

LAB 6

57118209 余悦

Task1.A

编译运行相关命令文件：

```
[07/21/21]seed@VM:~/kernel_module$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/kernel_module modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/kernel_module/hello.o
  Building modules, stage 2.
  MODPOST 1 modules
WARNING: modpost: missing MODULE_LICENSE() in /home/seed/kernel_module/hello.o
see include/linux/module.h for more information
  CC [M]  /home/seed/kernel_module/hello.mod.o
  LD [M]  /home/seed/kernel_module/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[07/21/21]seed@VM:~/kernel_module$ sudo insmod hello.ko

[07/21/21]seed@VM:~/kernel_module$ lsmod | grep hello
hello                16384  0
[07/21/21]seed@VM:~/kernel_module$ dmesg
[    0.000000] Linux version 5.4.0-54-generic (buildd@lcy01-amd64-024) (gcc vers
ion 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)) #60-Ubuntu SMP Fri Nov 6 10:37:59 UTC
2020 (Ubuntu 5.4.0-54.60-generic 5.4.65)
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-54-generic root=UUID
=a91f1a43-2770-4684-9fc3-b7abfd786c1d ro quiet splash
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Hygon HygonGenuine
[    0.000000]   Centaur CentaurHauls
[    0.000000]   zhaoxin   Shanghai
[    0.000000] Disabled fast string operations
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point regi
sters'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x008: 'MPX bounds registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x010: 'MPX CSR'
```

Task1.B

1: make seedFilter 代码：

```
seed@VM: ~/.../packet_filter
[07/23/21]seed@VM:~/.../packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Labsetup/Files/packet_f
ilter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M] /home/seed/Labsetup/Files/packet_filter/seedFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M] /home/seed/Labsetup/Files/packet_filter/seedFilter.mod.o
  LD [M] /home/seed/Labsetup/Files/packet_filter/seedFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[07/23/21]seed@VM:~/.../packet_filter$ sudo insmod seedFilter.ko
[07/23/21]seed@VM:~/.../packet_filter$ lsmod | grep seedFilter
seedFilter                16384  0
[07/23/21]seed@VM:~/.../packet_filter$
```

Dig8.8.8.8，发现报文请求被拦截了：

```
[07/23/21]seed@VM:~/.../packet_filter$ dig @8.8.8.8 www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> @8.8.8.8 www.example.com
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached

[07/23/21]seed@VM:~/.../packet_filter$
```

2：修改代码如下：

```
hook1.hook = printInfo;
hook1.hooknum = NF_INET_PRE_ROUTING;
hook1.pf = PF_INET;
hook1.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook1);

hook2.hook = printInfo;
hook2.hooknum = NF_INET_LOCAL_IN;
hook2.pf = PF_INET;
hook2.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook2);

hook3.hook = printInfo;
hook3.hooknum = NF_INET_FORWARD;
hook3.pf = PF_INET;
hook3.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook3);

hook4.hook = printInfo;
hook4.hooknum = NF_INET_LOCAL_OUT;
hook4.pf = PF_INET;
hook4.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook4);

hook5.hook = printInfo;
hook5.hooknum = NF_INET_POST_ROUTING;
hook5.pf = PF_INET;
hook5.priority = NF_IP_PRI_FIRST;
```

```

nf_register_net_hook(&init_net, &hook5);

hook6.hook = blockUDP;
hook6.hooknum = NF_INET_POST_ROUTING;
hook6.pf = PF_INET;
hook6.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook6);

```

Dig 百度地址:

```
[07/24/21]seed@VM:~/.../packet_filter$ dig @180.76.76.76 www.baidu.com
```

```

; <<>> DiG 9.16.1-Ubuntu <<>> @180.76.76.76 www.baidu.com
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached

```

查看 dmesg:

```

[ 260.094031] Registering filters.
[ 265.597995] *** LOCAL_OUT
[ 265.598001] 127.0.0.1 --> 224.0.0.251 (UDP)
[ 265.598028] *** POST_ROUTING
[ 265.598030] 127.0.0.1 --> 224.0.0.251 (UDP)
[ 265.598051] *** PRE_ROUTING
[ 265.598053] 127.0.0.1 --> 224.0.0.251 (UDP)
[ 265.598057] *** LOCAL_IN
[ 265.598220] 127.0.0.1 --> 224.0.0.251 (UDP)
[ 295.842528] *** LOCAL_OUT
[ 295.842533] 127.0.0.1 --> 127.0.0.1 (UDP)
[ 295.842552] *** POST_ROUTING
[ 295.842554] 127.0.0.1 --> 127.0.0.1 (UDP)
[ 295.842573] *** PRE_ROUTING
[ 295.842574] 127.0.0.1 --> 127.0.0.1 (UDP)
[ 295.842577] *** LOCAL_IN
[ 295.842578] 127.0.0.1 --> 127.0.0.1 (UDP)

