1A

1

第一步:告诉程序需要嗅探那张网卡,得到嗅探目标网卡的handle

第二步:告知程序嗅探的规则并编译应用

第三步:采取适当的方法嗅探(loop,next等等),并调用相应的处理函数

2

根据gdb的分析,程序会在compile这一步失败

```

Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0xb7eea500 in pcap_compile () from /usr/lib/i386-linux-gnu/libpcap.so.0.8

```

3

观察可知,若关闭混杂模式,则只能接收到与自己相关的包.必须打开混杂模式,才能收到同一局域网内其他用户的包

Task2.1B

被监听的两台机器互相ping

```
Received 21 packet containing 0 bytes
got a packet 15
  * Invalid IP header length: 0 bytes
got a packet 16
got an ICMP packet from 10.0.2.131 to 10.0.2.131
got a packet 17
got an ICMP packet from 10.0.2.132 to 10.0.2.132
got a packet 18
got an ICMP packet from 10.0.2.131 to 10.0.2.131
got a packet 19
got an ICMP packet from 10.0.2.132 to 10.0.2.132
got a packet 20
got an ICMP packet from 10.0.2.131 to 10.0.2.131
got a packet 21
got an ICMP packet from 10.0.2.132 to 10.0.2.132
got a packet 22
got a TCP packet from 45708 to 23
got a packet 23
```

telnet登录(23端口)

```
09/28/19] seed@vuln:~/7.1.1/SEEDLAB$ gcc task2.c
ot a packet 2
ot a TCP packet from 60060 to 23
ot a packet 3
ot a packet 4
ot a TCP packet from 60060 to 23
ot a packet 5
ot a TCP packet from 60060 to 23
ot a packet 6
ot a packet 7
ot a packet 8
    * Invalid IP header length: 0 bytes
ot a packet 9
    * Invalid IP header length: 0 bytes
ot a packet 10
ot a packet 11
ot a packet 12
ot a TCP packet from 60060 to 23
ot a packet 13
ot a packet 14
ot a TCP packet from 60060 to 23
ot a packet 15
```

Task2.1c

使用示例程序中的payload,监听所有23端口的讯息,可以得到如下结果.

受害者视角.原理与之前一样.错误的延迟\是因为直接讲收到的icmp包更改类型之后发送,没有改变时间戳.

```
[09/28/19]seed@VM:~/.../SEEDLAB$ ping www.google.com
1. PING www.google.com (31.13.78.65) 56(84) bytes of data.
64 bytes from 31.13.78.65: icmp_seq=5 ttl=64 time=1059 ms
64 bytes from 31.13.78.65: icmp_seq=6 ttl=64 time=1062 ms
64 bytes from 31.13.78.65: icmp_seq=7 ttl=64 time=1084 ms
64 bytes from 31.13.78.65: icmp_seq=8 ttl=64 time=1104 ms
^C
--- www.google.com ping statistics ---
```