

数据包嗅探和欺骗实验

57118217 崔浩为

1.1A

对 icmp 的嗅探 python 脚本

[illegible]

root 权限下开启嗅探

```
IndentationError: expected an indented block
root@VM:/volumes# python3 mycode.py
```

```
###[ Ethernet ]###
dst      = 02:42:0a:09:00:05
src      = 02:42:cb:71:ee:6f
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 57214
```

嗅探结束

```
seed@VM: ~/.../volumes
id      = 25007
flags   =
frag    = 0
ttl     = 64
proto   = icmp
chksum  = 0x4e3
src     = 10.9.0.5
dst     = 10.9.0.1
\options \
###[ ICMP ]###
      type    = echo-reply
      code    = 0
      chksum  = 0x945d
      id      = 0x1
      seq     = 0x1f
###[ Raw ]###
      load    = '\xef\xe4\xe2\x00\x00\x00\x00\xd9i\x01\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#%&\'()*+,-./01234567'
```

无 root 权限结果

```
^Croot@VM:/volumes# su seed
seed@VM:/volumes$ ls
aaa.py baowen.py mycode.py si.py sniffer1.py sniffer2.py
seed@VM:/volumes$ python3 mycode.py
Traceback (most recent call last):
  File "mycode.py", line 7, in <module>
    pkt = sniff(iface='br-f137d4b9e132', 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$
```

1.1. B

a) 仅捕获 ICMP 数据包

嗅探脚本


```

1#!/usr/bin/env python3
2from scapy.all import *
3
4def print_pkt(pkt):
5    pkt.show()
6
7pkt = sniff(iface='br-f137d4b9e132',filter='tcp dst port 23 and (((ip[2:2] -
    ((ip[0]&0xf)<<2)) - ((tcp[12]&0xf0)>>2)) != 0)', prn=print_pkt)

```

结果

```

root@VM:/volumes# python3 sniffer1.py
###[ Ethernet ]###
    dst      = 02:42:e8:25:cb:77
    src      = 02:42:0a:09:00:05
    type     = IPv4
###[ IP ]###
    version  = 4
    ihl      = 5
    tos      = 0x10
    len      = 64
    id       = 19510
    flags    = DF
    frag     = 0
    ttl      = 64
    proto    = tcp
    chksum   = 0xda5a
    src      = 10.9.0.5
    dst      = 10.9.0.1
    \options \
###[ TCP ]###
    sport    = 45902
    .
    .
    .

```

c) 捕获来自或前往特定子网的数据包

伪造报文的脚本

```

^Croot@VM:/volumes# cat aaa.py
from scapy.all import *
a = IP()
a.src='128.230.0.1'
a.dst='10.9.0.5'
b = ICMP()
p = a/b
send(p)
root@VM:/volumes#

```

检测的脚本和结果

```

from scapy.all import *

def print_pkt(pkt):
    pkt.show()

pkt=sniff(iface='br-f137d4b9e132',filter='dst net 128.230.0.0/16',prn=print_pkt)

root@VM:/volumes# python3 sniffer2.py
###[ Ethernet ]###
  dst      = 02:42:3a:e7:c1:1f
  src      = 02:42:0a:09:00:05
  type     = IPv4
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 28
  id       = 24321
  flags    =
  frag     = 0
  ttl      = 64
  proto    = icmp
  chksum   = 0x90eb
  src      = 10.9.0.5
  dst      = 128.230.0.1
  \options \
###[ ICMP ]###
  type     = echo-reply
  code     = 0
  chksum   = 0xffff
  id       = 0x0
  seq      = 0x0

^Croot@VM:/volumes# █

```

1.2 ICMP 数据包欺骗

伪造报文的脚本

```

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

```

发送报文

```
root@VM:/volumes# python3 aaa.py
```

Sent 1 packets.

```
root@VM:/volumes# █
```

抓包结果

5	2021-07-05 20:2...	10.9.0.1	10.9.0.5	ICMP	44 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response ...
6	2021-07-05 20:2...	10.9.0.1	10.9.0.5	ICMP	44 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (reply in 7) ...
7	2021-07-05 20:2...	10.9.0.5	10.9.0.1	ICMP	44 Echo (ping) reply	id=0x0000, seq=0/0, ttl=64 (request in 6)
8	2021-07-05 20:2...	10.9.0.5	10.9.0.1	ICMP	44 Echo (ping) reply	id=0x0000, seq=0/0, ttl=64

