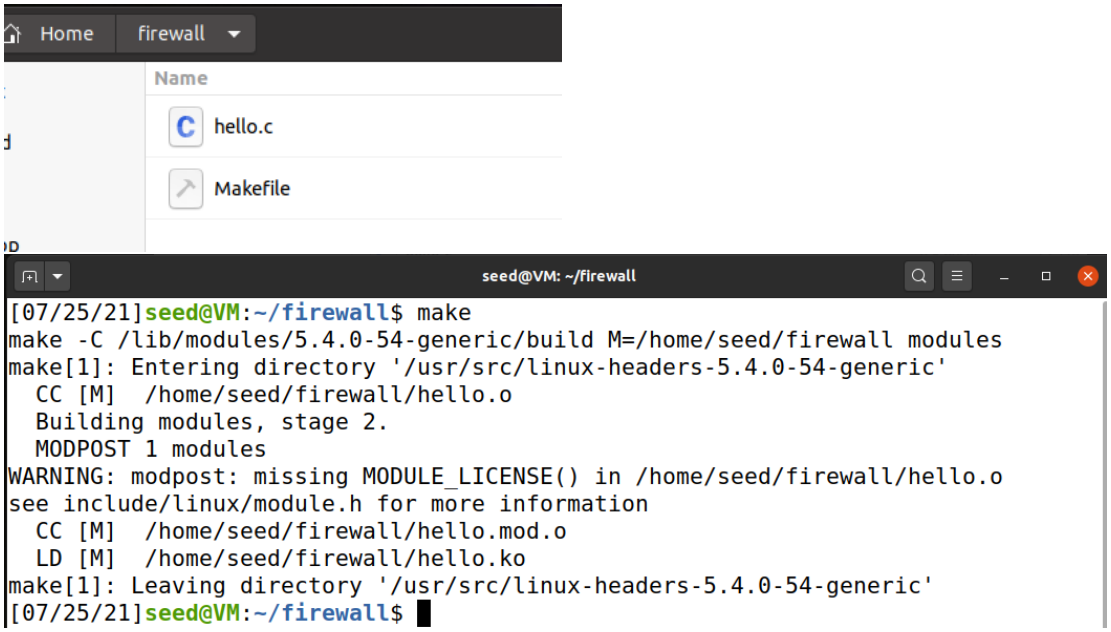
Firewall Exploration Lab

57118107 任子悦

Task 1: Implementing a Simple Firewall

Task 1.A: Implement a Simple Kernel Module

在 seed 创建文件夹 fierwall，将 hello.c 和 Makefile 拷贝至文件夹，输入命令 make 进行编译：



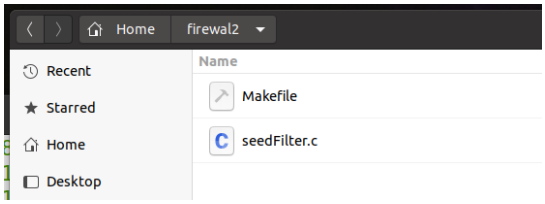
编译成功后测试以下命令：

```
[07/25/21]seed@VM:~/firewall$ sudo insmod hello.ko
[07/25/21]seed@VM:~/firewall$ lsmod | grep hello
hello                  16384  0
[07/25/21]seed@VM:~/firewall$ 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
```

Task 1.B: Implement a Simple Firewall Using Netfilter

1.

将两个文件复制到新的文件夹进行 Make 编译



```
seed@VM: ~/firewal2
[07/26/21]seed@VM:~/firewal2$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/firewal2 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/firewal2/seedFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/seed/firewal2/seedFilter.mod.o
  LD [M]  /home/seed/firewal2/seedFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[07/26/21]seed@VM:~/firewal2$
```

先不载入内核模块，输入命令 dig @8.8.8.8 www.example.com，得到如下结果：

```
[07/26/21]seed@VM:~/firewal2$ 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
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35290
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        20185   IN      A       93.184.216.34

;; Query time: 260 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Mon Jul 26 02:32:05 EDT 2021
;; MSG SIZE  rcvd: 60
```

载入模块可见防火墙起效：

```
[07/26/21]seed@VM:~/firewal2$ sudo insmod seedFilter.ko
[07/26/21]seed@VM:~/firewal2$ 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
```

2.

