

COMPUTER NETWORK

SECURITY

LAB-7

FIREWALL

EXPLORATION LAB

NAME: VISHWAS M

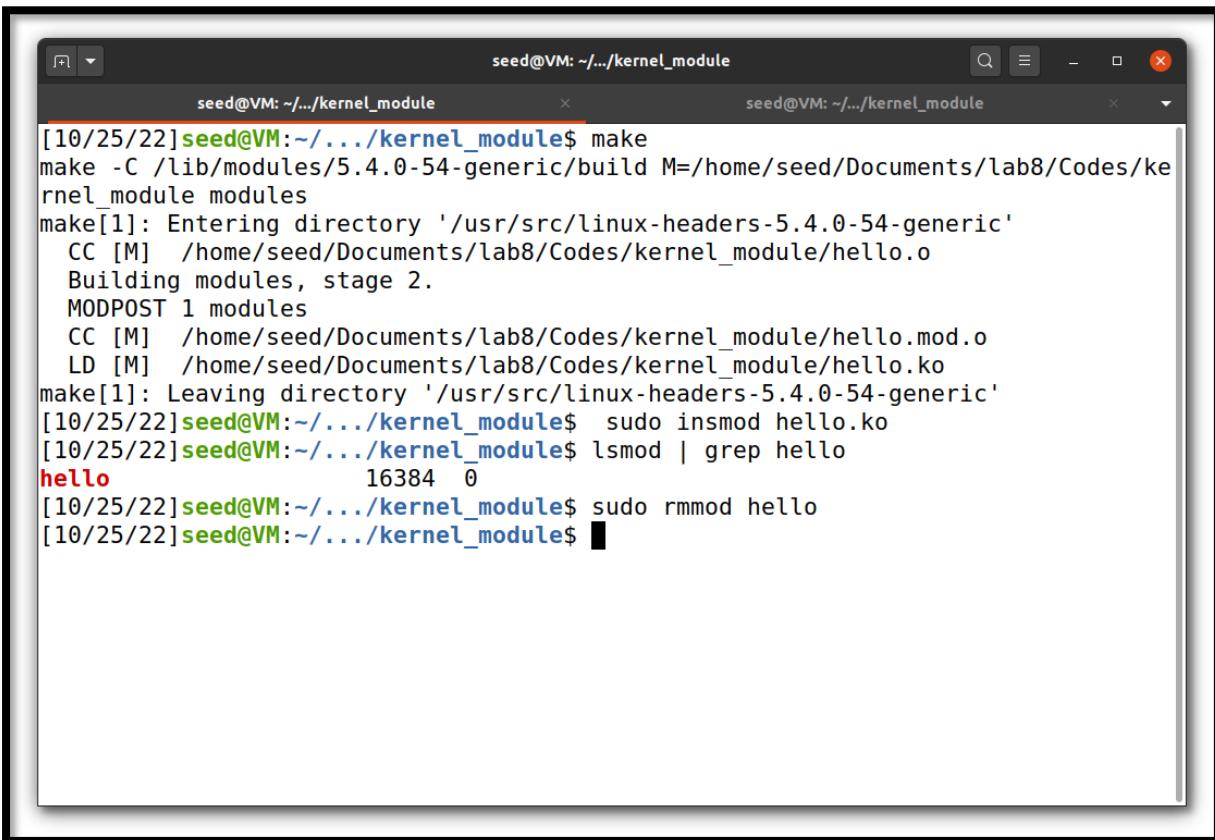
SRN: PES2UG20CS390

SEC: F

DATE:26/10/2022

Task 1:

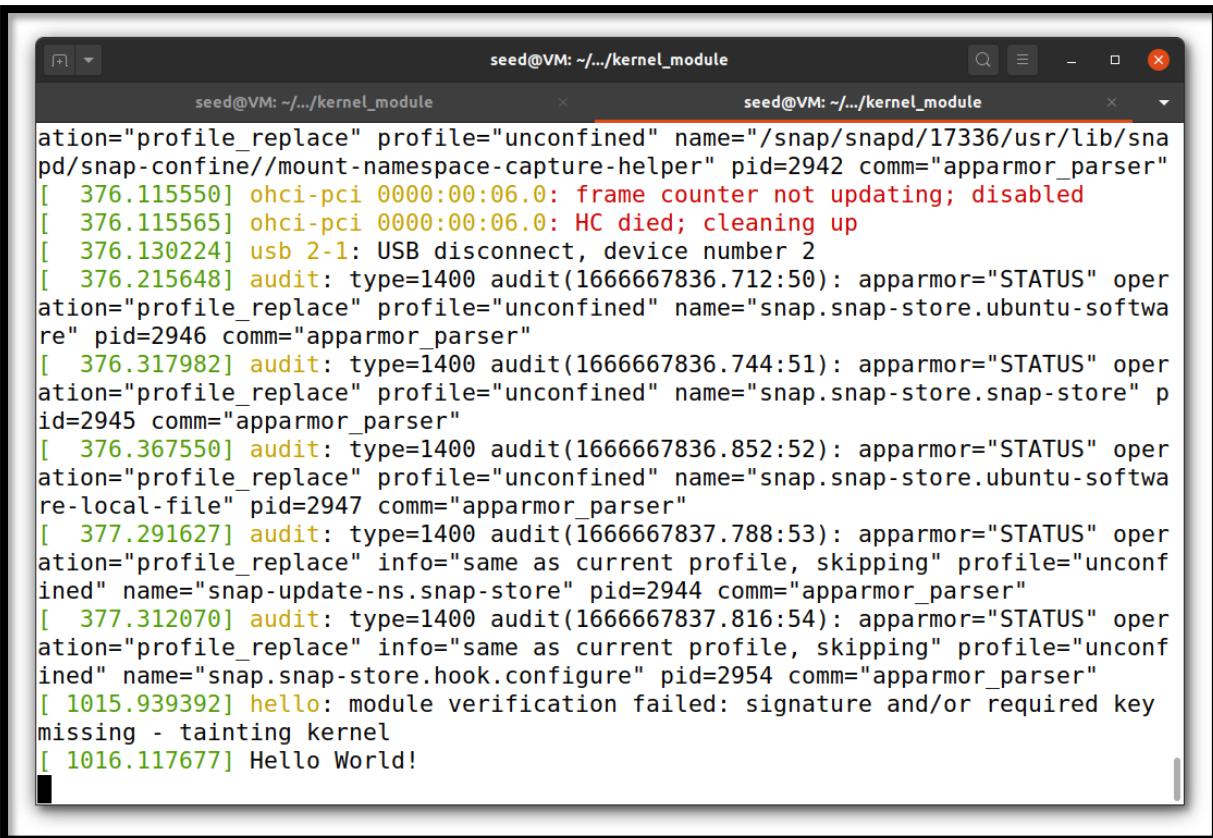
Task 1.A: Implement a simple kernel module



The screenshot shows a terminal window with two tabs, both titled "seed@VM: ~/.../kernel_module". The terminal is displaying a command-line session for building a kernel module named "hello".

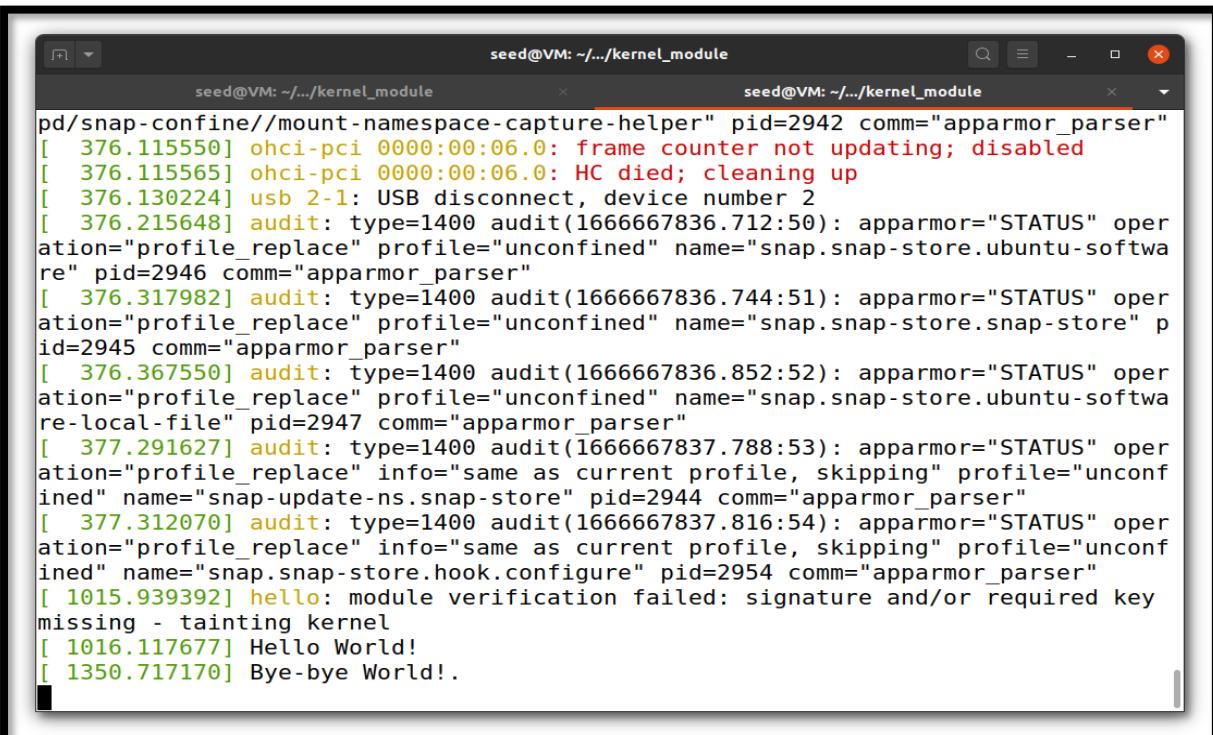
```
[10/25/22] seed@VM:~/.../kernel_module$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Documents/lab8/Codes/kernel_module modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M] /home/seed/Documents/lab8/Codes/kernel_module/hello.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M] /home/seed/Documents/lab8/Codes/kernel_module/hello.mod.o
  LD [M] /home/seed/Documents/lab8/Codes/kernel_module/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[10/25/22] seed@VM:~/.../kernel_module$ sudo insmod hello.ko
[10/25/22] seed@VM:~/.../kernel_module$ lsmod | grep hello
hello                  16384  0
[10/25/22] seed@VM:~/.../kernel_module$ sudo rmmod hello
[10/25/22] seed@VM:~/.../kernel_module$
```

In this task we are trying to make a new kernel module with the help of make command. Then we are making use of insmod command to insert the kernel module into the kernel. Then we use the grep command to view all the files which are uploaded now.



```
seed@VM: ~/kernel_module
[ 376.115550] ohci-pci 0000:00:06.0: frame counter not updating; disabled
[ 376.115565] ohci-pci 0000:00:06.0: HC died; cleaning up
[ 376.130224] usb 2-1: USB disconnect, device number 2
[ 376.215648] audit: type=1400 audit(1666667836.712:50): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="/snap/snapd/17336/usr/lib/snapd/snap-confine//mount-namespace-capture-helper" pid=2942 comm="apparmor_parser"
[ 376.317982] audit: type=1400 audit(1666667836.744:51): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="snap.snap-store.ubuntu-software" pid=2945 comm="apparmor_parser"
[ 376.367550] audit: type=1400 audit(1666667836.852:52): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="snap.snap-store.ubuntu-software-local-file" pid=2947 comm="apparmor_parser"
[ 377.291627] audit: type=1400 audit(1666667837.788:53): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap-update-ns.snap-store" pid=2944 comm="apparmor_parser"
[ 377.312070] audit: type=1400 audit(1666667837.816:54): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap.snap-store.hook.configure" pid=2954 comm="apparmor_parser"
[ 1015.939392] hello: module verification failed: signature and/or required key missing - tainting kernel
[ 1016.117677] Hello World!
```

The messages are not printed on the terminal when we run this module. It will get stored in /var/log/syslog . So we use the dmesg command to view the “Hello Word” message in the host terminal.