1.3 追踪路径

修改网络配置

名称	类型	外部连接	主机连接	DHCP	子网地址
VMnet0	桥接模式	Intel(R) Wireless-AC 9560	-	-	-
VMnet1	仅主机		已连接	已启用	10.168.22.0

脚本程序

```
from scapy.all import *

ans,unans=sr(IP(dst='202.108.22.5', ttl=(4,25))/TCP(flags=0x2))
for snd,rcv in ans:
    print(snd.ttl, rcv.src, isinstance(rcv.payload, TCP))
~
~
~
~
```

路径结果

```
root@VM:/volumes# python3 baowen.py
Begin emission:
Finished sending 22 packets.
.*****.^C
Received 20 packets, got 18 answers, remaining 4 packets
4 172.20.10.4 False
5 183.207.222.5 False
6 183.207.204.209 False
7 221.183.39.245 False
8 221.183.95.38 False
9 219.158.99.149 False
10 219.158.44.29 False
11 10.166.96.24 False
12 202.108.22.5 True
13 220.206.193.14 False
14 10.166.3.48 False
15 202.108.22.5 True
16 202.108.22.5 True
17 125.33.185.94 False
18 202.108.22.5 True
19 202.108.22.5 True
20 202.108.22.5 True
21 202.108.22.5 True
root@VM:/volumes#
```

1.4 嗅探并欺骗

脚本程序

```
from scapy.all import *  
def print_pkt(pkt):  
    a=IP()  
    a.src=pkt[IP].dst  
    a.dst=pkt[IP].src  
    b=ICMP()  
    b.type=0  
    b.id=pkt[ICMP].id  
    b.code=pkt[ICMP].code  
    b.seq=pkt[ICMP].seq  
    str=pkt[Raw].load  
    p=a/b/Raw(str)  
    send(p)  
  
pkt=sniff(iface='br-f137d4b9e132', filter='icmp[icmptype]==icmp-echo', prn=print_p  
kt)
```

"si.py" [readonly] 14L, 305C

14,34 All

Ping 1.2.3.4 的结果

由于路由返回不可抵达 ICMP，所以伪造报文，得到结果

```
[07/06/21] seed@VM:~/.../volumes$ docksh c7
root@c714fca5231c:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
From 10.9.0.1 icmp_seq=1 Destination Net Unreachable
64 bytes from 1.2.3.4: icmp_seq=1 ttl=64 time=57.3 ms
From 10.9.0.1 icmp_seq=2 Destination Net Unreachable
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=23.7 ms
From 10.9.0.1 icmp_seq=3 Destination Net Unreachable
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=17.8 ms
From 10.9.0.1 icmp_seq=4 Destination Net Unreachable
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=19.2 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=17.1 ms
^C
--- 1.2.3.4 ping statistics ---
5 packets transmitted, 5 received, +4 errors, 0% packet
loss, time 4009ms
rtt min/avg/max/mdev = 17.072/27.003/57.265/15.307 ms
root@c714fca5231c:/#
```

Ping 10.9.0.99 的结果

由于是局域网内，所以在局域网内查找，但实际上这个 IP 并不存在，所以会一直处在查找状态，并没有 ICMP 的返回，所以抓不到结果

```

root@c714fca5231c:/# 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
^C
--- 10.9.0.99 ping statistics ---
17 packets transmitted, 0 received, +15 errors, 100% packet loss, time 16361ms
pipe 4
root@c714fca5231c:/# █

```

Ping 8.8.8.8 的结果

由于真实存在所以得到真实和伪造的报文

```

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=72.6 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=49 time=113 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=20.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=49 time=103 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=64 time=19.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=49 time=93.5 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=4 ttl=64 time=15.2 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=49 time=82.3 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=5 ttl=64 time=18.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=49 time=80.9 ms (DUP!)
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, +5 duplicates, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 15.231/61.942/113.206/37.123 ms
root@c714fca5231c:/# █

```