修改 seedFilter.c 部分，将 printinfo 挂到所有 hook 上：

```
static struct nf_hook_ops hook1, hook2,hook3,hook4,hook5;

unsigned int printInfo(void *priv, struct sk_buff *skb,
                 const struct nf_hook_state *state)
{
   struct iphdr *iph;
   char *hook;
   char *protocol;

   switch (state->hook){
     case NF_INET_LOCAL_IN:      hook = "LOCAL_IN";      break;
     case NF_INET_LOCAL_OUT:     hook = "LOCAL_OUT";     break;
     case NF_INET_PRE_ROUTING:   hook = "PRE_ROUTING";   break;
     case NF_INET_POST_ROUTING:  hook = "POST_ROUTING";  break;
     case NF_INET_FORWARD:       hook = "FORWARD";       break;
     default:                    hook = "IMPOSSIBLE";    break;
   }
```

```
int registerFilter(void) {
   printk(KERN_INFO "Registering filters.\n");
   hook1.hook = printInfo;
   hook1.hooknum = NF_INET_LOCAL_OUT;
   hook1.pf = PF_INET;
   hook1.priority = NF_IP_PRI_FIRST;
   nf_register_net_hook(&init_net, &hook1);

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

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

   hook4.hook = printInfo;
   hook4.hooknum = NF_INET_FORWARD;
   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);
```

```
void removeFilter(void) {
   printk(KERN_INFO "The filters are being removed.\n");
   nf_unregister_net_hook(&init_net, &hook1);
   nf_unregister_net_hook(&init_net, &hook2);
   nf_unregister_net_hook(&init_net, &hook3);
   nf_unregister_net_hook(&init_net, &hook4);
   nf_unregister_net_hook(&init_net, &hook5);
}
```

编译并载入内核：

```
[07/26/21]seed@VM:~/firewal2$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/firewal2 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/firewal2/seedFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/seed/firewal2/seedFilter.mod.o
  LD [M]  /home/seed/firewal2/seedFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[07/26/21]seed@VM:~/firewal2$ sudo insmod seedFilter.ko
```

在 10.9.0.1 上 ping 10.9.0.5

输入 dmesg 查看出现 LOCAL_IN, LOCAL_OUT, PRE_ROUTING, POST_ROUTING

```
[ 1597.097318] *** LOCAL_IN
[ 1597.097320]     127.0.0.53  --> 127.0.0.1 (UDP)
[ 1598.112066] *** LOCAL_OUT
[ 1598.112068]     10.9.0.1  --> 224.0.0.251 (UDP)
[ 1598.112075] *** POST_ROUTING
[ 1598.112076]     10.9.0.1  --> 224.0.0.251 (UDP)
[ 1598.112080] *** PRE_ROUTING
[ 1598.112080]     10.9.0.1  --> 224.0.0.251 (UDP)
[ 1598.112082] *** LOCAL_IN
[ 1598.112083]     10.9.0.1  --> 224.0.0.251 (UDP)
```

① NF_INET_PRE_ROUTING: 除了混杂模式，所有数据包都将经过这个钩子点。它上面注册的 hook 函数在路由判决之前被调用。
②NF_INET_LOCAL_IN:数据包要进行路由判决，以决定需要被转发还是发往本机。前一种情况下，数据包将前往转发路径;而后一种情况下， 数据包将通过这个钩子点，之后被发送到网络协议栈，并最终被主机接收。
③NFINET FORWARD：需要被转发的数据包会到达这个函数
④NF_INET_LOCAL_OUT：本机产生的数据包将会第一个到达此 hook
⑤NF_INET_POST_ROUTING:需要被转发或者由本机产生的数据包都会经过这个钩子点，经处理后传输到网络时

