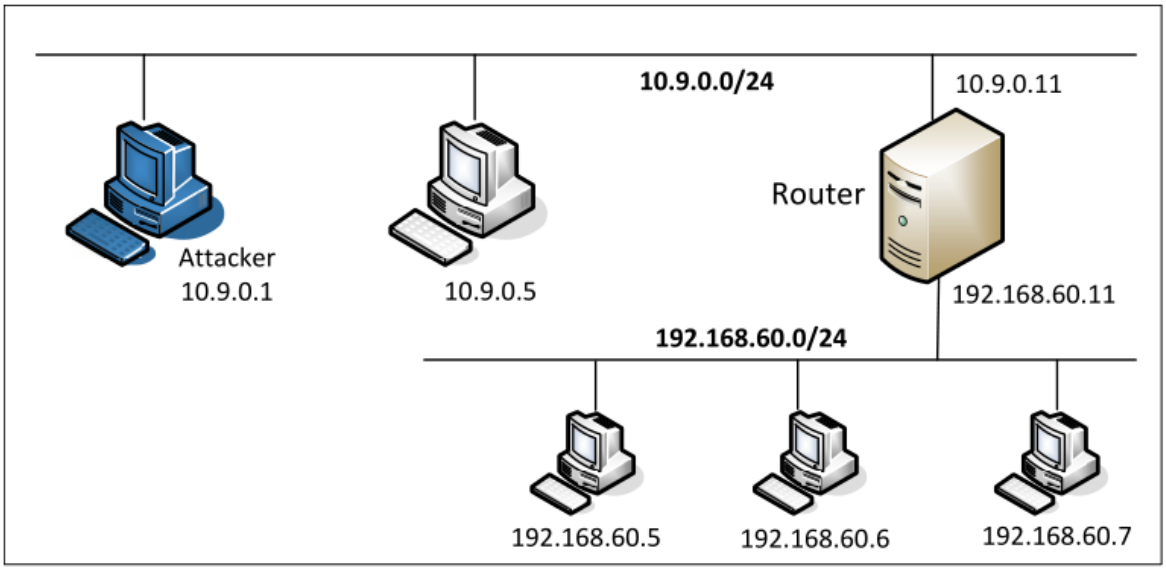
Lab 6：Firewall Exploration Lab

57118103 郭欣然

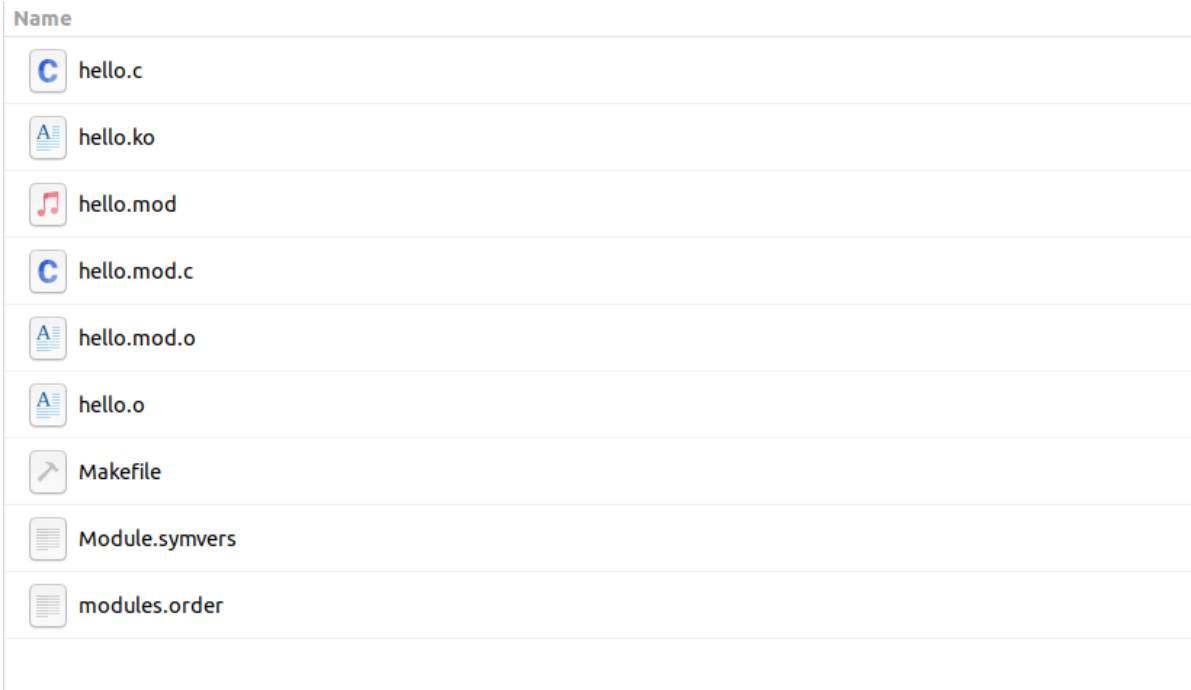
实验环境如下：



Task 1.A: Implementing a Simple Firewall

编译内核模块

```
[08/04/21]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'
[08/04/21]seed@VM:~/kernel_module$
```



模块信息如下

```
[08/04/21]seed@VM:~/kernel_module$ sudo insmod hello.ko
[08/04/21]seed@VM:~/kernel_module$ lsmod | grep hello
hello                  16384  0
[08/04/21]seed@VM:~/kernel_module$ sudo rmmod hello
[08/04/21]seed@VM:~/kernel_module$ dmesg
[    0.000000] Linux version 5.4.0-54-generic (buildd@lcy01-am
#60-Ubuntu SMP Fri Nov 6 10:37:59 UTC 2020 (Ubuntu 5.4.0-54.60
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-54
uiet splash
```

输出hello world和bye bye world

```
[30815.726315] hello: module license 'unspecified' taints kernel.
[30815.726318] Disabling lock debugging due to kernel taint
[30815.807258] hello: module verification failed: signature and/or re
[30816.155733] Hello World!
[30835.168016] Bye-bye World!.
[08/04/21]seed@VM:~/kernel_module$
```

Task 1.B: Implement a Simple Firewall Using Netfilter

1、首先在主机上利用dig查询www.example.com的DNS如下，可知能够获得相关信息。

```
[08/04/21]seed@VM:~/kernel_module$ 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: 29745
;; 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.        3899    IN      A       93.184.216.34

;; Query time: 75 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed Aug 04 22:40:37 EDT 2021
;; MSG SIZE  rcvd: 60

[08/04/21]seed@VM:~/kernel_module$
```

编译加载开启防火墙后，连接失败。

```