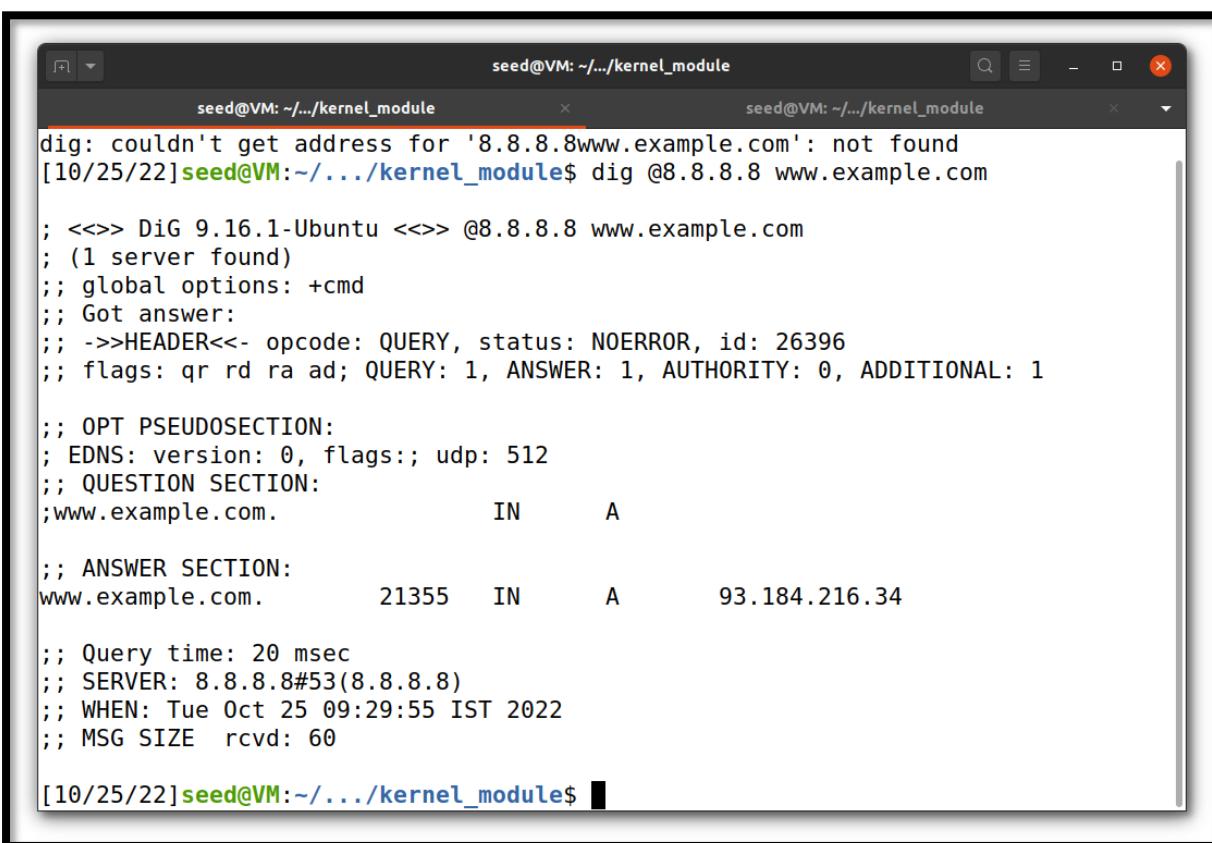


```
seed@VM: ~/kernel_module
[ 376.115550] ohci-pci 0000:00:06.0: frame counter not updating; disabled
[ 376.115565] ohci-pci 0000:00:06.0: HC died; cleaning up
[ 376.130224] usb 2-1: USB disconnect, device number 2
[ 376.215648] audit: type=1400 audit(1666667836.712:50): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="/snap/snapd/17336/usr/lib/snapd/snap-confine//mount-namespace-capture-helper" pid=2942 comm="apparmor_parser"
[ 376.317982] audit: type=1400 audit(1666667836.744:51): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="snap.snap-store.ubuntu-software" pid=2945 comm="apparmor_parser"
[ 376.367550] audit: type=1400 audit(1666667836.852:52): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="snap.snap-store.ubuntu-software-local-file" pid=2947 comm="apparmor_parser"
[ 377.291627] audit: type=1400 audit(1666667837.788:53): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap-update-ns.snap-store" pid=2944 comm="apparmor_parser"
[ 377.312070] audit: type=1400 audit(1666667837.816:54): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap.snap-store.hook.configure" pid=2954 comm="apparmor_parser"
[ 1015.939392] hello: module verification failed: signature and/or required key missing - tainting kernel
[ 1016.117677] Hello World!
[ 1350.717170] Bye-bye World!.
```

After executing the rmmod command to delete the kernel module that we just uploaded, the message “Bye-bye World” will get printed in the terminal.

Task 1.B: Implement a single firewall using Netfilter

1)



The screenshot shows a terminal window with two tabs, both titled "seed@VM: ~/.../kernel_module". The left tab contains the command "dig @8.8.8.8 www.example.com" and its output. The output shows a successful DNS query for the IP address 93.184.216.34. The right tab is empty. The terminal prompt "[10/25/22] seed@VM:~/.../kernel_module \$" is visible at the bottom.

```
seed@VM: ~/.../kernel_module
dig: couldn't get address for '8.8.8.8www.example.com': not found
[10/25/22]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: 26396
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

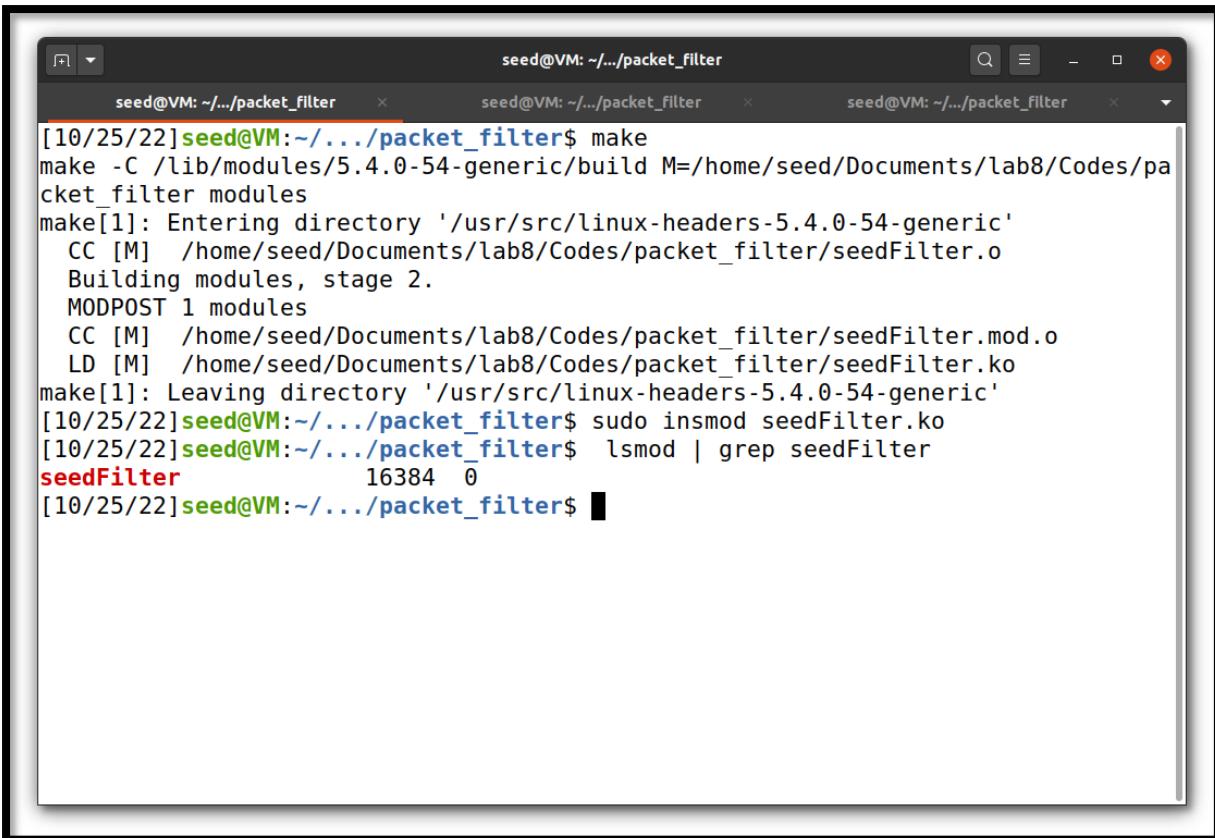
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.example.com.           IN      A

;; ANSWER SECTION:
www.example.com.        21355   IN      A      93.184.216.34

;; Query time: 20 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Oct 25 09:29:55 IST 2022
;; MSG SIZE  rcvd: 60

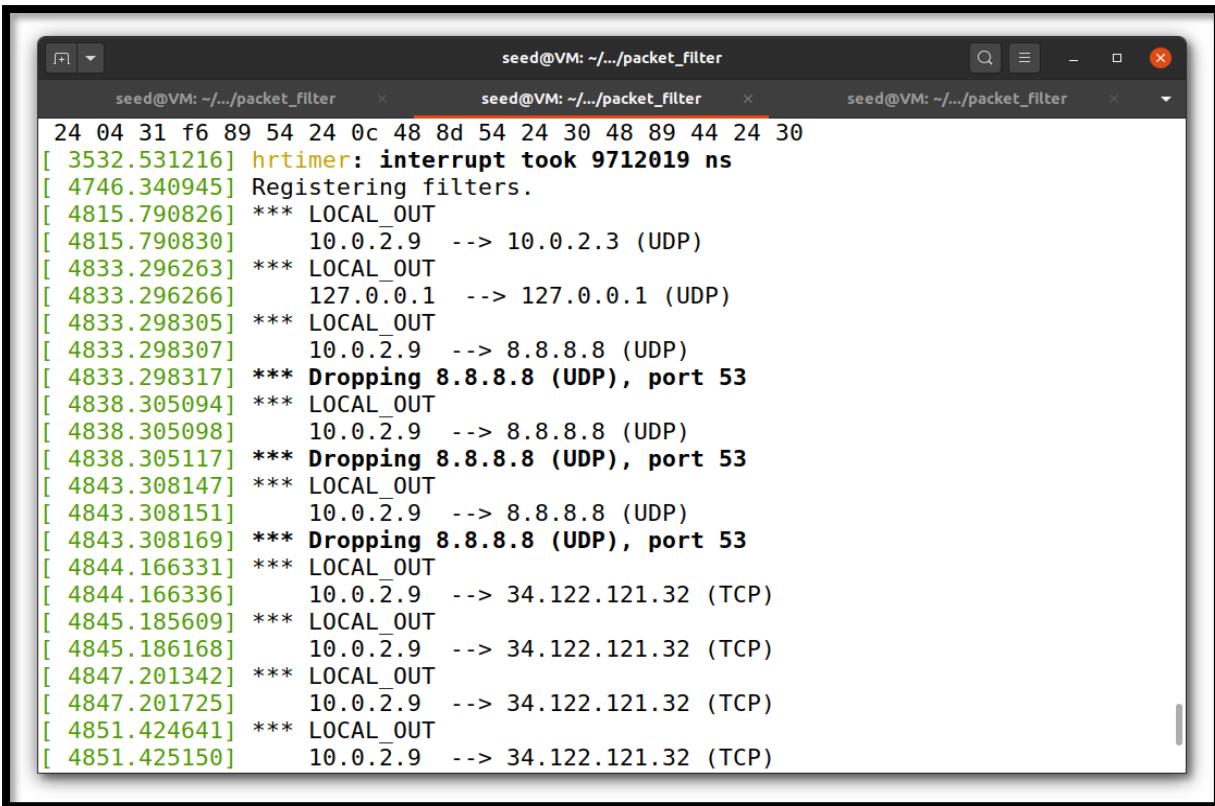
[10/25/22]seed@VM:~/.../kernel_module$
```

First we are trying to see whether the server 8.8.8.8 is reachable or not. By the above screenshot we can say that the server is reachable. And this is before manipulating the firewall.



```
seed@VM: ~/.../packet_filter
[10/25/22]seed@VM:~/.../packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Documents/lab8/Codes/pa
cket_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/Documents/lab8/Codes/packet_filter/seedFilter.o
Building modules, stage 2.
MODPOST 1 modules
  CC [M]  /home/seed/Documents/lab8/Codes/packet_filter/seedFilter.mod.o
  LD [M]  /home/seed/Documents/lab8/Codes/packet_filter/seedFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[10/25/22]seed@VM:~/.../packet_filter$ sudo insmod seedFilter.ko
[10/25/22]seed@VM:~/.../packet_filter$ lsmod | grep seedFilter
seedFilter                 16384  0
[10/25/22]seed@VM:~/.../packet_filter$
```

I am uploading a kernel module which is used to block all the requests going to 8.8.8.8 server.