3.
修改 seedFilter.c 的代码如下：

```
static struct nf_hook_ops hook1, hook2;

unsigned int blockTELNET(void *priv, struct sk_buff *skb,
                         const struct nf_hook_state *state)
{
   struct iphdr *iph;
   struct tcphdr *tcph;

   iph=ip_hdr(skb);
   tcph=tcp_hdr(skb);

   if(iph->protocol==IPPROTO_TCP&&tcph->dest==htons(23))
   {
        return NF_DROP;
   }
   else
   {
        return NF_ACCEPT;
   }
}

unsigned int blockICMP(void *priv, struct sk_buff *skb,
                        const struct nf_hook_state *state)
{
   struct iphdr *iph;
   iph=ip_hdr(skb);

   if(iph->protocol==IPPROTO_ICMP)
   {
        return NF_DROP;
   }
   else
   {
        return NF_ACCEPT;
   }
}
```

```c
int registerFilter(void) {
   printk(KERN_INFO "Registering filters.\n");

   hook1.hook = blockTELNET;
   hook1.hooknum = NF_INET_LOCAL_IN;
   hook1.pf = PF_INET;
   hook1.priority = NF_IP_PRI_FIRST;
   nf_register_net_hook(&init_net, &hook1);

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


   return 0;
}

void removeFilter(void) {
   printk(KERN_INFO "The filters are being removed.\n");
   nf_unregister_net_hook(&init_net, &hook1);
   nf_unregister_net_hook(&init_net, &hook2);
}
```

编译载入内核

```
[07/26/21]seed@VM:~/firewal2$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/firewal2 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/firewal2/seedFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/seed/firewal2/seedFilter.mod.o
  LD [M]  /home/seed/firewal2/seedFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[07/26/21]seed@VM:~/firewal2$ sudo insmod seedFilter.ko
```

进入 10.9.0.5，发现 ping 和 telnet 都失败

```
[07/26/21]seed@VM:~/.../Labsetup$ dockps
1e910868c21f  host1-192.168.60.5
528f49331bbc  host3-192.168.60.7
ab72ec7de863  seed-router
a0350857c671  hostA-10.9.0.5
14645b44af46  host2-192.168.60.6
[07/26/21]seed@VM:~/.../Labsetup$ docksh a0
root@a0350857c671:/# 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 ---
8 packets transmitted, 0 received, 100% packet loss, time 7176ms

root@a0350857c671:/# telnet 10.9.0.1
Trying 10.9.0.1...
^C
root@a0350857c671:/# telnet 10.9.0.1
Trying 10.9.0.1...

^C
root@a0350857c671:/#
```

Task 2: Experimenting with Stateless Firewall Rules

Task 2.A: Protecting the Router

在 router 上输入以下命令：

```
root@f1daa2da4050:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
root@f1daa2da4050:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@f1daa2da4050:/# iptables -P OUTPUT DROP
root@f1daa2da4050:/# iptables -P INPUT DROP
```

输入 iptables -L 查看结果：

```
root@f1daa2da4050:/# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             anywhere             icmp echo-reply
root@f1daa2da4050:/#
```

-A 指把规则加到 chain 上

-p icmp --icmp-type echo-request 指该规则只用于 icmp 响应报文

-p icmp --icmp-type echo-reply 指该规则只用于 icmp 请求报文

-j ACCEPT 指接受满足此规则的包

在 router 上 Ping10.9.0.5ping 不通

```
root@f1daa2da4050:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 10.9.0.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3076ms
```

在 10.9.0.5 上 ping router 可以通

```
[07/29/21]seed@VM:~/.../Labsetup$ dockps
a0c9490ba9a2  hostA-10.9.0.5
599136a7c198  host1-192.168.60.5
dc9e84c9bdf0  host3-192.168.60.7
f1daa2da4050  seed-router
5d12a3f6547e  host2-192.168.60.6
[07/29/21]seed@VM:~/.../Labsetup$ docksh a0
root@a0c9490ba9a2:/# 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=0.123 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.088 ms
^C
--- 10.9.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1013ms
rtt min/avg/max/mdev = 0.088/0.105/0.123/0.017 ms
```