```

NF_IP_LOCAL_OUT:离开主机

NF_IP_POST_ROUTING:进入不同网络

NF_IP_POST_PRE_ROUTING: 路由决策

NF_IP_LOCAL_IN:发送

NF_IP_FORWARD: 路由转发（这里没有体现出来）

3:

修改有关代码如下：增加两个函数与 hook

```
1 static struct nf_hook_ops hook1, hook2;
2
3 unsigned int telnetFilter(void *priv, struct sk_buff * skb, const struct nf_hook_state *state){
4
5     struct iphdr *iph;
6     struct tcphdr *tcph;
7     iph = ip_hdr(skb);
8     tcph = (void *)iph+iph->ihl*4;
9
10    if(iph->protocol == IPPROTO_TCP && tcph->dest == htons(23))
11    {
12        printk(KERN_INFO "Dropping telnet packet to %d.%d.%d.%d\n",
13              ((unsigned char *)&iph->daddr)[0],
14              ((unsigned char *)&iph->daddr)[1],
15              ((unsigned char *)&iph->daddr)[2],
16              ((unsigned char *)&iph->daddr)[3]);
17        return NF_DROP;
18    }else{
19        return NF_ACCEPT;
20    }
21 }
22
23
24
25 unsigned int ICMPFilter(void *priv, struct sk_buff *skb,
26                        const struct nf_hook_state *state)
27 {
28     struct ethhdr *mac_header=(struct ethhdr *)skb_mac_header(skb);
29     struct iphdr *ip_header=(struct iphdr *)skb_network_header(skb);
30
31     if (!skb) return NF_ACCEPT;
32     if(ip_header->protocol == IPPROTO_ICMP)
33     {
34         printk(KERN_INFO"SRC_MAC:%pM\n",mac_header->h_source);
35         printk(KERN_INFO"DST_MAC:%pM\n",mac_header->h_dest);
36         printk(KERN_INFO"SRC_IP:%p14\n",&ip_header->saddr);
37         printk(KERN_INFO"DST_IP:%p14\n",&ip_header->daddr);
38
39         printk(KERN_INFO"the protocol ICMP(%d) is dropped...\n",IPPROTO_ICMP);
40         return NF_DROP;
41     }
42
43     return NF_ACCEPT;
44 }
45
```

```

56
57 int registerFilter(void) {
58     printk(KERN_INFO "Registering filters.\n");
59
60     hook1.hook = telnetFilter;
61     hook1.hooknum = NF_INET_LOCAL_IN;
62     hook1.pf = PF_INET;
63     hook1.priority = NF_IP_PRI_FIRST;
64     nf_register_net_hook(&init_net, &hook1);
65
66     hook2.hook = ICMPFilter;
67     hook2.hooknum = NF_INET_LOCAL_OUT;
68     hook2.pf = PF_INET;
69     hook2.priority = NF_IP_PRI_FIRST;
70     nf_register_net_hook(&init_net, &hook2);
71
72     return 0;
73 }
74
75 void removeFilter(void) {
76     printk(KERN_INFO "The filters are being removed.\n");
77     nf_unregister_net_hook(&init_net, &hook1);
78     nf_unregister_net_hook(&init_net, &hook2);
79 }
80
81 module_init(registerFilter);
82 module_exit(removeFilter);
83

```

在 docker 中的 10.9.0.5 中 ping 和 Telnet 10.9.0.1 均无法实现:

```

root@0ed0f6469752:/# ping 10.9.0.1
PING 10.9.0.1 (10.9.0.1) 56(84) bytes of data.
^C
--- 10.9.0.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3056ms

root@0ed0f6469752:/# █

root@0ed0f6469752:/# telnet 10.9.0.1
Trying 10.9.0.1...

