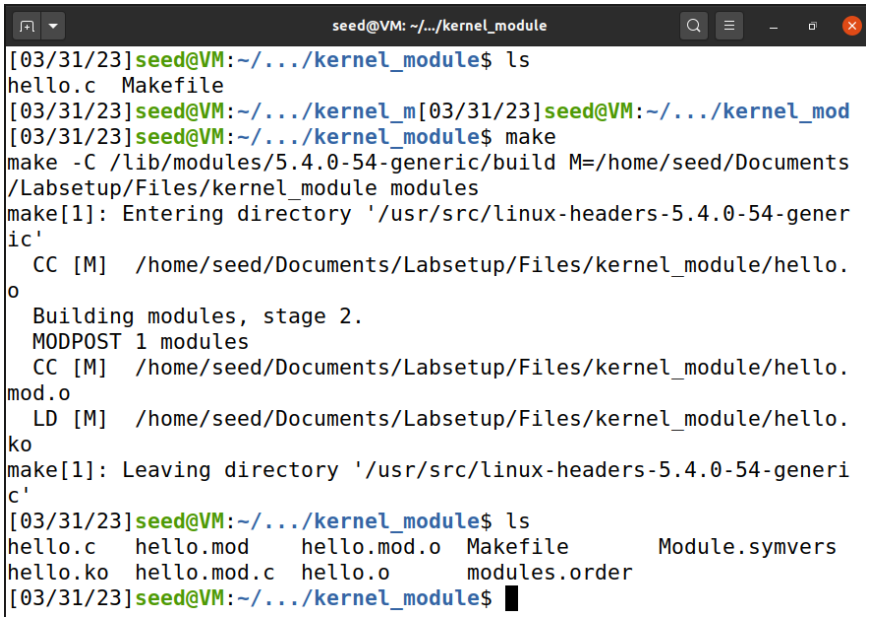
FIREWALL EXPLORATION LAB

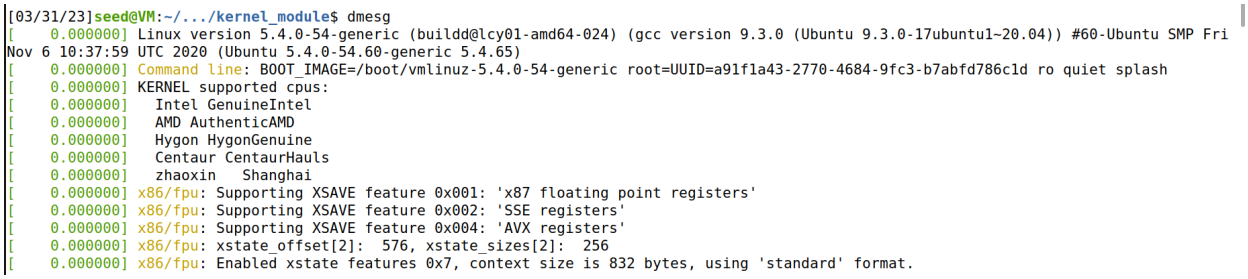
3.1:-Task 1.A: Implement a Simple Kernel Module

As shown above, hello.c file which contains the above-mentioned C code is accompanied by the makefile.

Invoking make will run the makefile to compile the hello.c program to generate below mentioned outputs.



displaying the list of modules:



```
seed@VM: ~/.../kernel_module
[03/31/23]seed@VM:~/.../kernel_module$ lsmod | grep hello
[03/31/23]seed@VM:~/.../kernel_module$ lsmod
Module                  Size  Used by
sctp                  344064  16
xt_nat                 16384  24
xt_tcpudp              20480  24
veth                   28672  0
xt_conntrack           16384  3
xt_MASQUERADE          20480  3
nf_conntrack_netlink    45056  0
nfnetlink              16384  2 nf_conntrack_netlink
xfrm_user              36864  1
xfrm_algo              16384  1 xfrm_user
xt_addrtype            16384  2
iptable_filter         16384  1
iptable_nat            16384  7
nf_nat                 40960  3 xt_nat,iptable_nat,xt_MASQUERADE
nf_conntrack          139264  5 xt_conntrack,nf_nat,xt_nat,nf_conntrack_netlink,xt_MASQUERADE
nf_defrag_ipv6         24576  1 nf_conntrack
nf_defrag_ipv4         16384  1 nf_conntrack
libcrc32c              16384  3 nf_conntrack,nf_nat,sctp
bpfilter               32768  0
br_netfilter           28672  0
bridge                176128  1 br_netfilter
stp                    16384  1 bridge
llc                    16384  2 bridge,stp
aufs                  262144  0
vboxsf                 81920  0
overlay               114688  6
nls_iso8859_1          16384  1
binfmt_misc            24576  1
intel_rapl_msr         20480  0
```