复原 filter 表的状态：

```
root@f1daa2da4050:/# iptables -F
root@f1daa2da4050:/# iptables -P OUTPUT ACCEPT
root@f1daa2da4050:/# iptables -P INPUT ACCEPT
root@f1daa2da4050:/#
```

Task 2.B: Protecting the Internal Network

在 router 上输入 ip addr 查看端口号：

```
root@f1daa2da4050:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
13: eth0@if14: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.11/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
17: eth1@if18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:a8:3c:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.60.11/24 brd 192.168.60.255 scope global eth1
       valid_lft forever preferred_lft forever
```

可见内网 192.168.60.0/24 端口为 eth1，10.9.0.0/24 方向端口为 eth0

在 router 上输入以下命令：

```
root@f1daa2da4050:/# iptables -A FORWARD -p icmp --icmp-type echo-request -i eth0 -j DROP
```

不允许转发从 eth0 进入的 ICMP 请求报文，满足外部无法 ping 通内部

```
root@f1daa2da4050:/# iptables -A FORWARD -p icmp --icmp-type echo-reply -i eth0 -j ACCEPT
```

允许转发 eth0 进入的 ICMP 响应报文，满足外部无法 ping 通内部

```
root@f1daa2da4050:/# iptables -A FORWARD -p icmp --icmp-type echo-request -i eth1 -j ACCEPT
```

允许转发从 eth1 进入的 ICMP 请求报文，满足外部无法 ping 通内部

```
root@f1daa2da4050:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

允许接受 ICMP 请求报文，满足外部 ping 通路由器

```
root@f1daa2da4050:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

允许发出 ICMP 回应报文，满足外部 ping 通路由器

```
root@f1daa2da4050:/# iptables -P OUTPUT DROP
root@f1daa2da4050:/# iptables -P INPUT DROP
root@f1daa2da4050:/# iptables -P FORWARD DROP
```

其他默认丢弃

输入 iptables -L 查看规则设置结果：

```
root@f1daa2da4050:/# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request

Chain FORWARD (policy DROP)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere             icmp echo-request
ACCEPT     icmp --  anywhere             anywhere             icmp echo-reply
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             anywhere             icmp echo-reply
root@f1daa2da4050:/#
```

ping 测试结果如下：

①外网 ping 不通内网

```
root@a0c9490ba9a2:/# 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 ---
12 packets transmitted, 0 received, 100% packet loss, time 11266ms
```

②外网 ping 的通路由器

```
root@a0c9490ba9a2:/# 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=0.089 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.151 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.247 ms
^C
--- 10.9.0.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.089/0.162/0.247/0.065 ms
root@a0c9490ba9a2:/#
```

③内网 ping 的通外网

```
[07/29/21]seed@VM:~/.../Labsetup$ dockps
a0c9490ba9a2  hostA-10.9.0.5
599136a7c198  host1-192.168.60.5
dc9e84c9bdf0  host3-192.168.60.7
f1daa2da4050  seed-router
5d12a3f6547e  host2-192.168.60.6
[07/29/21]seed@VM:~/.../Labsetup$ docksh 59
root@599136a7c198:/# 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=0.162 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.194 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.197 ms
^C
--- 10.9.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.162/0.184/0.197/0.015 ms
root@599136a7c198:/#
```

④其他流量被阻断

```
root@599136a7c198:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@599136a7c198:/#
```

Task 2.C: Protecting Internal Servers

在 router 上输入以下命令：

```
root@6f3eed90b715:/# iptables -A FORWARD -i eth0 -p tcp -d 192.168.60.5 --dport 23 -j ACCEPT
```

允许转发 interface 为 eth0 一侧的主机的目的端口为 23、目的地址为 192.168.60.5 的 tcp 报文，满足外部主机只能 telnet 登录 192.168.60.5

```
root@6f3eed90b715:/# iptables -A FORWARD -i eth1 -p tcp -s 192.168.60.5 -j ACCEPT
```

允许转发 interface 为 eth1 一侧的 IP 地址为 192.168.60.5 的主机的 tcp 报文，满足外部主机只能 telnet 登录 192.168.60.5

```
root@6f3eed90b715:/# iptables -P FORWARD DROP
```

