哈尔滨工业大学(深圳)

# 《网络与系统安全》 实验报告

实验六

防火墙 实验


学 院: 计算机科学与技术

姓 名: 宗晴

学 号: 200110513

专 业: 计算机


日 期: 2023 年 6 月

1. Task1：加载 seedFilter 模块，执行 dig dig @8.8.8.8 www.example.com，卸载 seedFilter 后再执行 dmesg 命令查看内核日志，把日志信息中加载、卸载 seefFilter 模块以及阻止 UDP 数据包的信息截图，并进行分析说明。

先执行 dig 命令查看，发现有回复，如下图所示：

```
[06/06/23]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: 48061
;; 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.        19450   IN      A       93.184.216.34

;; Query time: 64 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Jun 06 09:01:33 EDT 2023
;; MSG SIZE  rcvd: 60
```

使用 Labsetup/Files/packet_filter 下的代码，阻止 IP 地址是 8.8.8.8 和端口为 53 的 UDP 数据包。加载成功后再执行 dig @8.8.8.8 www.example.com 命令，查看结果如下图所示，已经得不到任何响应了，说明防火墙设置成功：

```
[06/06/23]seed@VM:~/.../kernel_module$ cd ..
[06/06/23]seed@VM:~/.../Files$ cd packet_filter/
[06/06/23]seed@VM:~/.../packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Firewall/Labsetup/Files
/packet_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/Firewall/Labsetup/Files/packet_filter/seedFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/seed/Firewall/Labsetup/Files/packet_filter/seedFilter.mod.o
  LD [M]  /home/seed/Firewall/Labsetup/Files/packet_filter/seedFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[06/06/23]seed@VM:~/.../packet_filter$ sudo insmod seedFilter.ko
[06/06/23]seed@VM:~/.../packet_filter$ lsmod | grep seedFilter
seedFilter             16384  0
[06/06/23]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
```

卸载 seedFilter 模块成功后再执行 dig @8.8.8.8 www.example.com 命令，又可以收到回复了：

```
[06/06/23]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
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7328
;; 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.        11304   IN      A       93.184.216.34

;; Query time: 47 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Jun 06 09:09:25 EDT 2023
;; MSG SIZE  rcvd: 60
```