```

观察 dmesg 发现防火墙均过滤掉了:

```

[ 339.133548] Dropping telnet packet to 10.9.0.1
[ 340.142101] Dropping telnet packet to 10.9.0.1
[ 342.157115] Dropping telnet packet to 10.9.0.1
[ 346.380560] Dropping telnet packet to 10.9.0.1
[ 354.581901] Dropping telnet packet to 10.9.0.1

```

Task2.A

进入容器:

```
[07/24/21]seed@VM:~/.../Labsetup$ dockps
7aab7c3c2ef3  host1-192.168.60.5
15b198c38213  seed-router
0dbbf2516a95  hostA-10.9.0.5
b37082ea62ab  host3-192.168.60.7
7427a51492c6  host2-192.168.60.6
[07/24/21]seed@VM:~/.../Labsetup$ docksh 15
root@15b198c38213:/# iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@15b198c38213:/# iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCE
PT
root@15b198c38213:/# iptables -P OUTPUT DROP
root@15b198c38213:/# iptables -P INPUT DROP
root@15b198c38213:/#
```

根据手册给的命令配置：

发现均无法 ping 和 Telnet 到路由器：

```
root@0dbbf2516a95:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
^C
--- 192.168.60.11 ping statistics ---
46 packets transmitted, 0 received, 100% packet loss, time 46079ms

root@0dbbf2516a95:/# telnet 10.9.0.11
Trying 10.9.0.11...
^C
```

将手册命令交换修改之后：

```
root@15b198c38213:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@15b198c38213:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEP
T
root@15b198c38213:/# iptables -P OUTPUT DROP
root@15b198c38213:/#
root@15b198c38213:/# iptables -P INPUT DROP
root@15b198c38213:/#
```

发现可以 ping 通路由器但是 Telnet 不成功：

```
root@0dbbf2516a95:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.135 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.255 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.208 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.159 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.215 ms
^C
--- 10.9.0.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4083ms
rtt min/avg/max/mdev = 0.135/0.194/0.255/0.042 ms
root@0dbbf2516a95:/#

root@0dbbf2516a95:/# telnet 10.9.0.11
Trying 10.9.0.11...
```

取消上述规则：

```
root@15b198c38213:/# iptables -F
root@15b198c38213:/# iptables -P OUTPUT ACCEPT
root@15b198c38213:/# iptables -P INPUT ACCEPT
root@15b198c38213:/#
```

发现均可以 ping 通和 Telnet 路由器:

```
root@0dbbf2516a95:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.252 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.201 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.175 ms
^Z
[1]+  Stopped                  ping 10.9.0.11
root@0dbbf2516a95:/# telnet 10.9.0.11
Trying 10.9.0.11...
Connected to 10.9.0.11.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
15b198c38213 login: █
```

Task2.B

配置 Iptables 规则如下:

```
root@15b198c38213:/# iptables -P OUTPUT DROP
root@15b198c38213:/# iptables -P INPUT DROP
root@15b198c38213:/# iptables -A FORWARD -p icmp --icmp-type echo-request -o eth
1 -j DROP
root@15b198c38213:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@15b198c38213:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEP
T
root@15b198c38213:/#
```

1.外部主机不可以 ping 通内部主机:

```
root@0dbbf2516a95:/# ping 192.168.60.6
PING 192.168.60.6 (192.168.60.6) 56(84) bytes of data.
^C
--- 192.168.60.6 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7171ms

root@0dbbf2516a95:/# █
```

2.外部主机可以 ping 通路由器两个接口:


```

root@0dbbf2516a95:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.105 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.219 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.149 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.160 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.215 ms
64 bytes from 10.9.0.11: icmp_seq=6 ttl=64 time=0.196 ms
^C
--- 10.9.0.11 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5101ms
rtt min/avg/max/mdev = 0.105/0.174/0.219/0.040 ms
root@0dbbf2516a95:/# █

root@0dbbf2516a95:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
64 bytes from 192.168.60.11: icmp_seq=1 ttl=64 time=0.315 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.188 ms
64 bytes from 192.168.60.11: icmp_seq=3 ttl=64 time=0.211 ms
^C
--- 192.168.60.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2059ms
rtt min/avg/max/mdev = 0.188/0.238/0.315/0.055 ms
root@0dbbf2516a95:/# █

```

3.内网主机可以 ping 通外部主机:

```

[07/24/21]seed@VM:~/.../Labsetup$ docksh 7a
root@7aab7c3c2ef3:/# 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=63 time=0.415 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.131 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.352 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.210 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=63 time=0.313 ms
^C
--- 10.9.0.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4094ms
rtt min/avg/max/mdev = 0.131/0.284/0.415/0.101 ms
root@7aab7c3c2ef3:/#

