Firewall Exploration Lab

57118229 袁超然

3 Task 1: Implementing a Simple Firewall

3.1 Task 1.A: Implement a Simple Kernel Module

将 kernel_module 这个文件夹移到一个没有空格的目录下，然后直接 make 编译，可以看到编译成功，然后下一步通过 insmod hello.ko 将其插入，可以这个模块已经成功进入了内核：

```
[07/26/21]seed@VM:~/kernel_module$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/kernel_modu
le 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_modu
le/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'
[07/26/21]seed@VM:~/kernel_module$ sudo insmod hello.ko
[07/26/21]seed@VM:~/kernel_module$ lsmod
Module                  Size  Used by
hello                  16384  0
xt_nat                 16384  2
```

再将其移除：

```
[07/26/21]seed@VM:~/kernel_module$ sudo rmmod hello
```

通过 dmesg 可查看日志看见其输出。

```
[ 1324.106882] Hello World!  [ 1687.774740] Bye-bye World!.
```

3.2 Task 1.B: Implement a Simple Firewall Using Netfilter

在没有使用 firewall 之前，使用 dig 命令，发现成功：

```
[07/26/21]seed@VM: $ 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: 63444
;; 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.        20683   IN      A       93.184.216.34

;; Query time: 344 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Mon Jul 26 18:51:54 EDT 2021
;; MSG SIZE  rcvd: 60
```

将示例代码运行编译进内核：

```
[07/26/21]seed@VM:~/packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/packet_fil
ter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generi
c'
  CC [M]  /home/seed/packet_filter/seedFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/seed/packet_filter/seedFilter.mod.o
  LD [M]  /home/seed/packet_filter/seedFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic
'
[07/26/21]seed@VM:~/packet_filter$ sudo insmod seedFilter.ko
[07/26/21]seed@VM:~/packet_filter$ lsmod | grep seedFilter
seedFilter             16384  0
[07/26/21]seed@VM:~/packet_filter$
```

再次进行 dig，发现连接失败，说明有效：

```
[07/26/21]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
```

五种钩子的位置与功能：

①LOCAL_OUT：本机产生的数据包到达的第一个钩子，此处可进行内置的源地址转换；

②POST_ROUTING：需要被转发或者由本机产生的数据包都会经过的一个钩子；

```
[ 3522.599580] *** LOCAL_OUT
[ 3522.599580]     192.168.88.130  --> 8.8.8.8 (UDP)
[ 3522.599583] *** Dropping 8.8.8.8 (UDP), port 53
[ 3527.597037] *** LOCAL_OUT
[ 3527.597040]     192.168.88.130  --> 8.8.8.8 (UDP)
[ 3527.597050] *** Dropping 8.8.8.8 (UDP), port 53
[ 3532.598357] *** LOCAL_OUT
[ 3532.598358]     192.168.88.130  --> 8.8.8.8 (UDP)
[ 3532.598367] *** Dropping 8.8.8.8 (UDP), port 53
```

③PRE_ROUTING：除了混杂模式，所有数据包都将经过这个钩子点。它上面注册的钩子函数在路由判决之前被调用，可进行目的地址转换；

```
[ 8345.421980]     112.80.248.75  --> 10.0.2.15 (ICMP)
[ 8345.422332] *** PRE_ROUTING
[ 8345.422335]     127.0.0.1  --> 127.0.0.53 (UDP)
[ 8345.425224] *** PRE_ROUTING
[ 8345.425228]     10.80.128.28  --> 10.0.2.15 (UDP)
[ 8345.427106] *** PRE_ROUTING
[ 8345.427109]     10.80.128.28  --> 10.0.2.15 (UDP)
[ 8345.427488] *** PRE_ROUTING
```

④LOCAL_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。

```
[ 8751.385140] *** LOCAL_IN
[ 8751.385145]     35.232.111.17  --> 10.0.2.15 (TCP)
[ 8751.385696] *** LOCAL_IN
[ 8751.385698]     35.232.111.17  --> 10.0.2.15 (TCP)
[ 8751.649998] *** LOCAL_IN
[ 8751.650047]     35.232.111.17  --> 10.0.2.15 (TCP)
[ 8751.650137] *** LOCAL_IN
[ 8751.650152]     35.232.111.17  --> 10.0.2.15 (TCP)
[ 8751.651494] *** LOCAL_IN
[ 8751.651499]     35.232.111.17  --> 10.0.2.15 (TCP)
```