默认设为丢弃

输入 iptables -L 查看：

```
root@6f3eed90b715:/# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             host1-192.168.60.5.net-192.168.60.0  tcp dpt:telnet
ACCEPT     tcp  --  host1-192.168.60.5.net-192.168.60.0  anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@6f3eed90b715:/#
```

内部主机可以访问内部服务器：

```
28757594b4f5  host1-192.168.60.5
dbfe8cbec053  hostA-10.9.0.5
[07/31/21]seed@VM:~/.../Labsetup$ docksh 28
root@28757594b4f5:/# telnet 192.168.60.6
Trying 192.168.60.6...
Connected to 192.168.60.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
6920abd75aa6 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

内部主机不能访问外部服务器：

```
seed@6920abd75aa6:~$ telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
```

外部只能访问到 192.168.60.5

```
root@dbfe8cbec053:/# telnet 192.168.60.6
Trying 192.168.60.6...
telnet: Unable to connect to remote host: Connection timed out
```

```
root@dbfe8cbec053:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
28757594b4f5 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

Task 3: Connection Tracking and Stateful Firewall

Task 3.A: Experiment with the Connection Tracking

在 router 上输入以下命令后，在 10.9.0.5 上 ping 192.168.10.5，再在 router 上输入 conntrack -L 追踪：

```
root@6f3eed90b715:/# iptables -F
root@6f3eed90b715:/# iptables -P OUTPUT ACCEPT
root@6f3eed90b715:/# iptables -P INPUT ACCEPT
root@6f3eed90b715:/# iptables -P FORWARD ACCEPT
root@6f3eed90b715:/# conntrack -L
icmp     1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=44 src=192.168.60.5 dst=10.9.0.5 type=0
code=0 id=44 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@6f3eed90b715:/#
```

```
root@dbfe8cbec053:/# 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=0.123 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.231 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.201 ms
```

在 192.168.60.5 上输入 nc -lu 9090，在 10.9.0.5 上输入 nc -u 192.168.60.5 9090，在 10.9.0.5 上输入 hello：

```
28757594b4f5  host1-192.168.60.5
dbfe8cbec053  hostA-10.9.0.5
[07/31/21]seed@VM:~/.../Labsetup$ docksh 28
root@28757594b4f5:/# nc -lu 9090
hello
```

```
root@dbfe8cbec053:/# nc -u 192.168.60.5 9090
hello
```

在 router 上追踪：

```
root@6f3eed90b715:/# conntrack -L
udp      17 21 src=10.9.0.5 dst=192.168.60.5 sport=56628 dport=9090 [UNREPLIED] src=192.168.60.5 dst=
10.9.0.5 sport=9090 dport=56628 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@6f3eed90b715:/#
```

先在 192.168.60.5 上输入 hello2，在 10.9.0.11 输入 conntrack -L，追踪状态：

```
root@28757594b4f5:/# nc -lu 9090
hello
hello2
```

```
root@dbfe8cbec053:/# nc -u 192.168.60.5 9090
hello
hello2
```

```
root@6f3eed90b715:/# conntrack -L
udp      17 24 src=192.168.60.5 dst=10.9.0.5 sport=9090 dport=56628 [UNREPLIED]
 src=10.9.0.5 dst=192.168.60.5 sport=56628 dport=9090 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@6f3eed90b715:/#
```

在 192.168.60.5 上输入 nc -l 9090，在 10.9.0.5 上输入 nc 192.168.60.5 9090，先在 192.168.60.5 上输入 hello3，在 router 上追踪状态：

```
root@28757594b4f5:/# nc -l 9090
hello3
```

```
^C
root@dbfe8cbec053:/# nc 192.168.60.5 9090
hello3
```

```
root@6f3eed90b715:/# conntrack -L
tcp      6 431989 ESTABLISHED src=10.9.0.5 dst=192.168.60.5 sport=40978 dport=9
090 src=192.168.60.5 dst=10.9.0.5 sport=9090 dport=40978 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@6f3eed90b715:/#
```