[08/04/21]seed@VM:~/packet_filter$ sudo insmod seedFilter.ko
[08/04/21]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

[08/04/21]seed@VM:~/packet_filter$
```

2、

修改代码，增加hook

```
int registerFilter(void) {
printk(KERN_INFO "Registering filters.\n");
// Hook 1
hook1.hook = printInfo;
hook1.hooknum = NF_INET_LOCAL_IN;
hook1.pf = PF_INET;
hook1.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook1);
// Hook 2
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);
// Hook 3
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);
// Hook 4
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);
// Hook 5
hook5.hook = printInfo;
hook5.hooknum = NF_INET_POST_ROUTING;
```

利用make命令编译可装载内核模块，并且利用insmod命令插入内核模块

```
[08/04/21]seed@VM:~/packet_filter$ sudo rmmod seedFilter
[08/04/21]seed@VM:~/packet_filter$ sudo insmod seedFilter.ko
[08/04/21]seed@VM:~/packet_filter$ lsmod | grep seedFilter
seedFilter             16384  0
[08/04/21]seed@VM:~/packet_filter$
```

ping 10.9.0.1，可以ping通

```
[08/04/21]seed@VM:~/packet_filter$ 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=13.8 ms
64 bytes from 10.9.0.1: icmp_seq=2 ttl=64 time=0.067 ms
64 bytes from 10.9.0.1: icmp_seq=3 ttl=64 time=0.062 ms
64 bytes from 10.9.0.1: icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from 10.9.0.1: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 10.9.0.1: icmp_seq=6 ttl=64 time=0.065 ms
64 bytes from 10.9.0.1: icmp_seq=7 ttl=64 time=0.049 ms
64 bytes from 10.9.0.1: icmp_seq=8 ttl=64 time=0.043 ms
64 bytes from 10.9.0.1: icmp_seq=9 ttl=64 time=0.058 ms
```