seed@VM: ~/.../packet_filter

```
24 04 31 f6 89 54 24 0c 48 8d 54 24 30 48 89 44 24 30
[ 3532.531216] hrtimer: interrupt took 9712019 ns
[ 4746.340945] Registering filters.
[ 4815.790826] *** LOCAL_OUT
[ 4815.790830] 10.0.2.9 --> 10.0.2.3 (UDP)
[ 4833.296263] *** LOCAL_OUT
[ 4833.296266] 127.0.0.1 --> 127.0.0.1 (UDP)
[ 4833.298305] *** LOCAL_OUT
[ 4833.298307] 10.0.2.9 --> 8.8.8.8 (UDP)
[ 4833.298317] *** Dropping 8.8.8.8 (UDP), port 53
[ 4838.305094] *** LOCAL_OUT
[ 4838.305098] 10.0.2.9 --> 8.8.8.8 (UDP)
[ 4838.305117] *** Dropping 8.8.8.8 (UDP), port 53
[ 4843.308147] *** LOCAL_OUT
[ 4843.308151] 10.0.2.9 --> 8.8.8.8 (UDP)
[ 4843.308169] *** Dropping 8.8.8.8 (UDP), port 53
[ 4844.166331] *** LOCAL_OUT
[ 4844.166336] 10.0.2.9 --> 34.122.121.32 (TCP)
[ 4845.185609] *** LOCAL_OUT
[ 4845.186168] 10.0.2.9 --> 34.122.121.32 (TCP)
[ 4847.201342] *** LOCAL_OUT
[ 4847.201725] 10.0.2.9 --> 34.122.121.32 (TCP)
[ 4851.424641] *** LOCAL_OUT
[ 4851.425150] 10.0.2.9 --> 34.122.121.32 (TCP)
```

Here we can see that the packets are getting dropped and the requests are not reaching the server 8.8.8.8.



```
[10/25/22]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

[10/25/22]seed@VM:~/.../packet_filter$ █
```

When we execute the dig command again, we can see that the server could not be reached because the firewall has blocked all the requests going to that particular server.

2)



```
seed@VM: ~/.../packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Documents/lab8/Codes/pa
cket_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M] /home/seed/Documents/lab8/Codes/packet_filter/seedPrint.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M] /home/seed/Documents/lab8/Codes/packet_filter/seedPrint.mod.o
  LD [M] /home/seed/Documents/lab8/Codes/packet_filter/seedPrint.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[10/25/22]seed@VM:~/.../packet_filter$ sudo insmod seedPrint.ko
[10/25/22]seed@VM:~/.../packet_filter$ lsmod | grep seedPrint
seedPrint                 16384  0
[10/25/22]seed@VM:~/.../packet_filter$
```

In this subtask we are uploading a module which prints the netfilter hooks.

```
seed@VM: ~/.../packet_filter
seed@VM: ~/.../packet_filter
seed@VM: ~/.../packet_filter

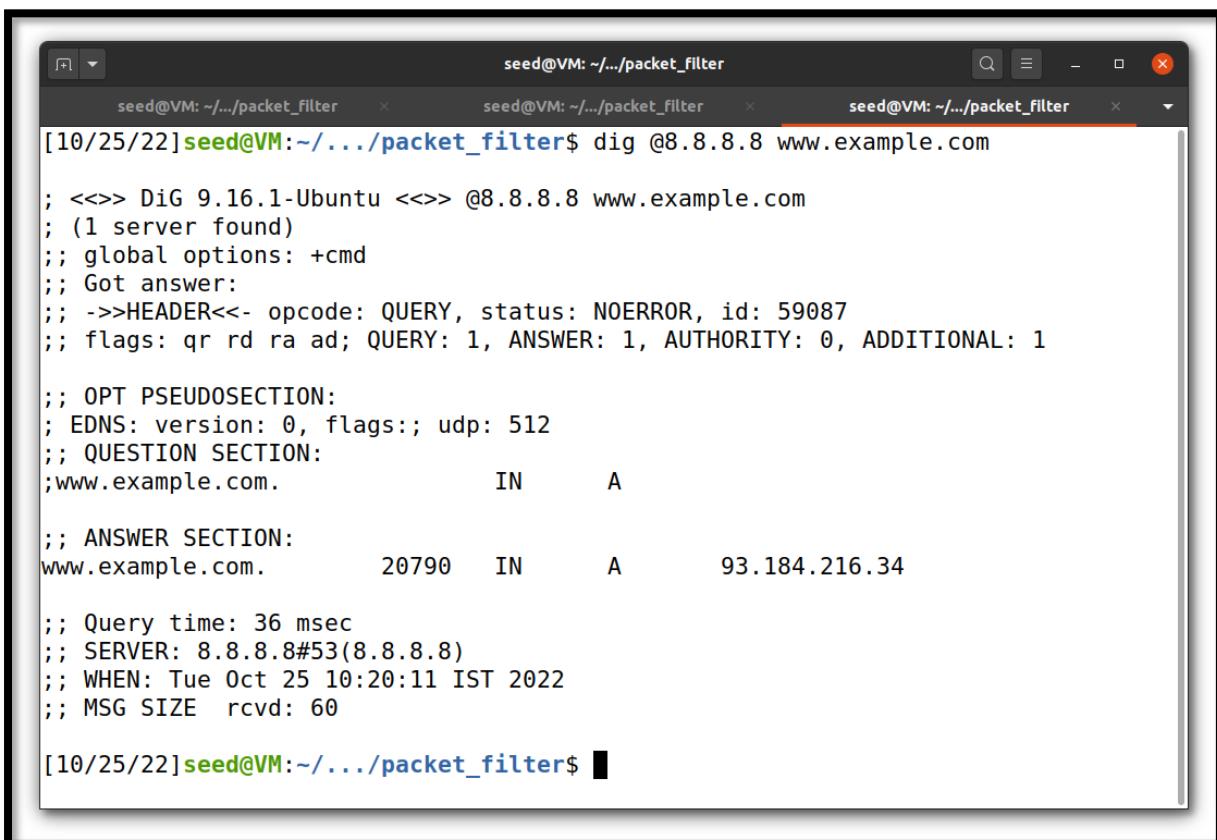
[10/25/22] seed@VM:~/.../packet_filter$ sudo dmesg -k -w
[ 5895.882157] Registering filters.
[ 5927.798423] *** LOCAL_OUT
[ 5927.798425] 10.0.2.9 --> 192.168.1.254 (UDP)
[ 5927.798442] *** POST_ROUTING
[ 5927.798443] 10.0.2.9 --> 192.168.1.254 (UDP)
[ 5927.829462] *** PRE_ROUTING
[ 5927.829765] 192.168.1.254 --> 10.0.2.9 (UDP)
[ 5927.829922] *** LOCAL_IN
[ 5927.830100] 192.168.1.254 --> 10.0.2.9 (UDP)
[ 5927.831775] *** LOCAL_OUT
[ 5927.831776] 127.0.0.1 --> 127.0.0.53 (UDP)
[ 5927.831787] *** POST_ROUTING
[ 5927.831788] 127.0.0.1 --> 127.0.0.53 (UDP)
[ 5927.831821] *** PRE_ROUTING
[ 5927.831822] 127.0.0.1 --> 127.0.0.53 (UDP)
[ 5927.831827] *** LOCAL_IN
[ 5927.831828] 127.0.0.1 --> 127.0.0.53 (UDP)
[ 5927.832132] *** LOCAL_OUT
[ 5927.832133] 10.0.2.9 --> 192.168.1.254 (UDP)
[ 5927.832138] *** POST_ROUTING
[ 5927.832139] 10.0.2.9 --> 192.168.1.254 (UDP)
[ 5927.930478] *** PRE_ROUTING
[ 5927.930760] 192.168.1.254 --> 10.0.2.9 (UDP)
```

```
seed@VM: ~/.../packet_filter
seed@VM: ~/.../packet_filter
seed@VM: ~/.../packet_filter

[ 5927.931790] *** LOCAL_OUT
[ 5927.931791] 127.0.0.53 --> 127.0.0.1 (UDP)
[ 5927.931796] *** POST_ROUTING
[ 5927.931796] 127.0.0.53 --> 127.0.0.1 (UDP)
[ 5927.931804] *** PRE_ROUTING
[ 5927.931804] 127.0.0.53 --> 127.0.0.1 (UDP)
[ 5927.931805] *** LOCAL_IN
[ 5927.931806] 127.0.0.53 --> 127.0.0.1 (UDP)
[ 5950.734833] *** LOCAL_OUT
[ 5950.734837] 127.0.0.1 --> 127.0.0.1 (UDP)
[ 5950.734856] *** POST_ROUTING
[ 5950.734857] 127.0.0.1 --> 127.0.0.1 (UDP)
[ 5950.734870] *** PRE_ROUTING
[ 5950.734870] 127.0.0.1 --> 127.0.0.1 (UDP)
[ 5950.734888] *** LOCAL_IN
[ 5950.734889] 127.0.0.1 --> 127.0.0.1 (UDP)
[ 5950.736329] *** LOCAL_OUT
[ 5950.736332] 10.0.2.9 --> 8.8.8.8 (UDP)
[ 5950.736346] *** POST_ROUTING
[ 5950.736347] 10.0.2.9 --> 8.8.8.8 (UDP)
[ 5950.769872] *** PRE_ROUTING
[ 5950.770132] 8.8.8.8 --> 10.0.2.9 (UDP)
[ 5950.770312] *** LOCAL_IN
[ 5950.770423] 8.8.8.8 --> 10.0.2.9 (UDP)
```

Here are the macros that are invoked during the experiment. The macros used are:

```
NF_INET_PRE_ROUTING  
NF_INET_LOCAL_IN  
NF_INET_FORWARD  
NF_INET_LOCAL_OUT  
NF_INET_POST_ROUTING
```

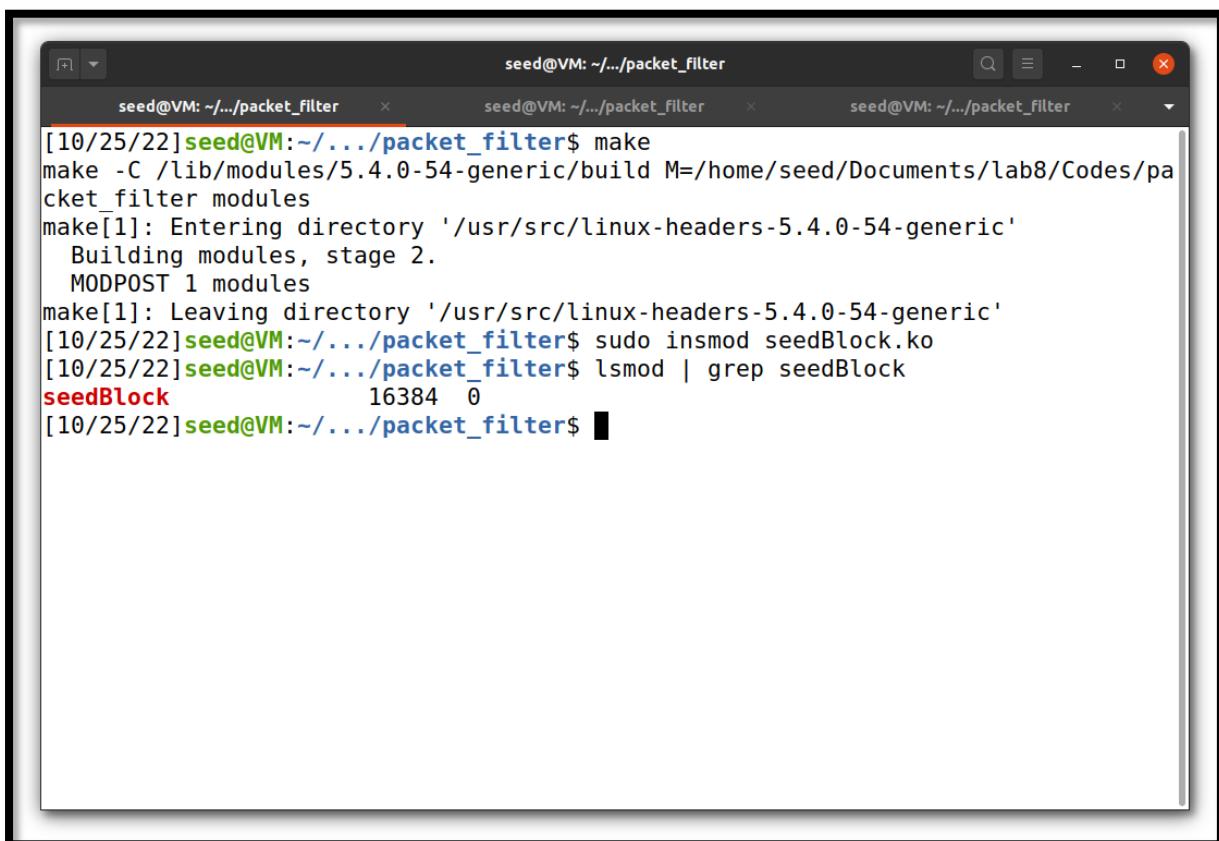


The screenshot shows a terminal window titled "seed@VM: ~.../packet_filter". It contains the output of the "dig" command. The command was run as "dig @8.8.8.8 www.example.com". The output shows a successful DNS query to an external server. The response includes the question section for "www.example.com.", the answer section with an IP address of "93.184.216.34", and various header and footer information.

```
[10/25/22]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: 59087  
;; 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. 20790 IN A 93.184.216.34  
  
;; Query time: 36 msec  
;; SERVER: 8.8.8.8#53(8.8.8.8)  
;; WHEN: Tue Oct 25 10:20:11 IST 2022  
;; MSG SIZE rcvd: 60  
  
[10/25/22]seed@VM:~.../packet_filter$ █
```

We can reach the server 8.8.8.8 when we use the dig command and simultaneously observe how the hook functions work as shown in the previous screenshots.