使用 dmesg 命令查看内核日志信息，可以看到注册和卸载的信息，以及防火墙阻止后丢掉的数据包：

```
[  430.711723] br-3c269c91d7e0: port 4(veth847b3ff) entered blocking state
[  430.711724] br-3c269c91d7e0: port 4(veth847b3ff) entered forwarding state
[  504.411985] hello: module verification failed: signature and/or required key mi
ssing - tainting kernel
[  504.413833] Hello World!
[  526.964122] Bye-bye World!.
[  670.886735] Registering filters.
[  675.905110] *** LOCAL_OUT
[  675.905158]     10.0.2.15  --> 23.195.91.154 (TCP)
[  675.905247] *** LOCAL_OUT
[  675.905265]     10.0.2.15  --> 23.195.91.154 (TCP)
[  675.905304] *** LOCAL_OUT
[  675.905321]     10.0.2.15  --> 23.195.91.154 (TCP)
[  686.145158] *** LOCAL_OUT
[  686.145206]     10.0.2.15  --> 23.195.91.154 (TCP)
[  686.145294] *** LOCAL_OUT
[  686.145313]     10.0.2.15  --> 23.195.91.154 (TCP)
[  686.145351] *** LOCAL_OUT
[  686.145367]     10.0.2.15  --> 23.195.91.154 (TCP)
[  696.385874] *** LOCAL_OUT
[  696.385880]     10.0.2.15  --> 23.195.91.154 (TCP)
[  696.385950] *** LOCAL_OUT
[  696.385953]     10.0.2.15  --> 23.195.91.154 (TCP)
[  696.385977] *** LOCAL_OUT
[  696.385979]     10.0.2.15  --> 23.195.91.154 (TCP)
[  697.508048] *** LOCAL_OUT
[  697.508050]     127.0.0.1  --> 127.0.0.1 (UDP)
[  697.508136] *** LOCAL_OUT
[  697.508137]     10.0.2.15  --> 8.8.8.8 (UDP)
[  697.508139] *** Dropping 8.8.8.8 (UDP), port 53
[  702.510076] *** LOCAL_OUT
[  702.510082]     10.0.2.15  --> 8.8.8.8 (UDP)
```

```
[  697.508137]     10.0.2.15  --> 8.8.8.8 (UDP)
[  697.508139] *** Dropping 8.8.8.8 (UDP), port 53
[  702.510076] *** LOCAL_OUT
[  702.510082]     10.0.2.15  --> 8.8.8.8 (UDP)
[  702.510102] *** Dropping 8.8.8.8 (UDP), port 53
[  706.626208] *** LOCAL_OUT
[  706.626210]     10.0.2.15  --> 23.195.91.154 (TCP)
[  706.626238] *** LOCAL_OUT
[  706.626239]     10.0.2.15  --> 23.195.91.154 (TCP)
[  706.626246] *** LOCAL_OUT
[  706.626247]     10.0.2.15  --> 23.195.91.154 (TCP)
[  707.514276] *** LOCAL_OUT
[  707.514277]     10.0.2.15  --> 8.8.8.8 (UDP)
[  707.514285] *** Dropping 8.8.8.8 (UDP), port 53
[  713.248951] *** LOCAL_OUT
[  713.248953]     10.0.2.15  --> 34.149.100.209 (TCP)
[  713.530709] *** LOCAL_OUT
[  713.530754]     10.0.2.15  --> 34.149.100.209 (TCP)
[  716.867493] *** LOCAL_OUT
[  716.867510]     10.0.2.15  --> 23.195.91.154 (TCP)
[  716.867540] *** LOCAL_OUT
[  716.867545]     10.0.2.15  --> 23.195.91.154 (TCP)
[  716.867555] *** LOCAL_OUT
[  716.867559]     10.0.2.15  --> 23.195.91.154 (TCP)
[  718.250147] *** LOCAL_OUT
[  718.250149]     10.0.2.15  --> 34.117.121.53 (TCP)
[  718.499455] *** LOCAL_OUT
[  718.499495]     10.0.2.15  --> 34.117.121.53 (TCP)
[  724.804190] The filters are being removed.
```

2. Task2：阻止 TCP 端口和 PING，把增加和修改的代码截图，并在卸载模块后将 dmesg 的日志信息的截图，并分析说明原因。

增加函数 blockICMP：

```
unsigned int blockICMP(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 (ICMP)\n", &(iph->daddr));
            return NF_DROP;
        }
   }
   return NF_ACCEPT;
}
```

增加函数 blockTCP：

```
unsigned int blockTCP(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 (TCP), port %d\n", &(iph->daddr), port);
            return NF_DROP;
        }
   }
   return NF_ACCEPT;
}
```