Task 3.B: Setting Up a Stateful Firewall

在 router 中输入以下命令：

```
root@6f3eed90b715:/# iptables -A FORWARD -i eth0 -p tcp -d 192.168.60.5 --dport 23 -m connt
rack --ctstate NEW,ESTABLISHED -j ACCEPT
root@6f3eed90b715:/# iptables -A FORWARD -i eth1 -p tcp -s 192.168.60.5 -m conntrack --ctst
ate ESTABLISHED,RELATED -j ACCEPT
root@6f3eed90b715:/# iptables -A FORWARD -i eth1 -p tcp -d 10.9.0.5 --dport 23 -m conntrack
 --ctstate ESTABLISHED,RELATED -j ACCEPT
root@6f3eed90b715:/# iptables -A FORWARD -i eth0 -p tcp -s 10.9.0.5 -m conntrack --ctstate
ESTABLISHED,RELATED -j ACCEPT
root@6f3eed90b715:/# iptables -P FORWARD DROP
```

输入 iptables -L 查看：

```
root@6f3eed90b715:/# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             host1-192.168.60.5.net-192.1
68.60.0  tcp dpt:telnet ctstate NEW,ESTABLISHED
ACCEPT     tcp  --  host1-192.168.60.5.net-192.168.60.0  anywhere
         ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             hostA-10.9.0.5.net-10.9.0.0
 tcp dpt:telnet ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  hostA-10.9.0.5.net-10.9.0.0  anywhere
 ctstate RELATED,ESTABLISHED

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@6f3eed90b715:/#
```

内部主机可以 telnet 访问外部服务器：

```
root@6920abd75aa6:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
506cbd21cb8a login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

外部主机只能 telnet 访问 192.168.60.5：

```
root@dbfe8cbec053:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
28757594b4f5 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

Task 4: Limiting Network Traffic

在 router 上输入命令 iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -j ACCEPT；iptables -A FORWARD -s 10.9.0.5 -j DROP，在 10.9.0.5 ping 192.168.60.5，发现

前 4 个包的回应都非常正常，但是从第 5 个包开始，每 10 秒才能收到一个正常的回应

```
root@dbfe8cbec053:/# 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=0.078 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.078 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.059 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.060 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.059 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.095 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.059 ms
^C
--- 192.168.60.5 ping statistics ---
20 packets transmitted, 8 received, 60% packet loss, time 19455ms
rtt min/avg/max/mdev = 0.059/0.068/0.095/0.012 ms
```

Task 5: Load Balancing

① round-robin

在 router 上输入以下命令：

```
root@6f3eed90b715:/# 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@6f3eed90b715:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic
--mode nth --every 2 --packet 0 -j DNAT --to-destination 192.168.60.6:8080
root@6f3eed90b715:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -j DNAT --to-
destination 192.168.60.7:8080
root@6f3eed90b715:/#
```

在 10.9.0.5 上输入命令 echo hello | nc -u 10.9.0.11 8080，可以观察到 192.168.60.5、192.168.60.6、192.168.60.7 依次按序收到“hello”

```
root@28757594b4f5:/# nc -luk 8080
hello
hello
```

```
root@6920abd75aa6:/# nc -luk 8080
hello
hello
```

```
root@12684e23bf51:/# nc -luk 8080
hello
hello
```

② random-mode

在 router 上输入命令

```
root@6f3eed90b715:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statisti
c --mode random --probability 0.33 -j DNAT --to-destination 192.168.60.5:8080
root@6f3eed90b715:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statisti
c --mode random --probability 0.5 -j DNAT --to-destination 192.168.60.6:8080
root@6f3eed90b715:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -j DNAT --t
o-destination 192.168.60.7:8080
root@6f3eed90b715:/#
```

重复操作发现收到的 hello 不完全相同：

```
root@28757594b4f5:/# nc -luk 8080
hello
hello
```

```
root@6920abd75aa6:/# nc -luk 8080
hello
hello
hello
hello
hello
```

```
root@12684e23bf51:/# nc -luk 8080
hello
hello
hello
```