使用dmesg查看信息

```
[33146.074028]     10.9.0.1  --> 10.9.0.1 (ICMP)
[33146.074040] *** PRE_ROUTING
[33146.074041]     192.168.220.132  --> 10.9.0.1 (ICMP)
[33146.074043] *** LOCAL_IN
[33146.074043]     192.168.220.132  --> 10.9.0.1 (ICMP)
[33146.074053] *** LOCAL_OUT
[33146.074054]     10.9.0.1  --> 192.168.220.132 (ICMP)
[33146.074057] *** POST_ROUTING
[33146.074057]     10.9.0.1  --> 10.9.0.1 (ICMP)
[33146.074060] *** PRE_ROUTING
[33146.074060]     10.9.0.1  --> 10.9.0.1 (ICMP)
[33146.074061] *** LOCAL_IN
[33146.074061]     10.9.0.1  --> 10.9.0.1 (ICMP)
[33147.094752] *** LOCAL_OUT
[33147.094755]     10.9.0.1  --> 10.9.0.1 (ICMP)
[33147.094761] *** POST_ROUTING
[33147.094762]     10.9.0.1  --> 10.9.0.1 (ICMP)
[33147.094774] *** PRE_ROUTING
[33147.094775]     192.168.220.132  --> 10.9.0.1 (ICMP)
[33147.094777] *** LOCAL_IN
[33147.094778]     192.168.220.132  --> 10.9.0.1 (ICMP)
[33147.094789] *** LOCAL_OUT
[33147.094790]     10.9.0.1  --> 192.168.220.132 (ICMP)
[33147.094793] *** POST_ROUTING
```

据报从进入系统，进行IP校验以后，首先经过第一个HOOK函数NF_IP_PRE_ROUTING进行处理；
然后就进入路由代码，其决定该数据报是需要转发还是发给本机的；
若该数据报是发被本机的，则该数据经过HOOK函数NF_IP_LOCAL_IN处理以后然后传递给上层协议；
若该数据报应该被转发则它被NF_IP_FORWARD处理；

经过转发的数据报经过最后一个HOOK函数NF_IP_POST_ROUTING处理以后，再传输到网络上。

本地产生的数据经过HOOK函数NF_IP_LOCAL_OUT 处理后，进行路由选择处理，然后经过NF_IP_POST_ROUTING处理后发送出去。

3、

修改代码如下：

```
#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>

static struct nf_hook_ops telnetFilterHook;

unsigned int telnetFilter(void *priv, struct sk_buff * skb, const struct
nf_hook_state *state){

            struct iphdr *iph;
            struct tcphdr *tcph;
            iph = ip_hdr(skb);
            tcph = (void *)iph+iph->ihl*4;

            if((iph->protocol == IPPROTO_TCP && (tcph->dest == htons(23)
```

```
                || tcph->dest== htons(22)
                || tcph->dest== htons(21)))
                || (iph->protocol == IPPROTO_ICMP &&((((unsigned char *)&iph-
>daddr)[0]==10 &&
                  ((unsigned char *)&iph->daddr)[1]==9
                  && ((unsigned char *)&iph->daddr)[2]==0 && ((unsigned char
*)&iph->daddr)[3]==1)
                || (((unsigned char *)&iph->daddr)[0]==10 &&  ((unsigned char
*)&iph->daddr)[1]==9
                  && ((unsigned char *)&iph->daddr)[2]==0 && ((unsigned char
*)&iph->daddr)[3]==1))))){
                 printk(KERN_INFO "Dropping telent packdt to %d.%d.%d.%d\n",
                 ((unsigned char *)&iph->daddr)[0],
                 ((unsigned char *)&iph->daddr)[1],
                 ((unsigned char *)&iph->daddr)[2],
                 ((unsigned char *)&iph->daddr)[3]);
                 return NF_DROP;
             }else{
                 return NF_ACCEPT;
             }
         }
void removeFilter(void){
    printk(KERN_INFO "Telnet filter has been removed.\n");
    nf_unregister_net_hook(&init_net,&telnetFilterHook);
}

int setUpFilter(void){

    telnetFilterHook.hook = telnetFilter;
    telnetFilterHook.hooknum = NF_INET_PRE_ROUTING;
    telnetFilterHook.pf = PF_INET;
    telnetFilterHook.priority = NF_IP_PRI_FILTER;

    if(nf_register_net_hook(&init_net,&telnetFilterHook)!=0){
        printk(KERN_WARNING "register Telnet filter hook error!\n");
        goto err;
    }
    printk(KERN_INFO "Registering a Telnet filter");
    return 0;

err:
    removeFilter();
    return -1;
}

module_init(setUpFilter);
module_exit(removeFilter);

MODULE_LICENSE("GPL");
```