```

4.内外无法互相 Telnet:

```

root@0dbbf2516a95:/# telnet 10.9.0.11
Trying 10.9.0.11...
█

```

Task2.C:

配置 iptables 有关规则如下:


```
root@15b198c38213:/# iptables -P OUTPUT DROP
root@15b198c38213:/# iptables -P FORWARD DROP
root@15b198c38213:/# iptables -P INPUT DROP
root@15b198c38213:/# iptables -A FORWARD -p tcp -m tcp --dport 23 -d 192.168.60.5 -j ACCEPT
root@15b198c38213:/# iptables -A FORWARD -p tcp --sport 23 -s 192.168.60.5 -j ACCEPT
root@15b198c38213:/# iptables -A FORWARD -i eth1 -o eth1 -j ACCEPT
root@15b198c38213:/#
```

1.外部主机能够 Telnet 192.168.60.5，不可以 Telnet 到其内部主机：

```
root@0dbbf2516a95:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
7aab7c3c2ef3 login:
```

```
root@0dbbf2516a95:/# telnet 192.168.60.6
Trying 192.168.60.6...
```

2.外部主机不可 ping 通内部 server:

```
root@0dbbf2516a95:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4094ms

root@0dbbf2516a95:/#
```

3. 内部主机可以互相 ping 通和 Telnet:

```

root@7aab7c3c2ef3:/# ping 192.168.60.6
PING 192.168.60.6 (192.168.60.6) 56(84) bytes of data.
64 bytes from 192.168.60.6: icmp_seq=1 ttl=64 time=0.140 ms
64 bytes from 192.168.60.6: icmp_seq=2 ttl=64 time=0.170 ms
64 bytes from 192.168.60.6: icmp_seq=3 ttl=64 time=0.154 ms
64 bytes from 192.168.60.6: icmp_seq=4 ttl=64 time=0.228 ms
64 bytes from 192.168.60.6: icmp_seq=5 ttl=64 time=0.179 ms
^Z
[1]+  Stopped                  ping 192.168.60.6
root@7aab7c3c2ef3:/# ping 192.168.60.7
PING 192.168.60.7 (192.168.60.7) 56(84) bytes of data.
64 bytes from 192.168.60.7: icmp_seq=1 ttl=64 time=0.139 ms
64 bytes from 192.168.60.7: icmp_seq=2 ttl=64 time=0.144 ms
64 bytes from 192.168.60.7: icmp_seq=3 ttl=64 time=0.180 ms
64 bytes from 192.168.60.7: icmp_seq=4 ttl=64 time=0.228 ms
^C
--- 192.168.60.7 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3078ms
rtt min/avg/max/mdev = 0.139/0.172/0.228/0.035 ms
root@7aab7c3c2ef3:/# █

root@7aab7c3c2ef3:/# telnet 192.168.60.6
Trying 192.168.60.6...
Connected to 192.168.60.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
7427a51492c6 login: ^CConnection closed by foreign host.
root@7aab7c3c2ef3:/# telnet 192.168.60.7
Trying 192.168.60.7...
Connected to 192.168.60.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
b37082ea62ab login: Connection closed by foreign host.
root@7aab7c3c2ef3:/#

```

4.内部主机不可以 Telnet 和 ping 通外部主机:

```

root@7aab7c3c2ef3:/# telnet 10.9.0.5
Trying 10.9.0.5...
^C
root@7aab7c3c2ef3:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
^Z
[2]+  Stopped                  ping 10.9.0.5
root@7aab7c3c2ef3:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
^C
--- 10.9.0.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3053ms

root@7aab7c3c2ef3:/# █

```

Task3.A

Ping 并执行 conntrack -L，观察得到 ICMP 连接时间默认持续 29 秒：

```
root@0dbbf2516a95:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=0.186 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.086 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.122 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.135 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.130 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.087 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.103 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.131 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.085 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.095 ms

root@15b198c38213:/# conntrack -L
icmp      1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=65 src=192.168.60.5
dst=10.9.0.5 type=0 code=0 id=65 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@15b198c38213:/#
```

执行如下命令，查看过程持续了 24 秒：

```
root@0dbbf2516a95:/# nc -u 192.168.60.5 9090
yyyy
yyyyyy
yyuuuuuu
```

```
root@7aab7c3c2ef3:/# nc -lu 9090
yyyy
yyyyyy
yyuuuuuu
```

```
root@15b198c38213:/# conntrack -L
udp      17 24 src=10.9.0.5 dst=192.168.60.5 sport=42930 dport=9090 [UNREPLIED]
src=192.168.60.5 dst=10.9.0.5 sport=9090 dport=42930 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@15b198c38213:/#
```