3)



The screenshot shows a terminal window with three tabs, all titled "seed@VM: ~.../packet_filter". The terminal displays the following command-line session:

```
[10/25/22]seed@VM:~/.../packet_filter$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Documents/lab8/Codes/packet_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  Building modules, stage 2.
    MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[10/25/22]seed@VM:~/.../packet_filter$ sudo insmod seedBlock.ko
[10/25/22]seed@VM:~/.../packet_filter$ lsmod | grep seedBlock
seedBlock               16384  0
[10/25/22]seed@VM:~/.../packet_filter$
```

In this task we are trying to add 2 more hooks to achieve:

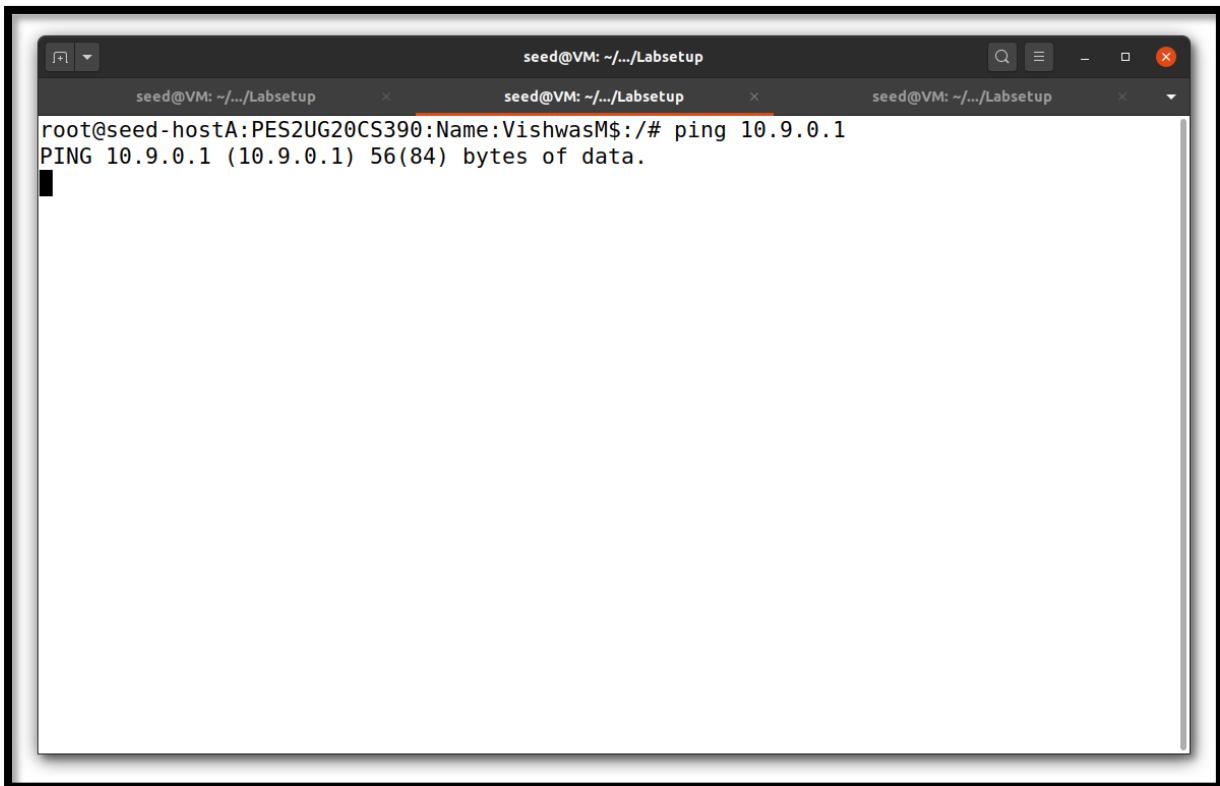
- a) Prevent other computers to ping the VM
- b) Prevent other computers from telnetting into the VM

We are adding a kernel module to perform the above two tasks. We are adding the above module by using the insmod command.

```
seed@VM: ~/.../packet_filter$ sudo dmesg -k -w
[ 8108.751331] Registering filters.
[ 8114.274894] *** LOCAL_OUT
[ 8114.274897] 10.0.2.9 --> 10.0.2.3 (UDP)
[ 8153.000466] *** Dropping 10.9.0.1 (ICMP)
[ 8154.031980] *** Dropping 10.9.0.1 (ICMP)
[ 8155.094676] *** Dropping 10.9.0.1 (ICMP)
[ 8156.112413] *** Dropping 10.9.0.1 (ICMP)
[ 8157.135906] *** Dropping 10.9.0.1 (ICMP)
[ 8158.168483] *** Dropping 10.9.0.1 (ICMP)
[ 8159.182932] *** Dropping 10.9.0.1 (ICMP)
[ 8160.207457] *** Dropping 10.9.0.1 (ICMP)
[ 8161.231422] *** Dropping 10.9.0.1 (ICMP)
[ 8162.256188] *** Dropping 10.9.0.1 (ICMP)
[ 8163.305147] *** Dropping 10.9.0.1 (ICMP)
[ 8164.335300] *** Dropping 10.9.0.1 (ICMP)
[ 8165.364660] *** Dropping 10.9.0.1 (ICMP)
[ 8166.385989] *** Dropping 10.9.0.1 (ICMP)
[ 8167.410285] *** Dropping 10.9.0.1 (ICMP)
[ 8168.441950] *** Dropping 10.9.0.1 (ICMP)
[ 8169.454867] *** Dropping 10.9.0.1 (ICMP)
[ 8170.481339] *** Dropping 10.9.0.1 (ICMP)
[ 8171.510844] *** Dropping 10.9.0.1 (ICMP)
[ 8172.526849] *** Dropping 10.9.0.1 (ICMP)
```

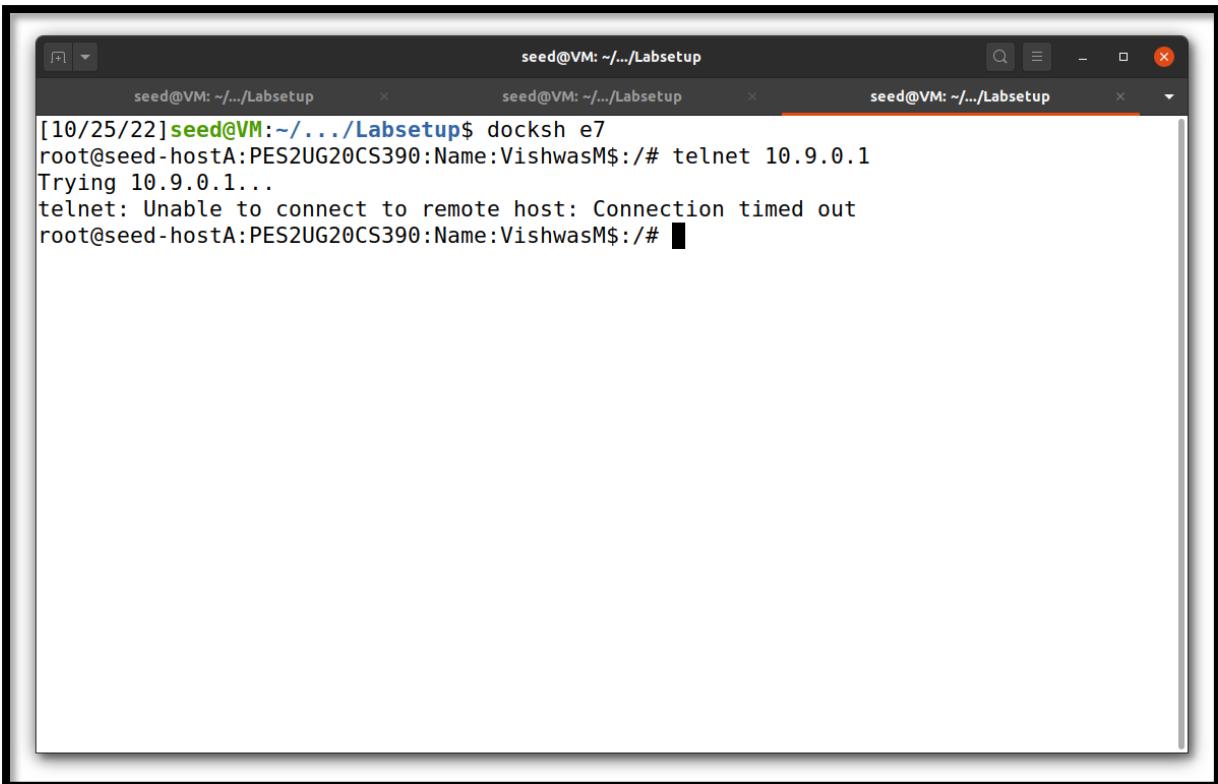
```
seed@VM: ~/.../packet_filter$ sudo dmesg -k -w
[ 8214.512593] *** Dropping 10.9.0.1 (ICMP)
[ 8215.325625] *** LOCAL_OUT
[ 8215.325878] 10.0.2.9 --> 34.122.121.32 (TCP)
[ 8215.326210] *** LOCAL_OUT
[ 8215.326212] 10.0.2.9 --> 34.122.121.32 (TCP)
[ 8215.538217] *** Dropping 10.9.0.1 (ICMP)
[ 8216.560410] *** Dropping 10.9.0.1 (ICMP)
[ 8217.583789] *** Dropping 10.9.0.1 (ICMP)
[ 8218.609743] *** Dropping 10.9.0.1 (ICMP)
[ 8219.632752] *** Dropping 10.9.0.1 (ICMP)
[ 8220.661133] *** Dropping 10.9.0.1 (ICMP)
[ 8221.681437] *** Dropping 10.9.0.1 (ICMP)
[ 8222.704011] *** Dropping 10.9.0.1 (ICMP)
[ 8223.728678] *** Dropping 10.9.0.1 (ICMP)
[ 8224.753417] *** Dropping 10.9.0.1 (ICMP)
[ 8225.777034] *** Dropping 10.9.0.1 (ICMP)
[ 8226.801761] *** Dropping 10.9.0.1 (ICMP)
[ 8227.824740] *** Dropping 10.9.0.1 (ICMP)
[ 8228.847866] *** Dropping 10.9.0.1 (ICMP)
[ 8229.871902] *** Dropping 10.9.0.1 (ICMP)
[ 8230.896705] *** Dropping 10.9.0.1 (ICMP)
[ 8231.920603] *** Dropping 10.9.0.1 (ICMP)
[ 8232.947240] *** Dropping 10.9.0.1 (ICMP)
[ 8233.116981] *** Dropping 10.9.0.1 (TCP), port 23
[ 8233.969055] *** Dropping 10.9.0.1 (ICMP)
[ 8234.127842] *** Dropping 10.9.0.1 (TCP), port 23
[ 8234.992810] *** Dropping 10.9.0.1 (ICMP)
[ 8236.018852] *** Dropping 10.9.0.1 (ICMP)
```

We can see that the ICMP packets as well as the TCP packets are getting dropped at the firewall because of the hooks that we had defined earlier. As the packets are getting dropped the host VM(10.9.0.1) is unable to reach from the container host(10.9.0.5). VM(10.9.0.1) is unable to reach from the container host(10.9.0.5).