加载内核

```
[08/04/21]seed@VM:~/packet_filter$ sudo rmmod seedFilter
[08/04/21]seed@VM:~/packet_filter$ sudo insmod seedFilter.ko
[08/04/21]seed@VM:~/packet_filter$ lsmod | grep hello seedFilter
grep: seedFilter: No such file or directory
[08/04/21]seed@VM:~/packet_filter$ lsmod | grep seedFilter
seedFilter             16384  0
[08/04/21]seed@VM:~/packet_filter$
```

在容器10.9.0.5 ping和telnet10.9.0.1，都不通过

```
root@19c11f625666:/# ping 10.9.0.1
PING 10.9.0.1 (10.9.0.1) 56(84) bytes of data.
```

```
root@19c11f625666:/# telnet 10.9.0.1
Trying 10.9.0.1...
telnet: Unable to connect to remote host: Connection timed out
root@19c11f625666:/#
```

查看发现数据包被丢弃

```
[ 3912.487214] Dropping telent packet to 10.9.0.1
[ 3913.515466] Dropping telent packet to 10.9.0.1
[ 3914.534730] Dropping telent packet to 10.9.0.1
[ 3915.559471] Dropping telent packet to 10.9.0.1
[ 3916.609280] Dropping telent packet to 10.9.0.1
[ 3917.678165] Dropping telent packet to 10.9.0.1
[ 3918.725502] Dropping telent packet to 10.9.0.1
[ 3919.751259] Dropping telent packet to 10.9.0.1
[ 3920.775303] Dropping telent packet to 10.9.0.1
[ 3921.798707] Dropping telent packet to 10.9.0.1
[ 3922.823386] Dropping telent packet to 10.9.0.1
[ 3923.846708] Dropping telent packet to 10.9.0.1
[ 3924.901895] Dropping telent packet to 10.9.0.1
[ 3925.927033] Dropping telent packet to 10.9.0.1
[ 3926.950479] Dropping telent packet to 10.9.0.1
[ 3927.974915] Dropping telent packet to 10.9.0.1
[ 3928.998429] Dropping telent packet to 10.9.0.1
[ 3930.023247] Dropping telent packet to 10.9.0.1
[ 3931.046954] Dropping telent packet to 10.9.0.1
[ 3932.071251] Dropping telent packet to 10.9.0.1
[ 3933.094796] Dropping telent packet to 10.9.0.1
```

Task2.A

```
root@100b759c7e16:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
root@100b759c7e16:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@100b759c7e16:/# iptables -P OUTPUT DROP
root@100b759c7e16:/# iptables -P INPUT DROP
```

四条规则的作用为：

iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

允许接收icmp请求报文

iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT

允许发出icmp响应报文

iptables -P OUTPUT DROP 丢弃所有发送报文

iptables -P INPUT DROP 丢弃所有接收报文

可以ping通，但无法telnet

```
root@50f0b65f5d53:/# 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=0.060 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.079 ms
^C
--- 10.9.0.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
rtt min/avg/max/mdev = 0.040/0.059/0.079/0.015 ms
root@50f0b65f5d53:/# telnet 10.9.0.11
Trying 10.9.0.11...
^C
```

Task2.B

规则如图

```
root@100b759c7e16:/# iptables -A FORWARD -p icmp --icmp-type echo-reply -i eth0 -j ACCEPT
root@100b759c7e16:/# iptables -A FORWARD -p icmp --icmp-type echo-reply -o eth1 -j ACCEPT
root@100b759c7e16:/# iptables -A FORWARD -p icmp --icmp-type echo-request -i eth1 -j ACCEPT
root@100b759c7e16:/# iptables -A FORWARD -p icmp --icmp-type echo-request -o eth0 -j ACCEPT
root@100b759c7e16:/# iptables -A INPUT   -p icmp --icmp-type echo-request -j ACCEPT
root@100b759c7e16:/# iptables -A OUTPUT  -p icmp --icmp-type echo-reply -j ACCEPT
root@100b759c7e16:/# iptables -P INPUT DROP
root@100b759c7e16:/# iptables -P OUTPUT DROP
root@100b759c7e16:/# iptables -P FORWARD DROP
```