inserted the following hello.ko module and listed it below:

```
[03/31/23]seed@VM:~/.../kernel_module$ sudo insmod hello.ko
[03/31/23]seed@VM:~/.../kernel_module$ lsmod | grep hello
hello                  16384  0
[03/31/23]seed@VM:~/.../kernel_module$
```

use of dmesg command is to check the messages created by the module:

```
seed@VM: ~/.../kernel_module
[03/31/23]seed@VM:~/.../kernel_module$ dmesg
[10205.828307] Hello World!
```

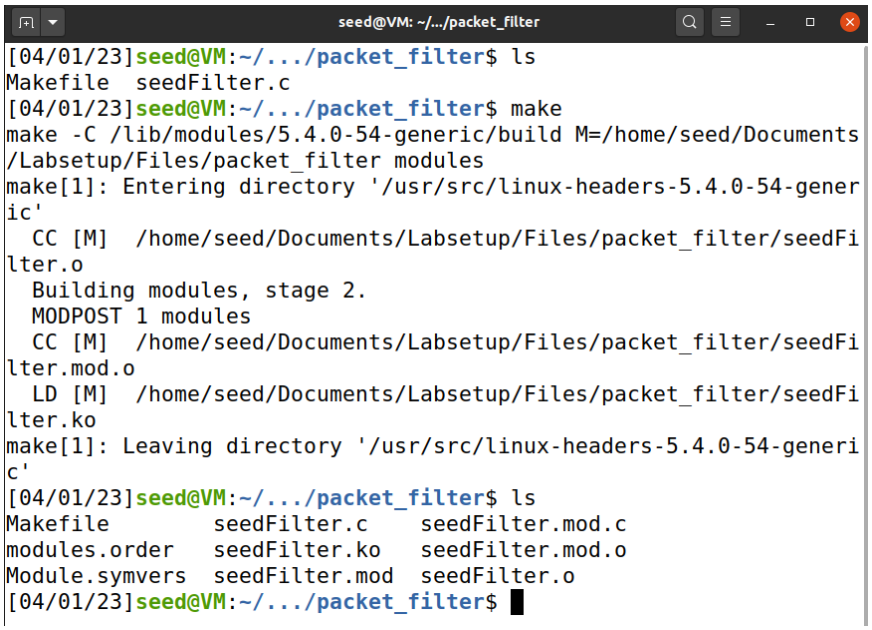
removing the hello.ko module as follows and check the message using dmesg:

```
seed@VM: ~/.../kernel_module
[03/31/23]seed@VM:~/.../kernel_module$ sudo rmmod hello.ko
[03/31/23]seed@VM:~/.../kernel_module$ dmesg
[10205.828307] Hello World!
[10732.089691] Bye-bye World!.
[03/31/23]seed@VM:~/.../kernel_module$
```

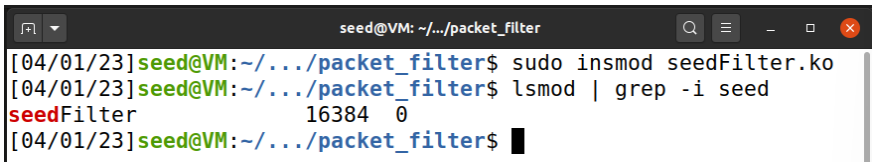
3.2 Task 1.B: Implement a Simple Firewall Using Netfilter

TASK1:

Compiled the seedFilter.c file using the make command as follows:



Inserted the seedFilter.ko



After running the command dig @8.8.8.8 www.example.com after adding sendFilter.ko, the connection could not be established. The image below shows message outputs, all requests are being dropped

This can be shown when the UDP packets are retried to be picked before getting dropped.

The above screenshot shows the outputs of invoking the make command and the corresponding lsmod output for seed.