A screenshot of a terminal window titled "seed@VM: ~.../Labsetup". The window contains three tabs, all showing the same command: "root@seed-hostA:PES2UG20CS390:Name:VishwasM\$:/# ping 10.9.0.1". The output shows "PING 10.9.0.1 (10.9.0.1) 56(84) bytes of data." followed by a blank line.

Here we can see that the ping command is not working as the packets are getting dropped at the firewall.



The screenshot shows a terminal window with three tabs open, all titled "seed@VM: ~/.../Labsetup". The active tab displays the following command and its output:

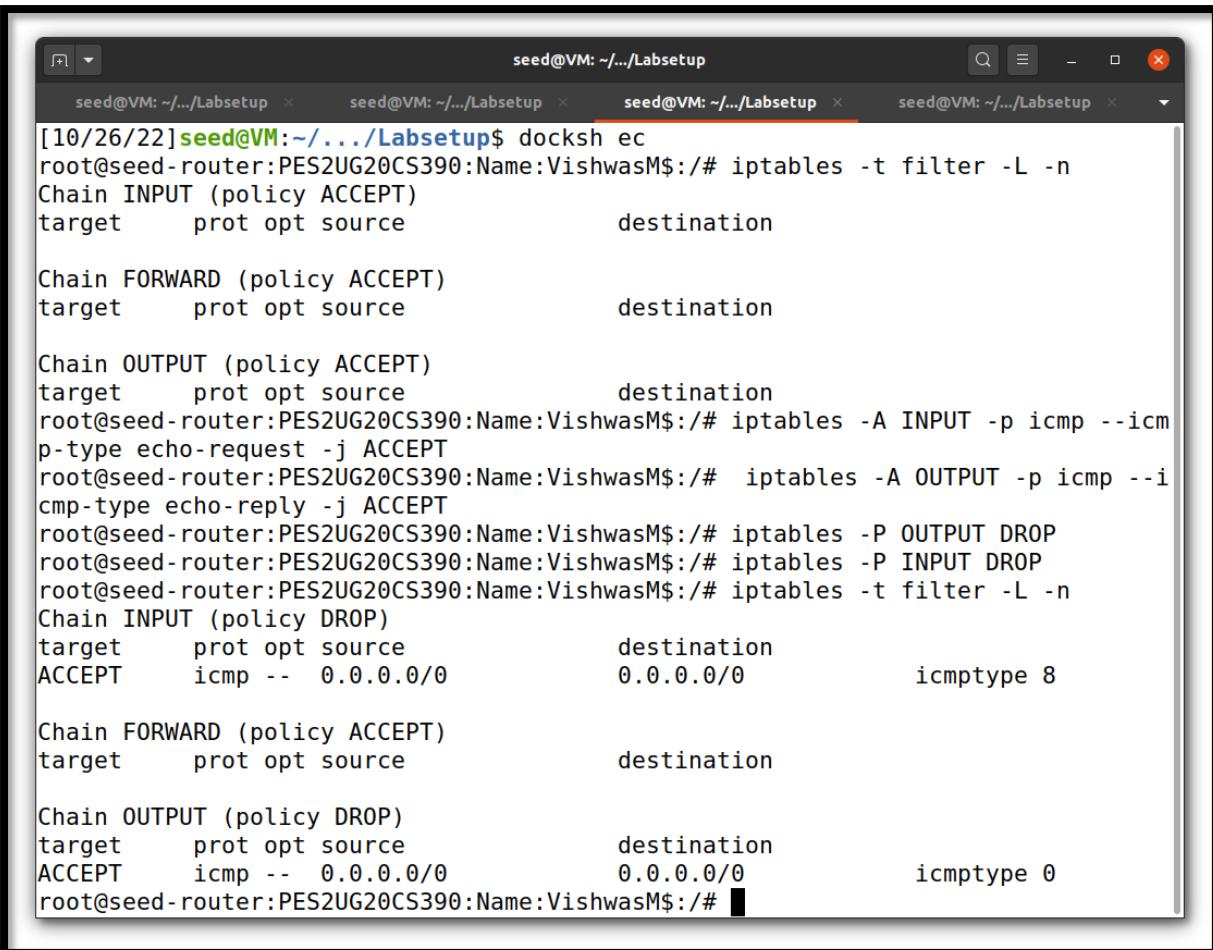
```
[10/25/22] seed@VM:~/.../Labsetup$ docksh e7
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# telnet 10.9.0.1
Trying 10.9.0.1...
telnet: Unable to connect to remote host: Connection timed out
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:#
```

Here we can see that telnet is also not working as the TCP packets are also getting dropped at the firewall.

Task 2: Experimenting with Stateless Firewall Rules

Task 2.A: Protecting the router

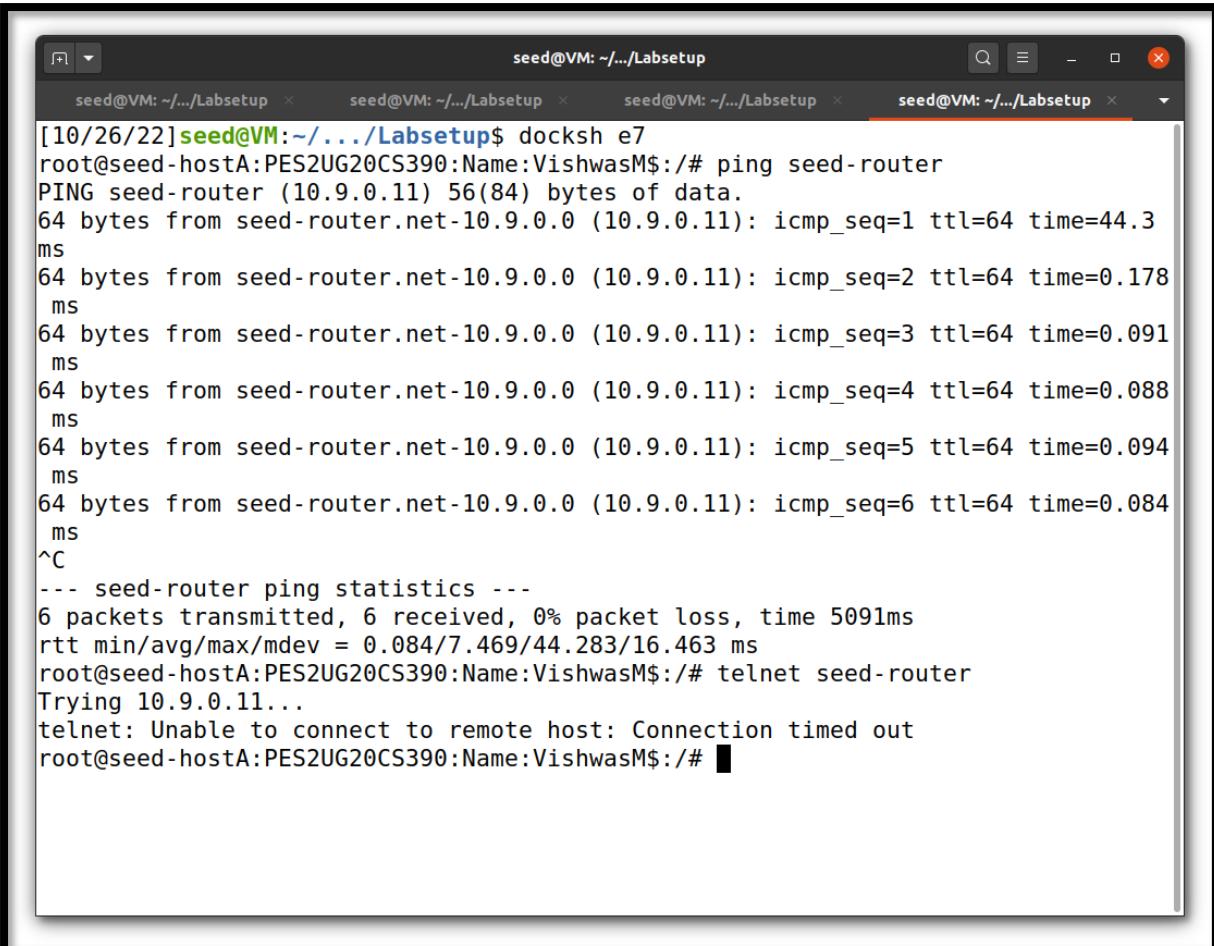
In this task, we will set some rules to prevent outside machines from accessing the router machines, except ping.



The screenshot shows a terminal window with four tabs, all titled 'seed@VM: ~.../Labsetup'. The active tab displays the following command and its output:

```
[10/26/22] seed@VM:~/.../Labsetup$ docksh ec
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -t filter -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -P OUTPUT DROP
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -P INPUT DROP
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -t filter -L -n
Chain INPUT (policy DROP)
target     prot opt source          destination
ACCEPT    icmp -- 0.0.0.0/0           0.0.0.0/0           icmptype 8
Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
Chain OUTPUT (policy DROP)
target     prot opt source          destination
ACCEPT    icmp -- 0.0.0.0/0           0.0.0.0/0           icmptype 0
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# █
```

As we can see in the above screenshot, we have applied some rules to the firewall. Here we are only accepting the ping requests and ping replies in the firewall and dropping all other packets.



The screenshot shows a terminal window with four tabs, all titled "seed@VM: ~.../Labsetup". The active tab displays the following command-line session:

```
[10/26/22] seed@VM:~/.../Labsetup$ docksh e7
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:/# ping seed-router
PING seed-router (10.9.0.11) 56(84) bytes of data.
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=1 ttl=64 time=44.3 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=2 ttl=64 time=0.178 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=3 ttl=64 time=0.091 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=4 ttl=64 time=0.088 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=5 ttl=64 time=0.094 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=6 ttl=64 time=0.084 ms
^C
--- seed-router ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5091ms
rtt min/avg/max/mdev = 0.084/7.469/44.283/16.463 ms
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:/# telnet seed-router
Trying 10.9.0.11...
telnet: Unable to connect to remote host: Connection timed out
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:/# █
```

Here we can see that when we ping the router from the host machine, the packets are getting routed properly. But when we do telnet, the packets won't get routed as the firewall blocks and drops the packets. Therefore connection will not take place between router and host.

1) Can you ping the router?

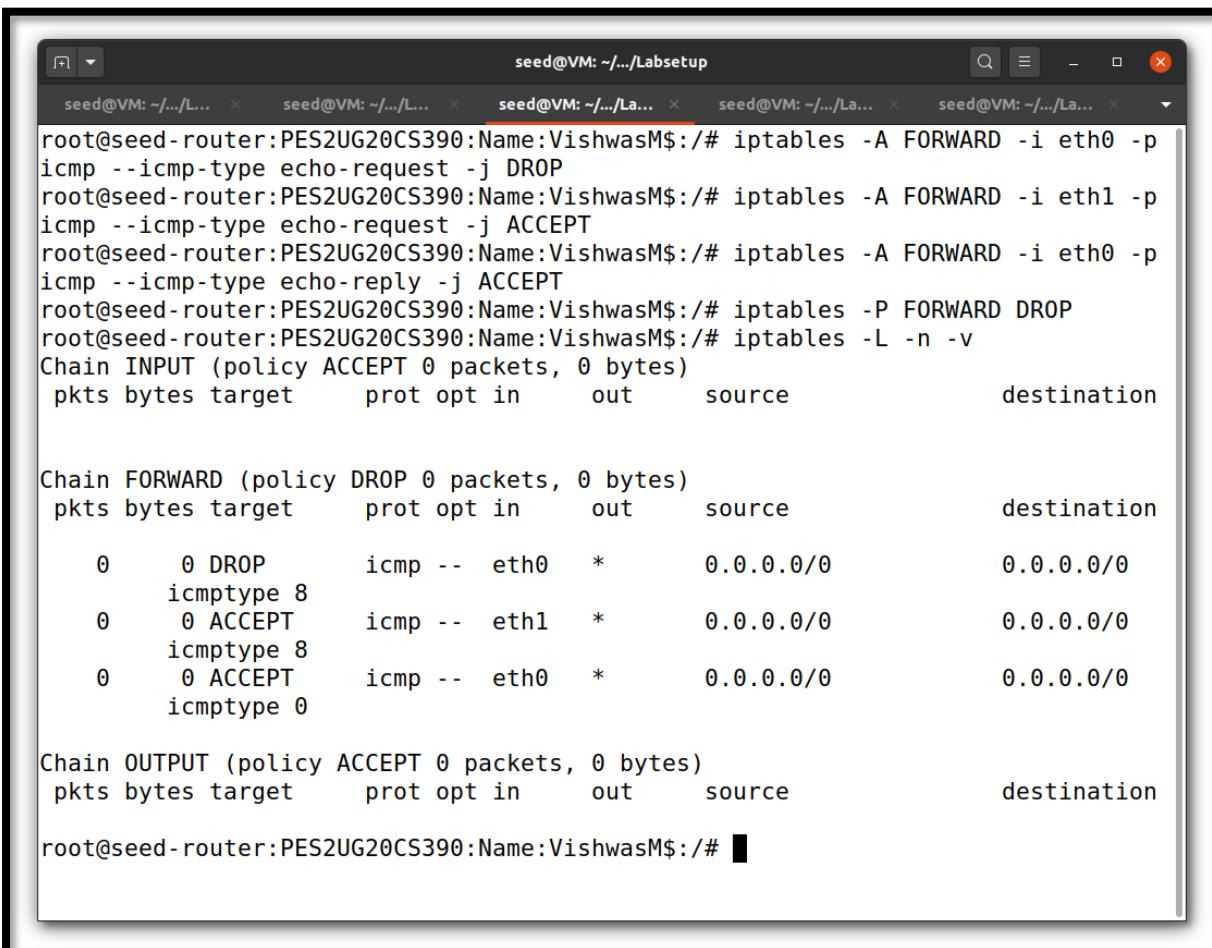
Ans: Yes, we can ping the router because of the rule that we applied to the router that only ping should be accepted.