⑤*FORWARD*：需要被转发的数据包会到达这个钩子，此处可进行 FORWARD 过滤。

```
[ 9208.982523] *** FORWARD
[ 9208.982526]     192.168.60.5  --> 10.9.0.6 (ICMP)
[ 9211.043983] *** FORWARD
[ 9211.043987]     192.168.60.11  --> 192.168.60.5 (ICMP)
[ 9211.043998] *** FORWARD
[ 9211.044000]     192.168.60.11  --> 192.168.60.5 (ICMP)
[ 9211.116597] *** FORWARD
[ 9211.116600]     192.168.60.5  --> 10.9.0.1 (ICMP)
```

接下来，针对第三小题，根据题目要求，分别设计两个钩子函数，一个为 blockTELNET，判断是否为 TCP 协议，并且目的端口为 23，另一个为 blockPING，判断是否为 ICMP 协议，两个函数都挂在 NF_INET_LOCAL_IN 这个钩子下。代码如下：

```c
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/udp.h>
#include <linux/if_ether.h>
#include <linux/inet.h>
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;
  u16 port = 23;
  char ip[16] = "10.9.0.1";
  u32 ip_addr;
  if (!skb) return NF_ACCEPT;
  iph = ip_hdr(skb);
  // Convert the IPv4 address from dotted decimal to 32-bit binary
  in4_pton(ip, -1, (u8 *)&ip_addr, '\0', NULL);

  if (iph->protocol == IPPROTO_TCP) {
    tcph = tcp_hdr(skb);
    if (iph->daddr == ip_addr && ntohs(tcph->dest) == port){
      printk(KERN_WARNING "*** Dropping %pI4 (TELNET)\n", &(iph->saddr));
      return NF_DROP;
    }
  }
return NF_ACCEPT;
}
```

```c
unsigned int blockPING(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
  struct iphdr *iph;
  char ip[16] = "10.9.0.1";
  u32 ip_addr;
  if (!skb) return NF_ACCEPT;
  iph = ip_hdr(skb);
  // Convert the IPv4 address from dotted decimal to 32-bit binary
  in4_pton(ip, -1, (u8 *)&ip_addr, '\0', NULL);
  if (iph->protocol == IPPROTO_ICMP) {
    if (iph->daddr == ip_addr){
      printk(KERN_WARNING "*** Dropping %pI4 (PING)\n", &(iph->saddr));
      return NF_DROP;
    }
  }
return NF_ACCEPT;
}
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 = blockPING;
  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);
}
module_init(registerFilter);
module_exit(removeFilter);
MODULE_LICENSE("GPL");
```

重新编译，然后下一步通过 insmod 命令将其插入：

```
[07/26/21]seed@VM:~/packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/packet_f
ilter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-gene
ric'
  CC [M]  /home/seed/packet_filter/seedFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/seed/packet_filter/seedFilter.mod.o
  LD [M]  /home/seed/packet_filter/seedFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-gener
ic'
[07/26/21]seed@VM:~/packet_filter$ sudo rmmod seedFilter
rmmod: ERROR: Module seedFilter is not currently loaded
[07/26/21]seed@VM:~/packet_filter$ sudo insmod seedFilter.ko
[07/26/21]seed@VM:~/packet_filter$ lsmod | grep seedFilter
seedFilter             16384  0
```

接下来，登录客户机 10.9.0.5，尝试 ping 10.9.0.1，发现失败：

```
root@2b46991133ca:/# ping 10.9.0.1
PING 10.9.0.1 (10.9.0.1) 56(84) bytes of data.
64 bytes from 10.9.0.1: icmp_seq=1 ttl=64 time=0.093 ms
64 bytes from 10.9.0.1: icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from 10.9.0.1: icmp_seq=3 ttl=64 time=0.063 ms
64 bytes from 10.9.0.1: icmp_seq=4 ttl=64 time=0.047 ms
64 bytes from 10.9.0.1: icmp_seq=5 ttl=64 time=0.200 ms
64 bytes from 10.9.0.1: icmp_seq=6 ttl=64 time=0.067 ms
64 bytes from 10.9.0.1: icmp_seq=7 ttl=64 time=0.042 ms
^Z
[1]+  Stopped                 ping 10.9.0.1
root@2b46991133ca:/# 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
```