TASK2:
For this task create a new file named seedPrint and add its executable kernel to the Makefile:

```
#obj-m += seedFilter.o
obj-m += seedPrint.o
all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

ins:
	sudo dmesg -C
	sudo insmod seedFilter.ko

rm:
	sudo rmmod seedFilter
```

adding 3 more hooks to the code:

Open | seedPrint.c ~/Documents/Labsetup/Files/packet_filter | Save

```
#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, hook3, hook4, hook5;
```

Set each hook to NF_INET_PRE_ROUTING, NF_INET_LOCAL_IN, NF_INET_FORWARD, NF_INET_LOCAL_OUT, NF_INET_POST_ROUTING inside the code.

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

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

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

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

    //NF_INET_FORWARD
    hook5.hook = printInfo;
    hook5.hooknum = NF_INET_FORWARD;
    hook5.pf = PF_INET;
```

Under the remove filter function, added the unregister for the 3 added hooks:

seedPrint.c
~/Documents/Labsetup/Files/packet_filter

```
 97    //NF_INET_LOCAL_IN
 98    hook4.hook = printInfo;
 99    hook4.hooknum = NF_INET_LOCAL_IN;
100    hook4.pf = PF_INET;
101    hook4.priority = NF_IP_PRI_FIRST;
102    nf_register_net_hook(&init_net, &hook4);
103
104    //NF_INET_FORWARD
105    hook5.hook = printInfo;
106    hook5.hooknum = NF_INET_FORWARD;
107    hook5.pf = PF_INET;
108    hook5.priority = NF_IP_PRI_FIRST;
109    nf_register_net_hook(&init_net, &hook5);
110
111    return 0;
112 }
113
114 void removeFilter(void) {
115    printk(KERN_INFO "The filters are being removed.\n");
116    nf_unregister_net_hook(&init_net, &hook1);
117    nf_unregister_net_hook(&init_net, &hook2);
118    nf_unregister_net_hook(&init_net, &hook3);
119    nf_unregister_net_hook(&init_net, &hook4);
120    nf_unregister_net_hook(&init_net, &hook5);
121 }
```

compiled the file into a kernel module using make command and inserted the seedPrint kernel module as follows:

```
seed@VM: ~/.../packet_filter
[04/01/23]seed@VM:~/.../packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Documents
/Labsetup/Files/packet_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-gener
ic'
  CC [M]  /home/seed/Documents/Labsetup/Files/packet_filter/seedPr
int.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/seed/Documents/Labsetup/Files/packet_filter/seedPr
int.mod.o
  LD [M]  /home/seed/Documents/Labsetup/Files/packet_filter/seedPr
int.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generi
c'
[04/01/23]seed@VM:~/.../packet_filter$ ls
Makefile        seedFilter.ko     seedFilter.o   seedPrint.mod.c
modules.order   seedFilter.mod    seedPrint.c    seedPrint.mod.o
Module.symvers  seedFilter.mod.c  seedPrint.ko   seedPrint.o
seedFilter.c    seedFilter.mod.o  seedPrint.mod
[04/01/23]seed@VM:~/.../packet_filter$ sudo insmod seedPrint.ko
[04/01/23]seed@VM:~/.../packet_filter$
```

We get the following messages when the kernel is registered:

```
[32294.757307] The filters are being removed.
[33241.535455] seedPrint file: Registering filters.
```

Again we make use of the dig command to check the generated UDP packets and the different functions being invoked:

```
seed@VM: ~/.../packet_filter
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached

[04/01/23]seed@VM:~/.../packet_filter$ dig @8.8.8.8 www.example.co
m

; <<>> 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

[04/01/23]seed@VM:~/.../packet_filter$
```

```
seed@VM: ~/.../kernel_module
[18790.707259]      127.0.0.1  --> 127.0.0.1 (UDP)
[18790.707272] *** PRE_ROUTING
[18790.707273]      127.0.0.1  --> 127.0.0.1 (UDP)
[18790.707275] *** LOCAL_IN
[18790.707276]      127.0.0.1  --> 127.0.0.1 (UDP)
[18790.707372] *** LOCAL_OUT
[18790.707373]      10.0.2.5  --> 8.8.8.8 (UDP)
[18790.707374] *** LOCAL_OUT
[18790.707375]      10.0.2.5  --> 8.8.8.8 (UDP)
[18790.707379] *** POST_ROUTING
[18790.707380]      10.0.2.5  --> 8.8.8.8 (UDP)
[18790.707381] *** Dropping 8.8.8.8 (UDP), port 53
[18795.705135] *** LOCAL_OUT
```

We can see that the LOCAL_OUT, LOCAL_IN, POST_ROUTING and PRE_ROUTING functions were invoked as the UDP packets were generated except the FORWARD function

removing the module:

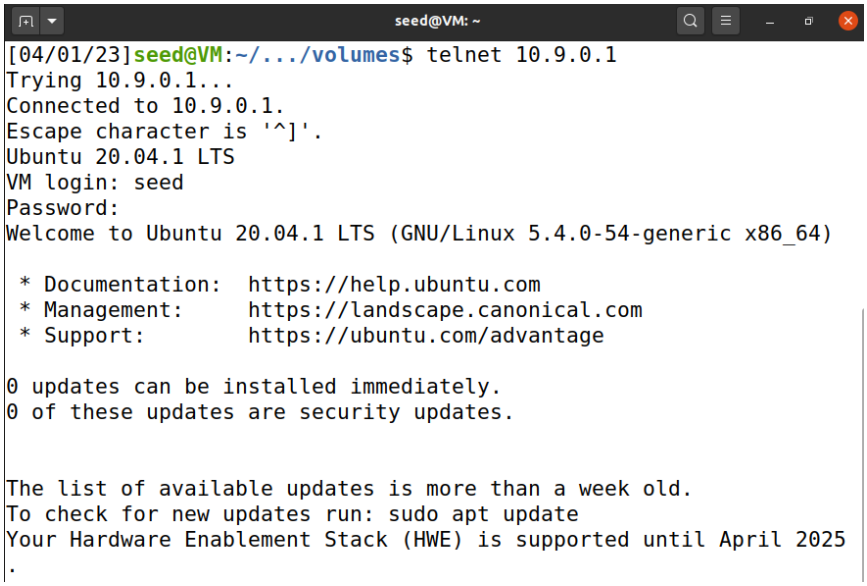
```
[04/01/23]seed@VM:~/.../packet_filter$ sudo rmmod seedPrint.ko
[04/01/23]seed@VM:~/.../packet_filter$
```

```
[33790.063378] seedPrint file: The filters are being removed.
```

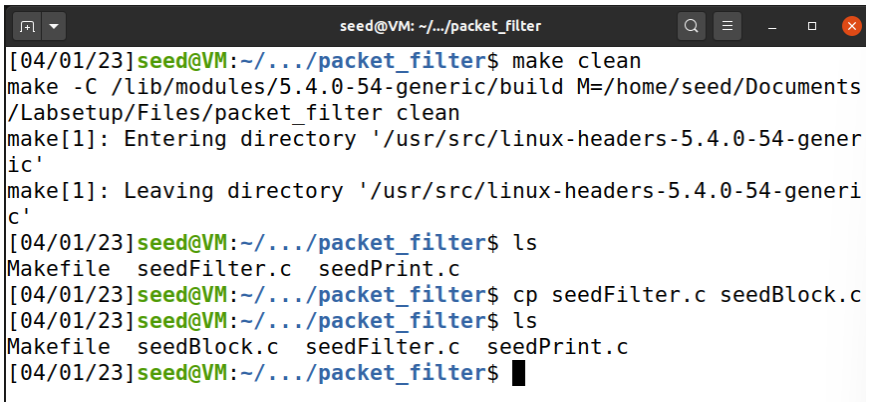
TASK3:

In order to demonstrate this scenario, we need to check first the result of ping 10.9.0.1 and telnet 10.9.0.1. the image below shows the result

```
seed@VM: ~
[04/01/23]seed@VM:~/.../volumes$ 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=0.043 ms
64 bytes from 10.9.0.1: icmp_seq=2 ttl=64 time=0.086 ms
64 bytes from 10.9.0.1: icmp_seq=3 ttl=64 time=0.070 ms
64 bytes from 10.9.0.1: icmp_seq=4 ttl=64 time=0.087 ms
64 bytes from 10.9.0.1: icmp_seq=5 ttl=64 time=0.072 ms
64 bytes from 10.9.0.1: icmp_seq=6 ttl=64 time=0.140 ms
64 bytes from 10.9.0.1: icmp_seq=7 ttl=64 time=0.107 ms
64 bytes from 10.9.0.1: icmp_seq=8 ttl=64 time=0.078 ms
64 bytes from 10.9.0.1: icmp_seq=9 ttl=64 time=0.163 ms
^C
--- 10.9.0.1 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8173ms
rtt min/avg/max/mdev = 0.043/0.094/0.163/0.035 ms
```



For this task we will create another new file named seedBlock as follows:



we will have two separate functions:
1. preventing other computers to ping the VM, and
2. preventing other computers to telnet into the VM.

They are implemented as follows we will have two separate functions:
1. preventing other computers to ping the VM, and
2. preventing other computers to telnet into the VM.

They are implemented as follows:

Added 2 hooks:

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

Added functions:

```c
// blocking ping to vm: 10.9.0.1
unsigned int blockICMP(void *priv, struct sk_buff *skb,
                       const struct nf_hook_state *state)
{
   struct iphdr *iph;
   struct icmphdr *icmph;

   //u16  port   = 53;
   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) {
       icmph = icmp_hdr(skb);
       if (iph->daddr == ip_addr && icmph->type == ICMP_ECHO){
            printk(KERN_WARNING "*** Dropping %pI4 (UDP)\n", &(iph->daddr));
            return NF_DROP;
        }
   }
   return NF_ACCEPT;
}
```

```c
// blocking telnet to vm: 10.9.0.1:23
unsigned int blockTelnet(void *priv, struct sk_buff *skb,
                         const struct nf_hook_state *state)
{
   struct iphdr *iph;
   struct tcphdr *tcph;

   u16  port   = 23; //telnet
   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 (UDP), port %d\n", &(iph->daddr), port);
            return NF_DROP;
        }
   }
   return NF_ACCEPT;
}
```

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

// blocking udp to 8.8.8.8:53
unsigned int blockUDP(void *priv, struct sk_buff *skb,
                      const struct nf_hook_state *state)
{
   struct iphdr *iph;
   struct udphdr *udph;

   u16  port   = 53;
   char ip[16] = "8.8.8.8";
   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_UDP) {
       udph = udp_hdr(skb);
       if (iph->daddr == ip_addr && ntohs(udph->dest) == port){
            printk(KERN_WARNING "*** Dropping %pI4 (UDP), port %d\n", &(iph->daddr), port);
            return NF_DROP;
        }
   }
   return NF_ACCEPT;
}
```

```c
   hook3.hook = blockICMP;
   hook3.hooknum = NF_INET_PRE_ROUTING;
   hook3.pf = PF_INET;
   hook3.priority = NF_IP_PRI_FIRST;
   nf_register_net_hook(&init_net, &hook3);

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

   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);
   nf_unregister_net_hook(&init_net, &hook3);
   nf_unregister_net_hook(&init_net, &hook4);
```

made module and inserted the kernel module as follows:

```
seed@VM: ~/.../packet_filter
cket_filter/seedBlock.o] Error 1
make[1]: *** [Makefile:1757: /home/seed/Documents/Labsetup/Files/packet_filter]
Error 2
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
make: *** [Makefile:5: all] Error 2
[04/03/23]seed@VM:~/.../packet_filter$ gedit seedBlock.c
[04/03/23]seed@VM:~/.../packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Documents/Labsetup/File
s/packet_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/Documents/Labsetup/Files/packet_filter/seedBlock.o
  Building modules, stage 2.
  MODPOST 1 modules
WARNING: modpost: missing MODULE_LICENSE() in /home/seed/Documents/Labsetup/File
s/packet_filter/seedBlock.o
see include/linux/module.h for more information
  CC [M]  /home/seed/Documents/Labsetup/Files/packet_filter/seedBlock.mod.o
  LD [M]  /home/seed/Documents/Labsetup/Files/packet_filter/seedBlock.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[04/03/23]seed@VM:~/.../packet_filter$ ls
Makefile        seedBlock.c    seedBlock.mod.c  seedFilter.c
modules.order   seedBlock.ko   seedBlock.mod.o  seedPrint.c
Module.symvers  seedBlock.mod  seedBlock.o
[04/03/23]seed@VM:~/.../packet_filter$
```

We can see the message being displayed using dmesg which shows that the filters are registered using the changes we made:

```
seed@VM: ~/.../Labsetup   seed@VM: ~/.../packet_fi...   seed@VM: ~/.../kernel_...   seed@V
[04/03/23]seed@VM:~/.../packet_filter$ sudo insmod seedBlock.ko
[04/03/23]seed@VM:~/.../packet_filter$
```

```
[29361.062681] seedBlock:Registering filters.
```