2) Can you telnet into the network?

Ans: No, we can't telnet into the network because we have applied the rule that except ping no other packets are sent or accepted by the firewall.

Task 2.B: Protecting the Internal Network

In this task we are going to set up firewall rules on the router to protect the internal network 192.168.60.0/24. We need to use the forward chain for this.

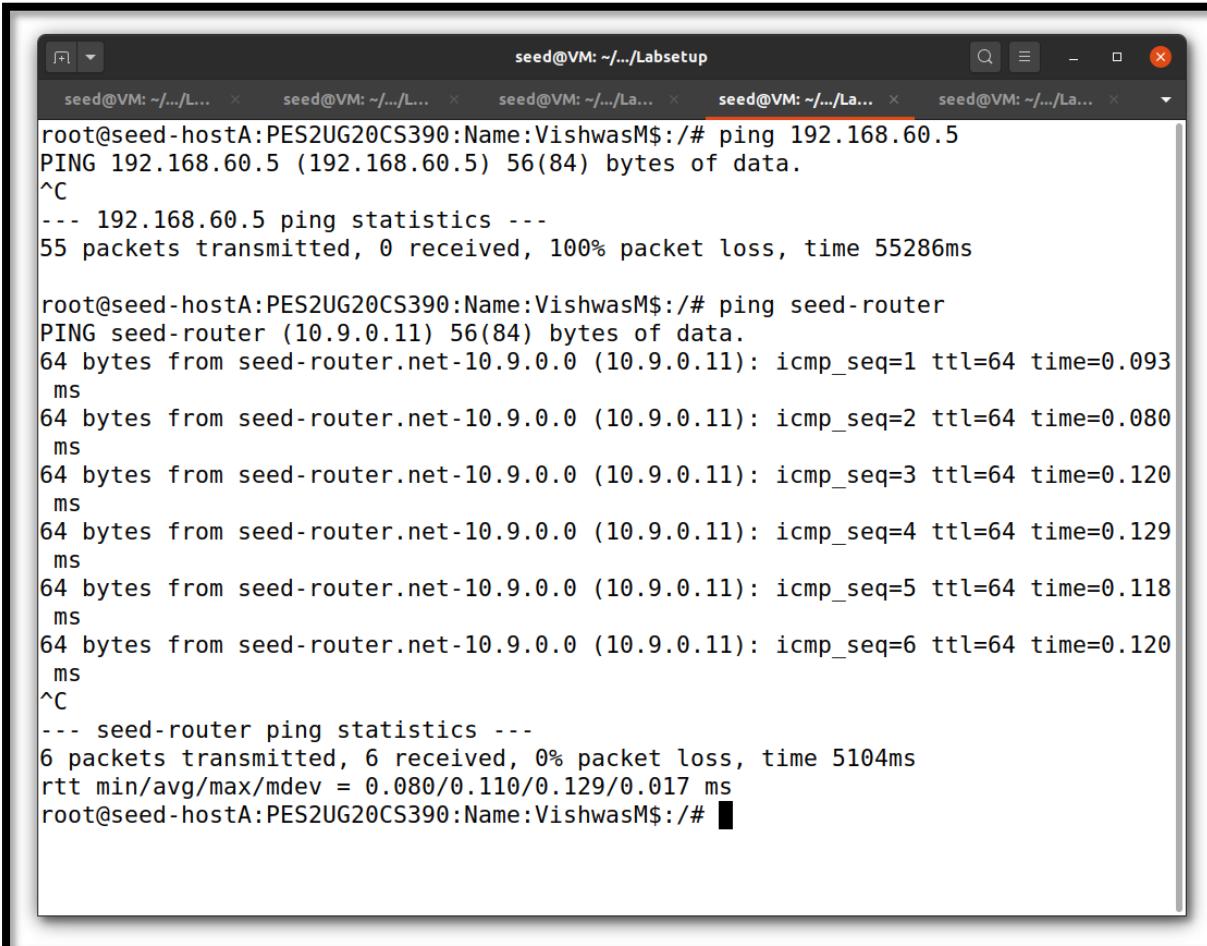


The screenshot shows a terminal window titled "seed@VM: ~/.../Labsetup". The terminal displays the following command output:

```
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-request -j DROP
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -A FORWARD -i eth1 -p icmp --icmp-type echo-request -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-reply -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -P FORWARD DROP
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in      out      source          destination
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in      out      source          destination
      0     0  DROP      icmp   --  eth0    *      0.0.0.0/0        0.0.0.0/0
      0     0  ACCEPT    icmp   --  eth1    *      0.0.0.0/0        0.0.0.0/0
      0     0  ACCEPT    icmp   --  eth0    *      0.0.0.0/0        0.0.0.0/0
      0     0  ACCEPT    icmp   --  eth0    *      0.0.0.0/0        0.0.0.0/0
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in      out      source          destination
root@seed-router:PES2UG20CS390:Name:VishwasM$:#
```

We are applying all the rules to the firewall in order to protect the internal network.

The restrictions applied are:



A screenshot of a terminal window titled "seed@VM: ~.../Labsetup". The terminal shows two ping commands. The first command, "ping 192.168.60.5", fails with a message indicating 100% packet loss. The second command, "ping seed-router", succeeds, displaying ping statistics for six packets sent to the router at 10.9.0.11.

```
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# 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 ---
55 packets transmitted, 0 received, 100% packet loss, time 55286ms

root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# ping seed-router
PING seed-router (10.9.0.11) 56(84) bytes of data.
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=1 ttl=64 time=0.093
ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=2 ttl=64 time=0.080
ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=3 ttl=64 time=0.120
ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=4 ttl=64 time=0.129
ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=5 ttl=64 time=0.118
ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=6 ttl=64 time=0.120
ms
^C
--- seed-router ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5104ms
rtt min/avg/max/mdev = 0.080/0.110/0.129/0.017 ms
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:#
```

In host A:

1. Outside hosts cannot ping internal hosts:

In the above screenshot we can observe that the ping command is not working when pinged to internal network from outside network.

2. Outside hosts can ping the router:

In the above screenshot we can observe that the ping command is working when pinged to router from outside network.

The screenshot shows a terminal window with multiple tabs, all titled 'seed@VM: ~.../Labsetup'. The active tab displays the following command-line session:

```
[10/26/22] seed@VM:~/.../Labsetup$ docksh 05
root@seed-host1:PES2UG20CS390:Name:VishwasM$:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=63 time=0.150 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.111 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.170 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.148 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=63 time=0.151 ms
64 bytes from 10.9.0.5: icmp_seq=6 ttl=63 time=0.118 ms
64 bytes from 10.9.0.5: icmp_seq=7 ttl=63 time=0.124 ms
64 bytes from 10.9.0.5: icmp_seq=8 ttl=63 time=0.111 ms
^C
--- 10.9.0.5 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7165ms
rtt min/avg/max/mdev = 0.111/0.135/0.170/0.020 ms
root@seed-host1:PES2UG20CS390:Name:VishwasM$:/# telnet 10.9.0.5
Trying 10.9.0.5...
^C
root@seed-host1:PES2UG20CS390:Name:VishwasM$:/#
```

In host 1:

3. Internal hosts can ping outside hosts:

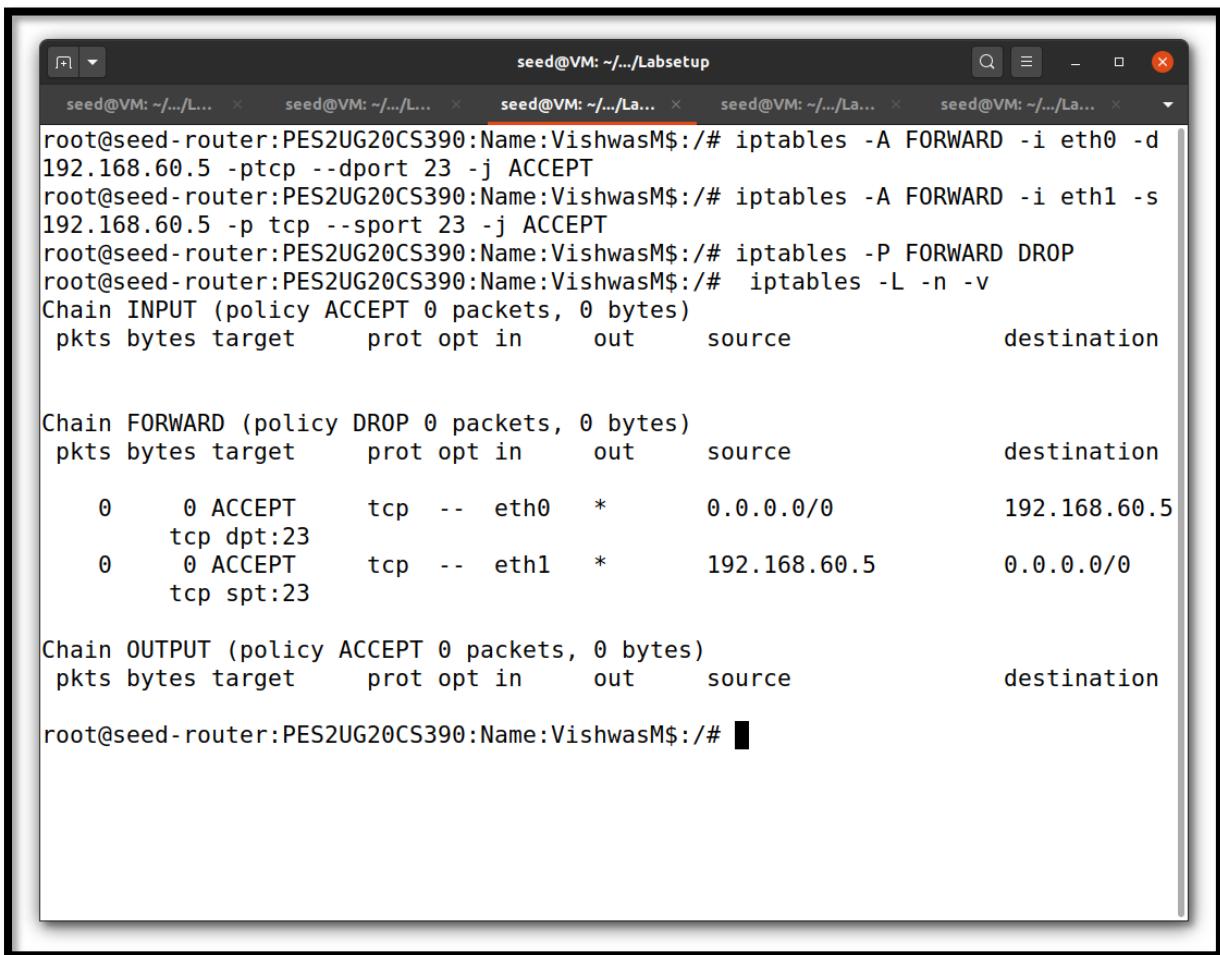
In the above screenshot we can observe that the ping command is working when pinged to external network from internal network.

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

In the above screenshot we can observe that the telnet command is not working from internal network to outside network.

Task 2.C: Protecting Internal Servers

In this task we are going to protect the TCP servers inside the internal network.