10.9.0.0/24无法ping通 192.168.60.0/24

```
root@50f0b65f5d53:/# 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 ---
9 packets transmitted, 0 received, 100% packet loss, time 8175ms
```

10.9.0.0/24可以ping通两个网段的路由接口

```
root@50f0b65f5d53:/# 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=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.038 ms
```

```
root@50f0b65f5d53:/# 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=0.078 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.085 ms
```

192.168.60.0/24可以反过来ping通10.9.0.0/24

```
root@fec5baf96287:/# 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=0.079 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.049 ms
```

10.9.0.0/24无法telnet通 192.168.60.0/24

```
root@50f0b65f5d53:/# telnet 192.168.60.11
Trying 192.168.60.11...
^C
```

```
root@fec5baf96287:/# telnet 10.9.0.5
Trying 10.9.0.5...
^C
```

Task2.C

规则如图：

```
root@100b759c7e16:/# iptables -A FORWARD -i eth0 -p tcp -d 192.168.60.5 --dport 23 -j ACCEPT
root@100b759c7e16:/# iptables -A FORWARD -o eth1 -p tcp -d 192.168.60.5 --dport 23 -j ACCEPT
root@100b759c7e16:/# iptables -A FORWARD -i eth1 -p tcp -s 192.168.60.5 -j ACCEPT
root@100b759c7e16:/# iptables -A FORWARD -o eth0 -p tcp -s 192.168.60.5 -j ACCEPT
root@100b759c7e16:/# iptables -P FORWARD DROP
```

telnet结果如下：

内部主机间telnet成功

```
c2e9f7f35913  host2-192.168.60.6
[07/24/21]seed@VM:~/.../Labsetup$ docksh c2
root@c2e9f7f35913:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
fec5baf96287 login:
```

外部telnet 192.168.60.5成功

```
root@50f0b65f5d53:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
fec5baf96287 login:
```

外部telnet其他主机失败

```
root@50f0b65f5d53:/# telnet 192.168.60.6
Trying 192.168.60.6...
^C
```

内部无法telent到外部

```
root@fec5baf96287:/# telnet 10.9.0.5
Trying 10.9.0.5...
^C
```

```
root@c2e9f7f35913:/# telnet 10.9.0.5
Trying 10.9.0.5...
^C
```

Task3.A

```
root@100b759c7e16:/# conntrack -L
icmp     1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=42 src=192.168.60.5 dst=10.9.0.5 type
=0 code=0 id=42 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@100b759c7e16:/# conntrack -L
icmp     1 27 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=42 src=192.168.60.5 dst=10.9.0.5 type
=0 code=0 id=42 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
```

ICMP 状态持续时间为30秒

```
root@100b759c7e16:/# conntrack -L
udp      17 27 src=10.9.0.5 dst=192.168.60.5 sport=38323 dport=9090 [UNREPLIED] src=192.168.60.5 d
st=10.9.0.5 sport=9090 dport=38323 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@100b759c7e16:/# conntrack -L
udp      17 24 src=10.9.0.5 dst=192.168.60.5 sport=38323 dport=9090 [UNREPLIED] src=192.168.60.5 d
st=10.9.0.5 sport=9090 dport=38323 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
```

UDP持续时间为30秒

```
root@100b759c7e16:/# conntrack -L
tcp      6 431996 ESTABLISHED src=10.9.0.5 dst=192.168.60.5 sport=44944 dport=9090 src=192.168.60.
5 dst=10.9.0.5 sport=9090 dport=44944 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
```

TCP连接持续时间为43200秒

```
root@100b759c7e16:/# conntrack -L
tcp      6 118 TIME_WAIT src=10.9.0.5 dst=192.168.60.5 sport=44944 dport=9090 src=192.168.60.5 dst
=10.9.0.5 sport=9090 dport=44944 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
```

TCP断开持续时间120秒

TCP持续时间比ICMP和UDP长

Task3.B

规则如下

```
root@100b759c7e16:/# iptables -A FORWARD -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
root@100b759c7e16:/# iptables -A FORWARD -p tcp -m conntrack -i eth1 --ctstate NEW -j ACCEPT
```

192.168.60.5可以telnet到10.9.0.5

```
root@fec5baf96287:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
50f0b65f5d53 login:
```