TCP 连接持续了 431999 秒：

Task3.B

在 2.c 的规则基础上加上下面两条规则：

```
root@15b198c38213:/# iptables -A FORWARD -p tcp -i eth1 --syn -m conntrack --ctstate NEW -j ACCEPT
root@15b198c38213:/# iptables -A FORWARD -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
root@15b198c38213:/#
```

可以发现内部主机可以 Telnet 外部主机：

```
root@7aab7c3c2ef3:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0dbbf2516a95 login: ^CConnection closed by foreign host.
root@7aab7c3c2ef3:/#
```

TASK 4

配置 iptables 的 limit 规则：

只写一条：

```
root@15b198c38213:/# iptables -A FORWARD -s 10.9.0.5 -m limit \
> --limit 10/minute --limit-burst 5 -j ACCEPT
root@15b198c38213:/#
```

可以观察到 ping 通了，且没有丢包：

```
64 bytes from 192.168.60.5: icmp_seq=33 ttl=63 time=0.088 ms
64 bytes from 192.168.60.5: icmp_seq=34 ttl=63 time=0.086 ms
^C
--- 192.168.60.5 ping statistics ---
34 packets transmitted, 34 received, 0% packet loss, time 33797ms
rtt min/avg/max/mdev = 0.073/0.101/0.151/0.021 ms
root@0dbbf2516a95:/#
```

增加第二条规则：

```
root@15b198c38213:/# iptables -A FORWARD -s 10.9.0.5 -j DROP
root@15b198c38213:/#
```

可以观察到出现丢包现象，流量限制成功：

```

64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.138 ms
64 bytes from 192.168.60.5: icmp_seq=25 ttl=63 time=0.084 ms
64 bytes from 192.168.60.5: icmp_seq=31 ttl=63 time=0.102 ms
64 bytes from 192.168.60.5: icmp_seq=37 ttl=63 time=0.147 ms
64 bytes from 192.168.60.5: icmp_seq=43 ttl=63 time=0.094 ms
64 bytes from 192.168.60.5: icmp_seq=48 ttl=63 time=0.117 ms
64 bytes from 192.168.60.5: icmp_seq=54 ttl=63 time=0.093 ms
64 bytes from 192.168.60.5: icmp_seq=60 ttl=63 time=0.120 ms
^C
--- 192.168.60.5 ping statistics ---
63 packets transmitted, 15 received, 76.1905% packet loss, time 63484ms
rtt min/avg/max/mdev = 0.080/0.107/0.147/0.020 ms
root@0dbbf2516a95:/#

```

所以第二条规则是必须要的，才能够限制流量，只有第二条在第一条的基础上才能继续匹配从而 drop 一些报文。

TASK 5

配置如下规则：

```

root@15b198c38213:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m\
> statistic --mode nth --every 3 --packet 0 -j DNAT --to-destination 192.168.60.
5:8080
root@15b198c38213:/#

```

每三次输出才会在其中显示出来，效果如下：

```

root@0dbbf2516a95:/# echo yy | nc -u 10.9.0.11 8080
^C
root@0dbbf2516a95:/# echo yy | nc -u 10.9.0.11 8080
root@0dbbf2516a95:/#
root@0dbbf2516a95:/# echo yy | nc -u 10.9.0.11 8080
root@0dbbf2516a95:/# echo yy | nc -u 10.9.0.11 8080
^C
root@0dbbf2516a95:/#

```

```

~
root@7aab7c3c2ef3:/# nc -luk 8080
yy
yy

```


修改为以 0.5 的概率的随机形式：

```

root@15b198c38213:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statis
tic --mode random --probability 0.5 -j DNAT --to-destination 192.168.60.5:8080
root@15b198c38213:/#

```

```
root@0dbbf2516a95:/# echo yy | nc -u 10.9.0.11 8080
^C
root@0dbbf2516a95:/#
root@0dbbf2516a95:/# echo yy | nc -u 10.9.0.11 8080
root@0dbbf2516a95:/# echo yy | nc -u 10.9.0.11 8080
^C
root@0dbbf2516a95:/# echo yy | nc -u 10.9.0.11 8080
root@0dbbf2516a95:/#
```



```
root@7aab7c3c2ef3:/# nc -luk 8080
```

```
yy
```

```
yy
```