The screenshot shows a terminal window titled "seed@VM: ~.../Labsetup". The user is root on a router named "seed" with IP 192.168.60.5. They have run several commands to update the iptables rules:

```
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -A FORWARD -i eth0 -d 192.168.60.5 -ptcp --dport 23 -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -A FORWARD -i eth1 -s 192.168.60.5 -p tcp --sport 23 -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -P FORWARD DROP
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -L -n -v

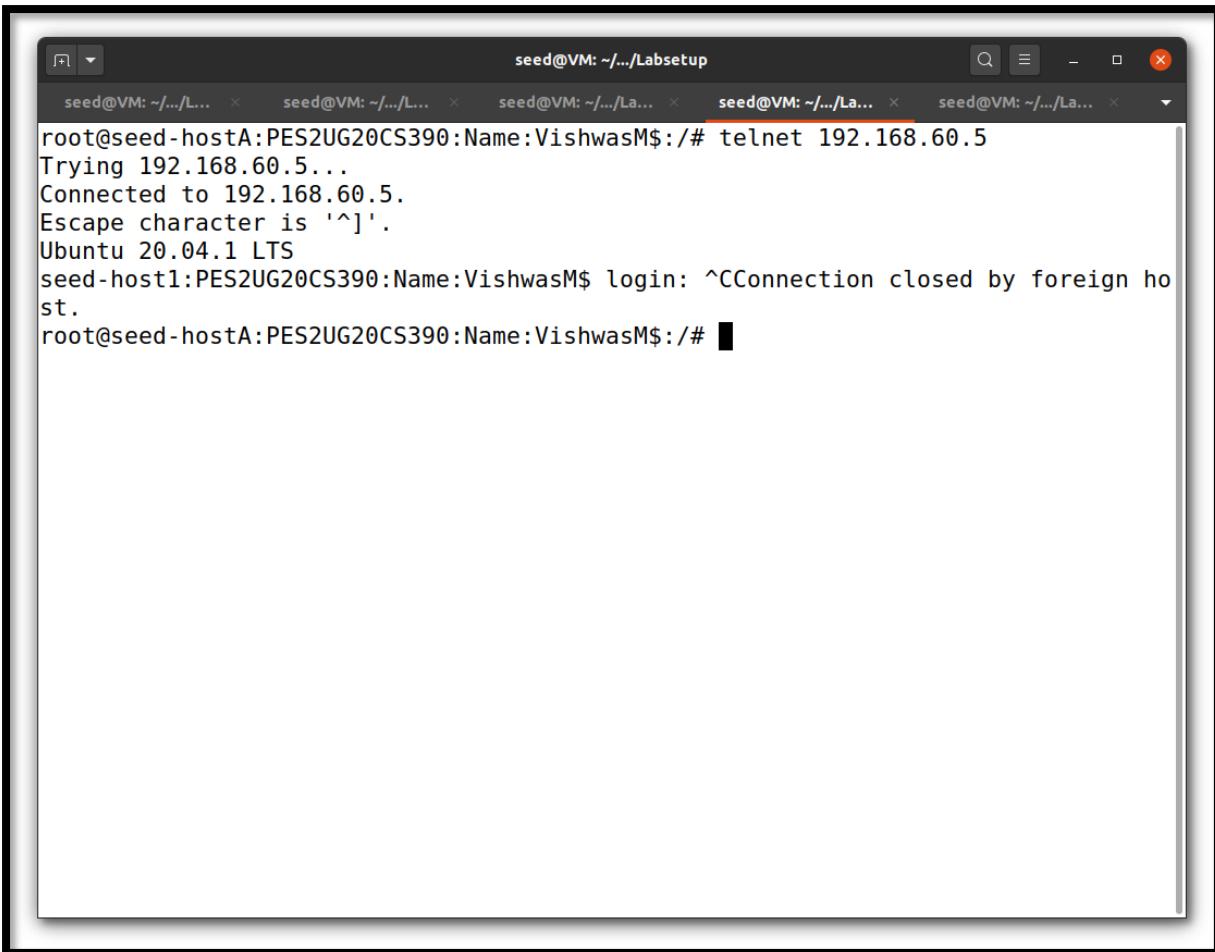
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
         

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
      0      0 ACCEPT     tcp   --  eth0    *      0.0.0.0/0            192.168.60.5
      0      0 ACCEPT     tcp   --  eth1    *      192.168.60.5        0.0.0.0/0
         

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@seed-router:PES2UG20CS390:Name:VishwasM$:#
```

We have updated the rules in the firewall.

The restrictions applied are:

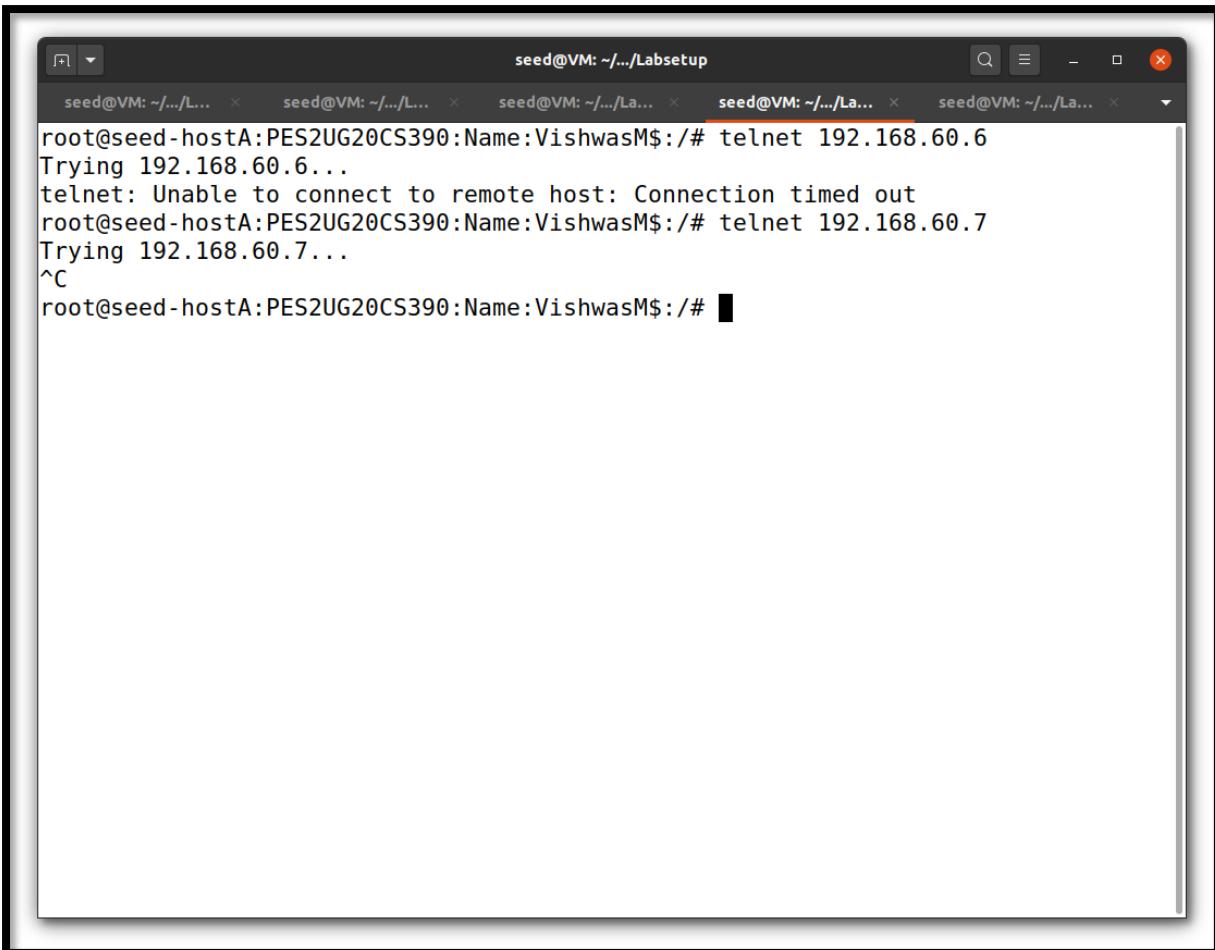


The screenshot shows a terminal window titled "seed@VM: ~/.../Labsetup". It contains the following text:

```
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
seed-host1:PES2UG20CS390:Name:VishwasM$ login: ^CConnection closed by foreign host.
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:/#
```

1. Outside hosts can only access the telnet server on 192.168.60.5:

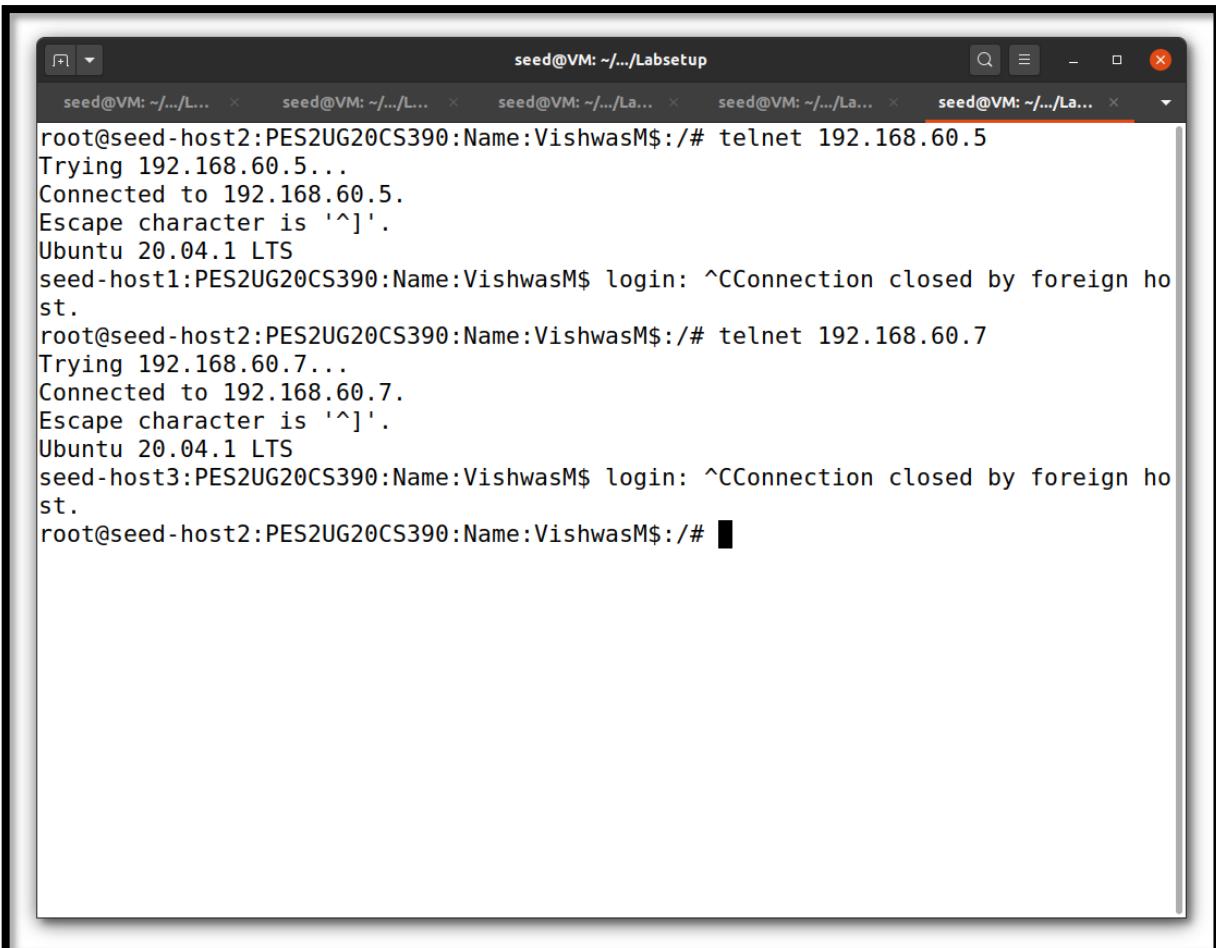
In the above screenshot we can only access telnet server on 192.168.60.5. When accessed on other IP addresses telnet will not work.



```
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# telnet 192.168.60.6
Trying 192.168.60.6...
telnet: Unable to connect to remote host: Connection timed out
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# telnet 192.168.60.7
Trying 192.168.60.7...
^C
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:#
```

2. Outside hosts cannot access the internal servers:

As we can see in the above screenshot, telnet is not working because of the rule that we applied.