查看日志文件，发现 ping 包被 drop：

```
[ 6719.377486] *** Dropping 10.9.0.5 (PING)
[ 6720.385750] *** Dropping 10.9.0.5 (PING)
[ 6721.409047] *** Dropping 10.9.0.5 (PING)
[ 6722.433644] *** Dropping 10.9.0.5 (PING)
```

再次尝试 telnet 连接，也失败：

```
root@2b46991133ca:/# telnet 10.9.0.1
Trying 10.9.0.1...
^C
```

说明防火墙成功工作：

```
[ 6811.868791] *** Dropping 10.9.0.5 (TELNET)
[ 6812.894130] *** Dropping 10.9.0.5 (TELNET)
[ 6814.912134] *** Dropping 10.9.0.5 (TELNET)
[ 6819.103926] *** Dropping 10.9.0.5 (TELNET)
[ 6827.297803] *** Dropping 10.9.0.5 (TELNET)
```

Task 2: Experimenting with Stateless Firewall Rules

Task 2.A: Protecting the Router

将前两条命令所在的 Chain 进行一个对换，在 10.9.0.11 端输入：

```
root@03d8d2073b04:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@03d8d2073b04:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
root@03d8d2073b04:/# iptables -P OUTPUT DROP
root@03d8d2073b04:/# iptables -P INPUT DROP
root@03d8d2073b04:/#
```

发现可以 ping 成果，但 telnet 连接失败：

```
root@bea8d3f9d13f:/# 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=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.062 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.051 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.048 ms
^C
--- 10.9.0.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4097ms
rtt min/avg/max/mdev = 0.048/0.059/0.089/0.015 ms
root@bea8d3f9d13f:/# telnet 10.9.0.11
Trying 10.9.0.11...
^C
root@bea8d3f9d13f:/#
```

Task 2.B: Protecting the Internal Network

根据要求，在 router 上执行如下 3 条命令：

```
root@03d8d2073b04:/# iptables -A FORWARD -i eth1 -p icmp --icmp-type echo-request -j ACCEPT
root@03d8d2073b04:/# iptables -A FORWARD -o eth1 -p icmp --icmp-type echo-reply -j ACCEPT
root@03d8d2073b04:/# iptables -P FORWARD DROP
```

接下来进行测试验证：

首先，尝试 outside host ping inside host，发现失败：

```
root@bea8d3f9d13f:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^Z
[2]+  Stopped                 ping 192.168.60.5
```

尝试 outside host ping route，发现可以 ping 通：

```
root@bea8d3f9d13f:/# 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=0.213 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.185 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.419 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.083 ms
^Z
[3]+  Stopped                 ping 10.9.0.11
root@bea8d3f9d13f:/# 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=0.071 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.057 ms
64 bytes from 192.168.60.11: icmp_seq=3 ttl=64 time=0.075 ms
^Z
[4]+  Stopped                 ping 192.168.60.11
```

尝试 inside host ping outside host，发现成功：

```
root@aa859c021f9b:/# 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=0.112 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.080 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.058 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.061 ms
^C
--- 10.9.0.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3071ms
rtt min/avg/max/mdev = 0.058/0.077/0.112/0.021 ms
```

All other packets between the internal and external networks should be blocked ：

```
root@aa859c021f9b:/# telnet 10.9.0.5
Trying 10.9.0.5...
^Z^C
```

```
root@bea8d3f9d13f:/# telnet 192.168.60.5
Trying 192.168.60.5...
^Z^C
```

Task 2.C: Protecting Internal Servers

根据要求，在 router 上运行如下命令：

```
root@03d8d2073b04:/# iptables -A FORWARD -i eth0 -d 192.168.60.5 -p tcp --dport 23 -j ACCEPT
root@03d8d2073b04:/# iptables -A FORWARD -o eth0 -s 192.168.60.5 -p tcp --sport 23 -j ACCEPT
root@03d8d2073b04:/# iptables -P FORWARD DROP
root@03d8d2073b04:/#
```

Outside hosts can only access the telnet server 192.168.60.5

```
root@bea8d3f9d13f:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
seed
deesUbuntu 20.04.1 LTS
seed
deesa4daeaff506c login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@a4daeaff506c:~$ exit
```

Outside hosts cannot access other internal hosts.

```
root@bea8d3f9d13f:/# telnet 192.168.60.6
Trying 192.168.60.6...
^Z^C
root@bea8d3f9d13f:/#
```

Internal hosts can access all the internal servers.

```
root@aa859c021f9b:/# telnet 192.168.60.7
Trying 192.168.60.7...
Connected to 192.168.60.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
66f43f5ed039 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

Internal hosts cannot access external servers.