增加两个钩子 hook3，hook4：

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

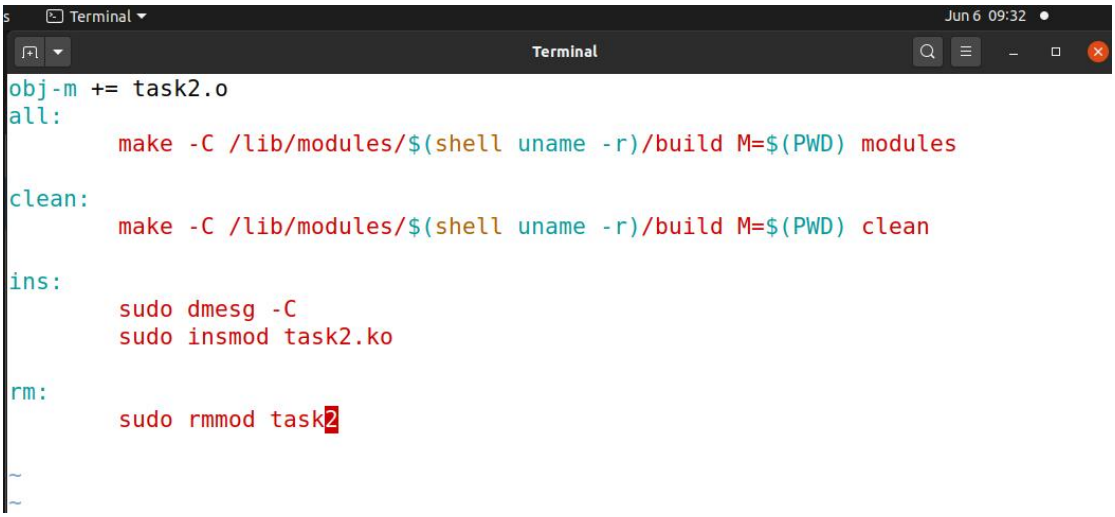
在 registerFilter 函数中注册：

```
123 int registerFilter(void) {
124     printk(KERN_INFO "Registering filters.\n");
125 
126     hook1.hook = printInfo;
127     hook1.hooknum = NF_INET_LOCAL_OUT;
128     hook1.pf = PF_INET;
129     hook1.priority = NF_IP_PRI_FIRST;
130     nf_register_net_hook(&init_net, &hook1);
131 
132     hook2.hook = blockUDP;
133     hook2.hooknum = NF_INET_POST_ROUTING;
134     hook2.pf = PF_INET;
135     hook2.priority = NF_IP_PRI_FIRST;
136     nf_register_net_hook(&init_net, &hook2);
137 
138     hook3.hook = blockICMP;
139     hook3.hooknum = NF_INET_LOCAL_OUT;
140     hook3.pf = PF_INET;
141     hook3.priority = NF_IP_PRI_FIRST;
142     nf_register_net_hook(&init_net, &hook3);
143 
144     hook4.hook = blockTCP;
145     hook4.hooknum = NF_INET_POST_ROUTING;
146     hook4.pf = PF_INET;
147     hook4.priority = NF_IP_PRI_FIRST;
148     nf_register_net_hook(&init_net, &hook4);
149 
150     return 0;
151 }
```

在 removeFilter 函数中删除:

```
153 void removeFilter(void) {
154     printk(KERN_INFO "The filters are being removed.\n");
155     nf_unregister_net_hook(&init_net, &hook1);
156     nf_unregister_net_hook(&init_net, &hook2);
157     nf_unregister_net_hook(&init_net, &hook3);
158     nf_unregister_net_hook(&init_net, &hook4);
159 }
```

修改 Makefile ， 将里面的 seedFilter 相关的内容修改为 task2 的内容:




分别输入 ping 10.9.0.1 和 telnet 10.9.0.1 命令，发现没有响应：

```
[06/06/23]seed@VM:~/.../packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Firewall/Labsetup/Files/p
acket_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/Firewall/Labsetup/Files/packet_filter/task2.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/seed/Firewall/Labsetup/Files/packet_filter/task2.mod.o
  LD [M]  /home/seed/Firewall/Labsetup/Files/packet_filter/task2.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[06/06/23]seed@VM:~/.../packet_filter$ sudo insmod task2.ko

[06/06/23]seed@VM:~/.../packet_filter$ lsmod | grep task2
task2                  16384  0
[06/06/23]seed@VM:~/.../packet_filter$ ping 10.9.0.1
PING 10.9.0.1 (10.9.0.1) 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
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 10.9.0.1 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5108ms


[06/06/23]seed@VM:~/.../packet_filter$ telnet 10.9.0.1
Trying 10.9.0.1...
telnet: Unable to connect to remote host: Connection timed out
[06/06/23]seed@VM:~/.../packet_filter$
```