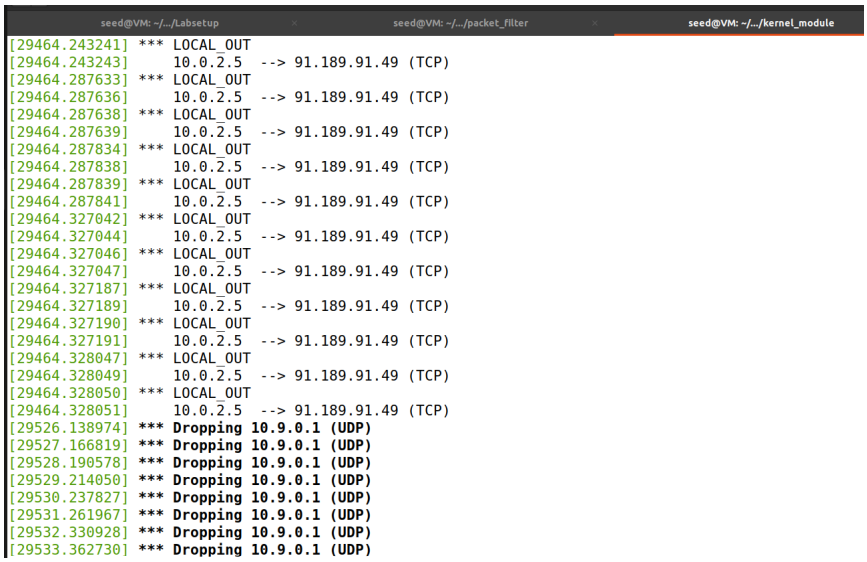
dockps:

```
seed@VM: ~/.../Labsetup   seed@VM: ~/.../pac
[04/03/23]seed@VM:~/.../Labsetup$ dockps
3546a13573ea  malicious-router-10.9.0.111
90d130a106b7  victim-10.9.0.5
afa5e22c5b4a  attacker-10.9.0.105
6778a2c78e57  router
50fd7a2061e7  host-192.168.60.6
4a543290e0f8  host-192.168.60.5
[04/03/23]seed@VM:~/.../Labsetup$
```

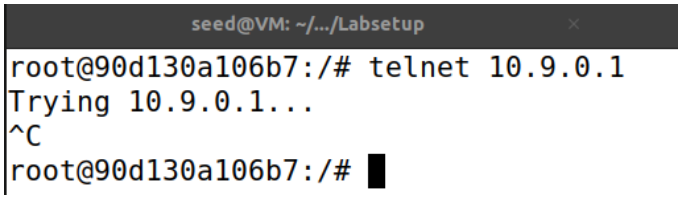
Now we try to ping into the VM and can see that the UDP packets are getting dropped:

```
seed@VM: ~/.../Labsetup   seed@VM: ~/.../packet_filter
root@90d130a106b7:/# 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 ---
25 packets transmitted, 0 received, 100% packet loss, time 24840ms

root@90d130a106b7:/#
```



We try to telnet into the VM and can see that it is not successful:



We can see the output as the UDP packets are being dropped:

```
[29885.421675] *** Dropping 10.9.0.1 (UDP), port 23
[29886.430222] *** Dropping 10.9.0.1 (UDP), port 23
[29888.459526] *** Dropping 10.9.0.1 (UDP), port 23
[29913.951361] *** LOCAL_OUT
[29913.951364]     10.0.2.5  --> 24.112.36.86 (UDP)
[29913.951365] *** LOCAL_OUT
[29913.951366]     10.0.2.5  --> 24.112.36.86 (UDP)
[29917.716095] *** LOCAL_OUT
[29917.716101]     127.0.0.1  --> 127.0.0.53 (UDP)
[29917.716104] *** LOCAL_OUT
[29917.716107]     127.0.0.1  --> 127.0.0.53 (UDP)
[29917.716466] *** LOCAL_OUT
[29917.716470]     127.0.0.53  --> 127.0.0.1 (UDP)
[29917.716471] *** LOCAL_OUT
[29917.716474]     127.0.0.53  --> 127.0.0.1 (UDP)
[29917.797991] *** LOCAL_OUT
[29917.797993]     127.0.0.1  --> 127.0.0.53 (UDP)
[29917.797994] *** LOCAL_OUT
[29917.797994]     127.0.0.1  --> 127.0.0.53 (UDP)
[29917.798165] *** LOCAL_OUT
[29917.798167]     127.0.0.53  --> 127.0.0.1 (UDP)
[29917.798168] *** LOCAL_OUT
[29917.798168]     127.0.0.53  --> 127.0.0.1 (UDP)
```