```
root@aa859c021f9b:/# telnet 10.9.0.5
Trying 10.9.0.5...
^C
root@aa859c021f9b:/#
```

Task 3: Connection Tracking and Stateful Firewall

Task 3.A: Experiment with the Connection Tracking

在 10.9.0.5 上来 ping 192.168.60.5，在 router 上进行 conntrack -L，发现第三个字段的值基本以秒为单位递减，递减至 0 时，连接消失，猜测是连接状态的计时器。也就是说 ICMP 连接持续时间为 30s：

```
root@1411aa150635:/# conntrack -L
icmp     1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=29 src=192.168.60.5
 dst=10.9.0.5 type=0 code=0 id=29 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
```

UDP 实验步骤基本同上，UDP 的连接时间随着输入字符串长度变长而变长，最长为 30s：

```
root@1411aa150635:/# conntrack -L
udp      17 21 src=10.9.0.5 dst=192.168.60.5 sport=60359 dport=9090 [UNREPLIED]
src=192.168.60.5 dst=10.9.0.5 sport=9090 dport=60359 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
```

```
root@1411aa150635:/# conntrack -L
udp      17 27 src=10.9.0.5 dst=192.168.60.5 sport=60359 dport=9090 [UNREPLIED]
src=192.168.60.5 dst=10.9.0.5 sport=9090 dport=60359 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
```

TCP 实验：

```
root@1411aa150635:/# conntrack -L
tcp      6 431996 ESTABLISHED src=10.9.0.5 dst=192.168.60.5 sport=57384 dport=90
90 src=192.168.60.5 dst=10.9.0.5 sport=9090 dport=57384 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
```

处于 TCP ESTABLISHED 连接状态时，router 上保持连接的时间为 432000s = 7200min = 120h = 5day。

在一方使用 Ctrl + C 命令中止连接后处于 TIME_WAIT 状态，这时连接时间会从 120s 开始逐渐变短，直至为 0：

```
root@1411aa150635:/# conntrack -L
tcp      6 114 TIME_WAIT src=10.9.0.5 dst=192.168.60.5 sport=57384 dport=9090 sr
c=192.168.60.5 dst=10.9.0.5 sport=9090 dport=57384 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1411aa150635:/# conntrack -L
tcp      6 12 TIME_WAIT src=10.9.0.5 dst=192.168.60.5 sport=57384 dport=9090 src
=192.168.60.5 dst=10.9.0.5 sport=9090 dport=57384 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1411aa150635:/# conntrack -L
tcp      6 3 TIME_WAIT src=10.9.0.5 dst=192.168.60.5 sport=57384 dport=9090 src=
192.168.60.5 dst=10.9.0.5 sport=9090 dport=57384 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1411aa150635:/# conntrack -L
tcp      6 1 TIME_WAIT src=10.9.0.5 dst=192.168.60.5 sport=57384 dport=9090 src=
192.168.60.5 dst=10.9.0.5 sport=9090 dport=57384 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1411aa150635:/# conntrack -L
conntrack v1.4.5 (conntrack-tools): 0 flow entries have been shown.
```

Task 3.B: Setting Up a Stateful Firewall

在 router 上运行以下命令：

```
root@1411aa150635:/# iptables -A FORWARD -p tcp -m conntrack --ctstate ESTABLISH
ED,RELATED -j ACCEPT
root@1411aa150635:/# iptables -A FORWARD -p tcp -d 192.168.60.5 -i eth0 --dport
23 --syn -m conntrack --ctstate NEW -j ACCEPT
root@1411aa150635:/# iptables -A FORWARD -p tcp -i eth1 --dport 23 --syn -m conn
track --ctstate NEW -j ACCEPT
root@1411aa150635:/# iptables -P FORWARD DROP
```

Outside hosts can only access the telnet server 192.168.60.5 ：