卸载 task2 模块成功后再执行 ping 10.9.0.1 和 telnet 10.9.0.1 命令，恢复正常：

```
[06/06/23]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=0.024 ms
64 bytes from 10.9.0.1: icmp_seq=2 ttl=64 time=0.071 ms
64 bytes from 10.9.0.1: icmp_seq=3 ttl=64 time=0.072 ms
64 bytes from 10.9.0.1: icmp_seq=4 ttl=64 time=0.029 ms
64 bytes from 10.9.0.1: icmp_seq=5 ttl=64 time=0.032 ms
^C
--- 10.9.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4072ms
rtt min/avg/max/mdev = 0.024/0.045/0.072/0.021 ms
[06/06/23]seed@VM:~/.../packet_filter$ telnet 10.9.0.1
Trying 10.9.0.1...
Connected to 10.9.0.1.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
VM 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

681 updates can be installed immediately.
494 of these updates are security updates.
To see these additional updates run: apt list --upgradable
```

```
681 updates can be installed immediately.
494 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Jun  6 09:12:40 EDT 2023 from VM on pts/2
[06/06/23]seed@VM:~$
```

dmesg 的日志信息如下，发现有防火墙阻止后丢掉的数据包：

```
[ 2553.175116] Registering filters.
[ 2553.177058] *** LOCAL_OUT
[ 2553.177059]     10.9.0.1  --> 10.0.2.15 (TCP)
[ 2553.177076] *** LOCAL_OUT
[ 2553.177076]     10.9.0.1  --> 10.9.0.1 (TCP)
[ 2553.177077] *** Dropping 10.9.0.1 (TCP), port 23
[ 2553.384203] *** LOCAL_OUT
[ 2553.384205]     10.9.0.1  --> 10.0.2.15 (TCP)
[ 2553.384234] *** LOCAL_OUT
[ 2553.384235]     10.9.0.1  --> 10.9.0.1 (TCP)
[ 2553.384237] *** Dropping 10.9.0.1 (TCP), port 23
[ 2553.591976] *** LOCAL_OUT
[ 2553.592021]     10.9.0.1  --> 10.0.2.15 (TCP)
[ 2553.592717] *** LOCAL_OUT
[ 2553.592722]     10.9.0.1  --> 10.9.0.1 (TCP)
[ 2553.592729] *** Dropping 10.9.0.1 (TCP), port 23
[ 2554.003760] *** LOCAL_OUT
[ 2554.003792]     10.9.0.1  --> 10.0.2.15 (TCP)
[ 2554.004333] *** LOCAL_OUT
[ 2554.004336]     10.9.0.1  --> 10.9.0.1 (TCP)
[ 2554.004341] *** Dropping 10.9.0.1 (TCP), port 23
[ 2554.677052] *** LOCAL_OUT
[ 2554.677057]     10.9.0.1  --> 10.9.0.1 (TCP)
[ 2554.677071] *** Dropping 10.9.0.1 (TCP), port 23
[ 2554.835905] *** LOCAL_OUT
[ 2613.045520] *** Dropping 10.9.0.1 (TCP), port 23
[ 2619.701272] *** LOCAL_OUT
[ 2619.701273]     10.9.0.1  --> 10.9.0.1 (TCP)
[ 2619.701278] *** Dropping 10.9.0.1 (TCP), port 23
[ 2621.601610] *** Dropping 10.9.0.1 (ICMP)
[ 2622.336300] *** LOCAL_OUT
[ 2622.336305]     10.0.2.15  --> 10.248.98.30 (UDP)
[ 2622.613490] *** Dropping 10.9.0.1 (ICMP)
[ 2623.637531] *** Dropping 10.9.0.1 (ICMP)
[ 2624.661489] *** Dropping 10.9.0.1 (ICMP)
[ 2625.685974] *** Dropping 10.9.0.1 (ICMP)
[ 2626.709597] *** Dropping 10.9.0.1 (ICMP)
[ 2633.013684] *** LOCAL_OUT
[ 2633.013731]     10.9.0.1  --> 10.9.0.1 (TCP)
[ 2633.013760] *** Dropping 10.9.0.1 (TCP), port 23
[ 2634.082459] *** LOCAL_OUT
[ 2634.082464]     127.0.0.1  --> 127.0.0.53 (UDP)
[ 2634.082786] *** LOCAL_OUT
[ 2634.082789]     127.0.0.53  --> 127.0.0.1 (UDP)
[ 2634.113472] *** LOCAL_OUT
[ 2634.113478]     127.0.0.1  --> 127.0.0.53 (UDP)
[ 2634.114425] *** LOCAL_OUT
[ 2634.114430]     127.0.0.53  --> 127.0.0.1 (UDP)
[ 2637.716830] *** LOCAL_OUT
[ 2637.716832]     10.9.0.1  --> 10.9.0.1 (TCP)
[ 2637.716839] *** Dropping 10.9.0.1 (TCP), port 23
[ 2638.741778] *** LOCAL_OUT
```