192.168.60.6也可以telnet到10.9.0.5

```
root@c2e9f7f35913:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
50f0b65f5d53 login:
```

Task4

```
iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -j ACCEPT
```

```
--- 10.9.0.5 ping statistics ---
30 packets transmitted, 30 received, 0% packet loss, time 29688ms
rtt min/avg/max/mdev = 0.046/0.058/0.198/0.028 ms
```

只有一条规则，不具有拦截效果

```
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=0.114 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.087 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.109 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=63 time=0.049 ms
64 bytes from 10.9.0.5: icmp_seq=7 ttl=63 time=0.049 ms
64 bytes from 10.9.0.5: icmp_seq=13 ttl=63 time=0.065 ms
64 bytes from 10.9.0.5: icmp_seq=19 ttl=63 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=25 ttl=63 time=0.051 ms
64 bytes from 10.9.0.5: icmp_seq=31 ttl=63 time=0.063 ms
64 bytes from 10.9.0.5: icmp_seq=37 ttl=63 time=0.051 ms
64 bytes from 10.9.0.5: icmp_seq=43 ttl=63 time=0.047 ms
64 bytes from 10.9.0.5: icmp_seq=48 ttl=63 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=54 ttl=63 time=0.051 ms
^C
--- 10.9.0.5 ping statistics ---
58 packets transmitted, 14 received, 75.8621% packet loss, time 58357ms
rtt min/avg/max/mdev = 0.047/0.062/0.114/0.022 ms
```

两条规则时，可以看到第5条后开始出现拦截

原因是当没有第二条规则时，第一条规则过滤的包也默认ACCEPT，只有加入第二条后才被DROP。

Task5

开始代码如下

```
root@100b759c7e16:/# 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
```

负载均衡配置如下：

```
root@26287811e9cb:/# 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@26287811e9cb:/# 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@26287811e9cb:/# iptables -t nat -A PREROUTING -p udp --dport 8080  -j DNAT --to-destination 192.16
8.60.6:8080
```

主机上：

```
echo hello|nc -u 10.9.0.11 8080

echo hello1|nc -u 10.9.0.11 8080
echo hello_1|nc -u 10.9.0.11 8080

echo hello_2|nc -u 10.9.0.11 8080

echo hello_3|nc -u 10.9.0.11 8080

echo hello_4|nc -u 10.9.0.11 8080
echo hello_4|nc -u 10.9.0.11 8080

echo hello_5|nc -u 10.9.0.11 8080

echo hello_6|nc -u 10.9.0.11 8080
echo hello_6|nc -u 10.9.0.11 8080

echo hello_7|nc -u 10.9.0.11 8080

echo hello_8|nc -u 10.9.0.11 8080
```

在服务器192.168.60.5上监听8080端口

```
hello
hello_2
hello_4
hello_6
hello_9
```

服务器192.168.60.6上监听8080端口

```
root@49e64dea0623:/# nc -luk 8080
hello_3
hello_7
```

在路由器上利用iptables命令，采用random模式创建负载均衡规则

在Host A上

```
root@137e7980c0ce:/# echo hello_1|nc -u 10.9.0.11 8080
^C
root@137e7980c0ce:/# echo hello_2|nc -u 10.9.0.11 8080
^C
root@137e7980c0ce:/# echo hello_3|nc -u 10.9.0.11 8080
^C
root@137e7980c0ce:/# echo hello_4|nc -u 10.9.0.11 8080
^C
root@137e7980c0ce:/# echo hello_5|nc -u 10.9.0.11 8080
^C
root@137e7980c0ce:/# echo hello_6|nc -u 10.9.0.11 8080
^C
root@137e7980c0ce:/# echo hello_7|nc -u 10.9.0.11 8080
^C
root@137e7980c0ce:/# echo hello_8|nc -u 10.9.0.11 8080
^C
root@137e7980c0ce:/# echo hello_9|nc -u 10.9.0.11 8080
root@137e7980c0ce:/# echo hello_9|nc -u 10.9.0.11 8080
```

```
root@d998244af73c:/# nc -luk 8080
hello_1
hello_4
hello_5
hello_7
hello_9
```

```
root@552e72b0412e:/# nc -luk 8080
hello_2
hello_3
hello_8
```

```
root@49e64dea0623:/# nc -luk 8080
hello_6
```

可以发现，等概率发送数据，每个主机收到的数量各不相同，但当样本数量足够多时， 趋于平均。