```
root@6a797811d48d:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
e683486742cd login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

Outside hosts cannot access other internal hosts.

```
root@6a797811d48d:/# telnet 192.168.60.6
Trying 192.168.60.6...
^C
root@6a797811d48d:/#
```

Internal hosts can access all the internal servers.

```
root@e683486742cd:/# telnet 192.168.60.6
Trying 192.168.60.6...
Connected to 192.168.60.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
ad9cad5618ce login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

Internal hosts can access external servers

```
[07/28/21]seed@VM:~/.../Labsetup$ docksh e6
root@e683486742cd:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
6a797811d48d 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 上运行下面这条命令：

```
root@1411aa150635:/# iptables -A FORWARD -s 10.9.0.5 -m limit --lim
it 10/minute --limit-burst 5 -j ACCEPT
```

然后在 10.9.0.5 上 ping192.168.60.5.此时 ICMP 报文并没有收到限制：

```
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=0.124 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.055 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.071 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.158 ms
^C
--- 192.168.60.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4072ms
rtt min/avg/max/mdev = 0.055/0.097/0.158/0.038 ms
root@6a797811d48d:/#
```

增加下面这条命令，然后再次进行 PING 命令:

```
root@1411aa150635:/# iptables -A FORWARD -s 10.9.0.5 -j DROP
```

可以通过序列号明显的发现，从第 5 个报文后，开始出现丢包的现象，也就是说受到了限制。

```
root@6a797811d48d:/# 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=0.106 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.065 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.093 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.069 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.063 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.058 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.059 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.096 ms
64 bytes from 192.168.60.5: icmp_seq=25 ttl=63 time=0.060 ms
64 bytes from 192.168.60.5: icmp_seq=31 ttl=63 time=0.054 ms
64 bytes from 192.168.60.5: icmp_seq=37 ttl=63 time=0.067 ms
64 bytes from 192.168.60.5: icmp_seq=43 ttl=63 time=0.089 ms
64 bytes from 192.168.60.5: icmp_seq=48 ttl=63 time=0.098 ms
^C
--- 192.168.60.5 ping statistics ---
53 packets transmitted, 13 received, 75.4717% packet loss, time 532
39ms
```

Task 5: Load Balancing

Using the nth mode (round-robin)：

在 router 上运行如下命令：

```
root@1411aa150635:/# iptables -t nat -A PREROUTING -p udp --dport 8
080 -m statistic --mode nth --every 3 --packet 0 -j DNAT --to-desti
nation 192.168.60.5:8080
root@1411aa150635:/# iptables -t nat -A PREROUTING -p udp --dport 8
080 -m statistic --mode nth --every 3 --packet 1 -j DNAT --to-desti
nation 192.168.60.6:8080
root@1411aa150635:/# iptables -t nat -A PREROUTING -p udp --dport 8
080 -m statistic --mode nth --every 3 --packet 2 -j DNAT --to-desti
nation 192.168.60.7:8080
```

可以看到三个 server 上依次收到报文：

```
root@e683486742cd:/# nc -luk 8080
hello
hello
```

```
root@ad9cad5618ce:/# nc -luk 8080
hello
```

```
root@ad19f2e30234:/# nc -luk 8080
hello
```

Using the random mode

在 router 上运行如下命令三个 server 概率均等为 0.33：

```
root@1411aa150635:/# iptables -t nat -A PREROUTING -p udp --dport 8
080 -m statistic --mode random --probability 0.33 -j DNAT --to-dest
ination 192.168.60.5:8080
root@1411aa150635:/# iptables -t nat -A PREROUTING -p udp --dport 8
080 -m statistic --mode random --probability 0.33 -j DNAT --to-dest
ination 192.168.60.6:8080
root@1411aa150635:/# iptables -t nat -A PREROUTING -p udp --dport 8
080 -m statistic --mode random --probability 0.33 -j DNAT --to-dest
ination 192.168.60.7:8080
```

可以看到基本上处于一个负载均衡的状态：

```
root@e683486742cd:/# nc -luk 8080
hello
hello
hello
hello
hello
```

```
root@ad19f2e30234:/# nc -luk 8080
hello
hello
hello
```

```
root@ad9cad5618ce:/# nc -luk 8080
hello
hello
hello
hello
```