ping 10.9.0.1 和 telnet 10.9.0.1 均没有响应，是因为在 task2 中实现了防火墙，从而拦截并丢弃了 IP 地址是 10.9.0.1 的 ICMP 数据包以及 IP 地址为 10.9.0.1、端口号为 23 的 TCP 数据包。

3. Task3：保护 Router，将配置 iptables 规则前后 ping 和 telnet 的连通性测试结果截图，并分析说明原因。

初始时，直接在容器 A 中执行 ping 10.9.0.11（Router IP）和 telnet 10.9.0.11（Router IP）命令，发现均可以连通：

```
[06/06/23]seed@VM:~$ dockps
5d8cec424cdb  seed-router
f9d8f9540d20  host2-192.168.60.6
6463209e0e85  host1-192.168.60.5
d78fd61c0dfc  host3-192.168.60.7
636271af4666  hostA-10.9.0.5
[06/06/23]seed@VM:~$ docksh 63
root@636271af4666:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of
root@636271af4666:/# 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=0.038 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.109 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.040 ms
^C
--- 10.9.0.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4073ms
rtt min/avg/max/mdev = 0.038/0.056/0.109/0.026 ms
root@636271af4666:/# telnet 10.9.0.11
Trying 10.9.0.11...
Connected to 10.9.0.11.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
5d8cec424cdb 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.
```

```
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@5d8cec424cdb:~$
```

然后，设置 Router 允许 icmp 类型协议的应答，其他没有设置的协议类型默认拒绝：

```
[06/06/23]seed@VM:~/.../packet_filter$ dockps
5d8cec424cdb  seed-router
f9d8f9540d20  host2-192.168.60.6
6463209e0e85  host1-192.168.60.5
d78fd61c0dfc  host3-192.168.60.7
636271af4666  hostA-10.9.0.5
[06/06/23]seed@VM:~/.../packet_filter$ docksh 5d
root@5d8cec424cdb:/#
root@5d8cec424cdb:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
root@5d8cec424cdb:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@5d8cec424cdb:/# iptables -P OUTPUT DROP
root@5d8cec424cdb:/# iptables -P INPUT DROP
root@5d8cec424cdb:/#
```

在 HostA 容器中再次执行 ping 10.9.0.11（Router IP）和 telnet 10.9.0.11（Router IP）命令，发现 ping 10.9.0.11 可以连通，而 telnet 10.9.0.11 不行：

```
[06/06/23]seed@VM:~/.../packet_filter$ docksh 63
root@636271af4666:/# 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=0.044 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.028 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.050 ms
^C
--- 10.9.0.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3068ms
rtt min/avg/max/mdev = 0.028/0.043/0.050/0.009 ms
root@636271af4666:/# telnet 10.9.0.11
Trying 10.9.0.11...
telnet: Unable to connect to remote host: Connection timed out
root@636271af4666:/#
```

ping 10.9.0.11 可以连通，是因为防火墙允许 icmp-type echo-request 数据包以及 icmp-type echo-reply 数据包通过。而 telnet 10.9.0.11 不行，是因为防火墙设置其他没有设置的协议类型都默认拒绝，又由于 telnet 使用了 TCP

协议，因此相关的报文将被防火墙拦截并丢弃。

4、Task4：保护内网，将配置 iptables 规则前后 ping 的连通性测试结果截图，并分析说明原因。

最初时，在 HostA 容器中执行 ping 192.168.60.5（内网 host1 IP）和 telent 192.168.60.5（内网 host1 IP）命令，发现均可以连通：

```
[06/06/23]seed@VM:~/.../packet_filter$ dockps
5d8cec424cdb  seed-router
f9d8f9540d20  host2-192.168.60.6
6463209e0e85  host1-192.168.60.5
d78fd61c0dfc  host3-192.168.60.7
636271af4666  hostA-10.9.0.5
```

```
[06/06/23]seed@VM:~/.../packet_filter$ docksh 63
root@636271af4666:/# 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=0.074 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.133 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.134 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.051 ms
^C
--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3064ms
rtt min/avg/max/mdev = 0.051/0.098/0.134/0.036 ms
root@636271af4666:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.

Ubuntu 20.04.1 LTS

6463209e0e85 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.
Last login: Tue Jun  6 14:21:52 UTC 2023 from seed-router.net-192.168.60.0 on pts/1
seed@6463209e0e85:~$
```

在 Router 容器中配置 iptables 之后，在 HostA 容器中执行 ping 192.168.60.5（内网 Host1 IP）和 telent 192.168.60.5（内网 Host1 IP）命令，发现均无法连通：

```
[06/06/23]seed@VM:~/.../packet_filter$ docksh 63
root@636271af4666:/# 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 ---
28 packets transmitted, 0 received, 100% packet loss, time 27631ms

root@636271af4666:/# telnet 192.168.60.5
Trying 192.168.60.5...
telnet: Unable to connect to remote host: Connection timed out
root@636271af4666:/#
```

这是因为配置规则 -A FORWARD -i eth0 -p icmp --icmp-type echo-request -j DROP 表示从外网接口 eth0 来的 icmp-type echo-request 数据包均需要被拦截并丢弃。因此，HostA 无法 ping 通内网的 Host1。

同时，配置规则-P FORWARD DROP 表示除上述三种类别外的其它数据包都将被拦截并丢弃。telnet 使用的是 TCP 协议，因此数据包被丢弃，无法连通。

在 Host1（192.168.60.5）中分别执行 ping 192.168.60.11 和 ping 10.9.0.5（HostA），观察发现均可以连通：

```
[06/06/23]seed@VM:~/.../packet_filter$ dockps
5d8cec424cdb  seed-router
f9d8f9540d20  host2-192.168.60.6
6463209e0e85  host1-192.168.60.5
d78fd61c0dfc  host3-192.168.60.7
636271af4666  hostA-10.9.0.5
```

```
seed@VM: ~/.../Labsetup   seed@VM: ~/.../packet_filter   seed@VM: ~/.../packet_filter   seed@VM: ~/.../packet_filter
[06/06/23]seed@VM:~/.../packet_filter$ docksh 64
root@6463209e0e85:/# 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=0.043 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.104 ms
64 bytes from 192.168.60.11: icmp_seq=3 ttl=64 time=0.109 ms
64 bytes from 192.168.60.11: icmp_seq=4 ttl=64 time=0.036 ms
^C
--- 192.168.60.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.036/0.073/0.109/0.033 ms
root@6463209e0e85:/# 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=0.060 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.053 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.051 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.134 ms
^C
--- 10.9.0.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3074ms
rtt min/avg/max/mdev = 0.051/0.074/0.134/0.034 ms
root@6463209e0e85:/#
```

这是因为配置规则 -A FORWARD -i eth1 -p icmp --icmp-type echo-request -j ACCEPT 表示从内网接口 eth1 来的 icmp-type echo-request 数据包被允许通过。配置规则 -A FORWARD -p icmp --icmp-type echo-reply -j ACCEPT 表示其它的 icmp-type echo-reply 数据包均被允许通过。

所以，执行 ping 192.168.60.11 时，Host1 发送 icmp-type echo-request 数据包到 Router（192.168.60.11），防火墙允许该数据包通过；Router 收到请求并回复 icmp-type echo-reply 数据包，防火墙也允许该数据包通过，所以 Host1 可以收到回复。 因此可以连通。

同理，执行 ping 10.9.0.5 时，Host1 和 HostA 之间的数据包也被防火墙允许通过，所以 Host1 和 HostA 也能连通。