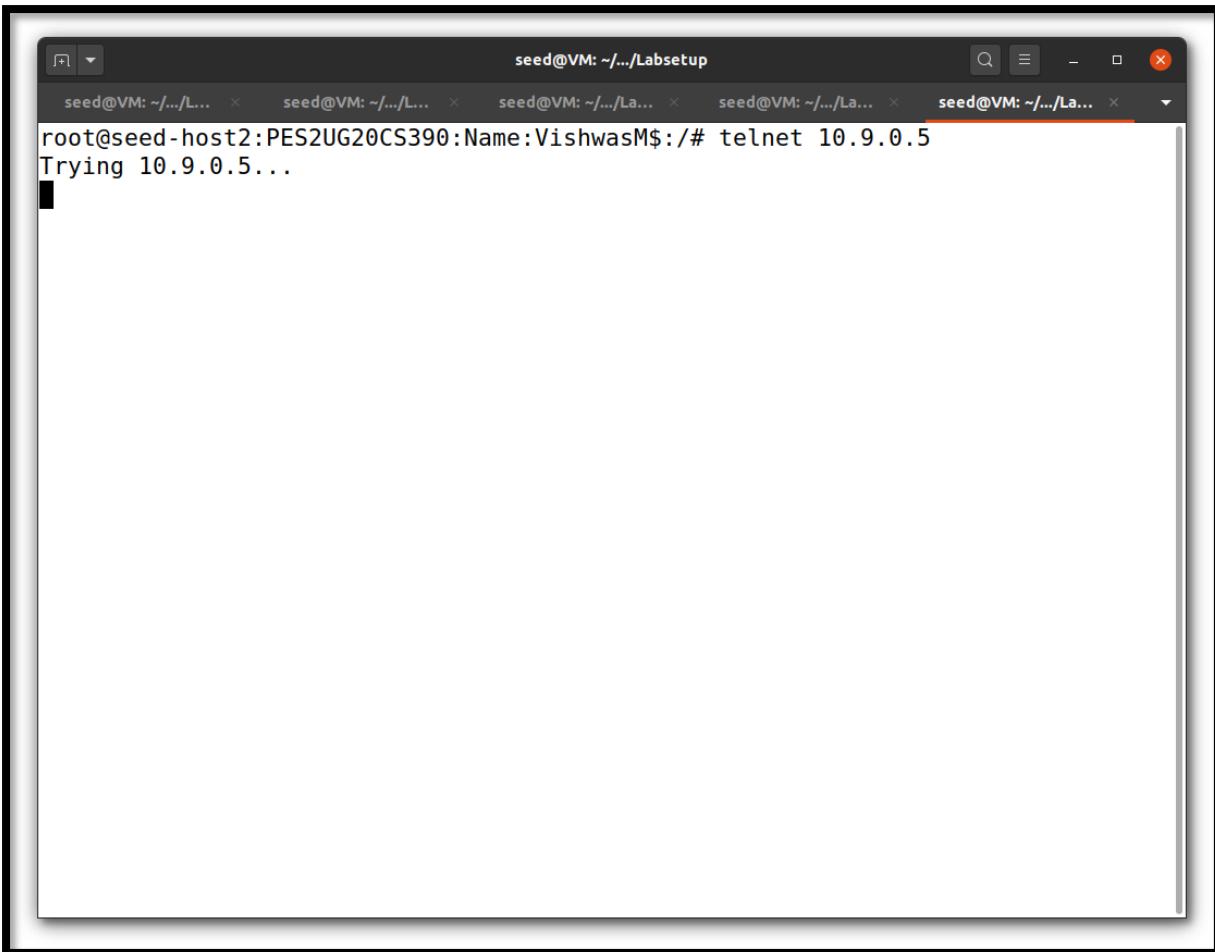


The screenshot shows a terminal window titled "seed@VM: ~/.../Labsetup". It contains five tabs, all showing the same terminal session. The session shows a user attempting to connect via telnet to an internal server at 192.168.60.5 and 192.168.60.7. Both attempts result in a connection being closed by the foreign host.

```
root@seed-host2:PES2UG20CS390:Name:VishwasM$:# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
seed-host1:PES2UG20CS390:Name:VishwasM$ login: ^CConnection closed by foreign host.
root@seed-host2:PES2UG20CS390:Name:VishwasM$:# telnet 192.168.60.7
Trying 192.168.60.7...
Connected to 192.168.60.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
seed-host3:PES2UG20CS390:Name:VishwasM$ login: ^CConnection closed by foreign host.
root@seed-host2:PES2UG20CS390:Name:VishwasM$:#
```

3. Internal hosts can access the internal servers:

In the above ss, we can see that the telnet works from internal hosts to internal server.



The screenshot shows a terminal window titled "seed@VM: ~/.../Labsetup". The command entered is "telnet 10.9.0.5", followed by "Trying 10.9.0.5...". The terminal is otherwise empty, indicating no further output or connection was established.

4. Internal servers cannot access the external hosts:

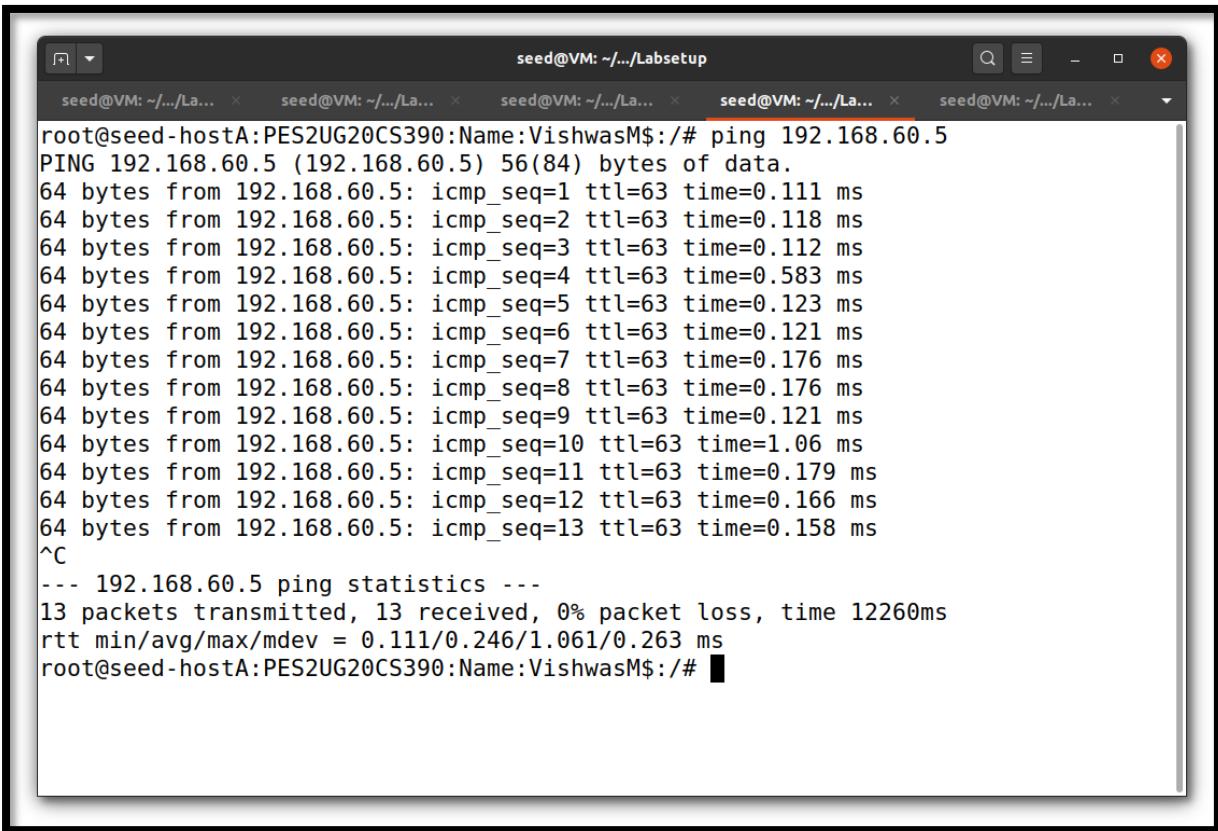
In the above we are trying to connect with external host(10.9.0.5) from the internal host(host 2) which is failing to connect.

Task 3: Connection Tracking and Stateful Firewall

Task 3.A: Experimenting with the Connecting Tracking

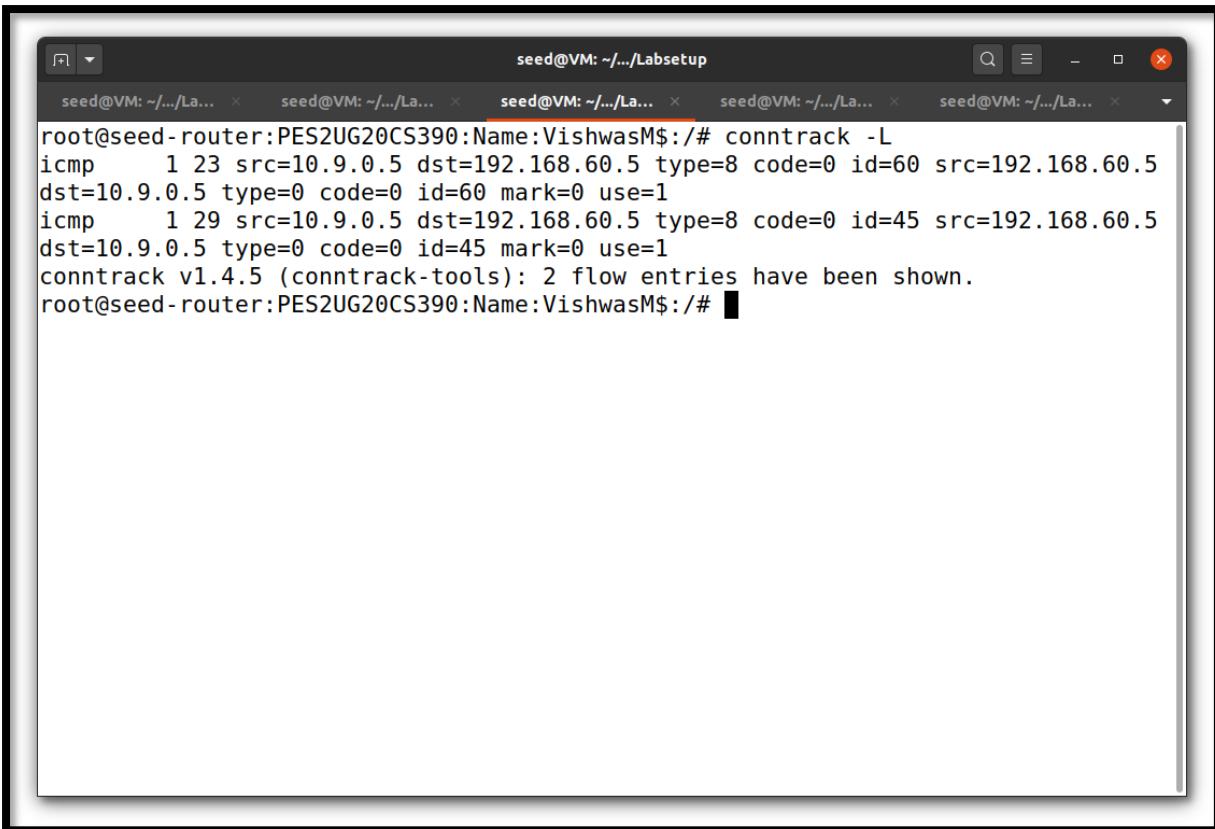
The goal of this task is to use a series of experiments to understand the connection concept in this tracking mechanism, especially for the ICMP and UDP protocols, unlike TCP, they do not have connections.

ICMP Experiment:



```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=0.111 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.112 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.583 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.123 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.121 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.176 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.176 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.121 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=1.06 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.179 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.166 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.158 ms
^C
--- 192.168.60.5 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12260ms
rtt min/avg/max/mdev = 0.111/0.246/1.061/0.263 ms
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:#
```

We are pinging from an external host to an internal host to observe the connection status inside the router.



The screenshot shows a terminal window titled "seed@VM: ~/.../Labsetup". It displays the output of the "conntrack -L" command. The output shows two ICMP entries:

```
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# conntrack -L
icmp    1 23 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=60 src=192.168.60.5
dst=10.9.0.5 type=0 code=0 id=60 mark=0 use=1
icmp    1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=45 src=192.168.60.5
dst=10.9.0.5 type=0 code=0 id=45 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 2 flow entries have been shown.
root@seed-router:PES2UG20CS390:Name:VishwasM$:/#
```

After stopping the connection in the external network, with the help of the command used in the above ss, we can find the connection status inside the router.

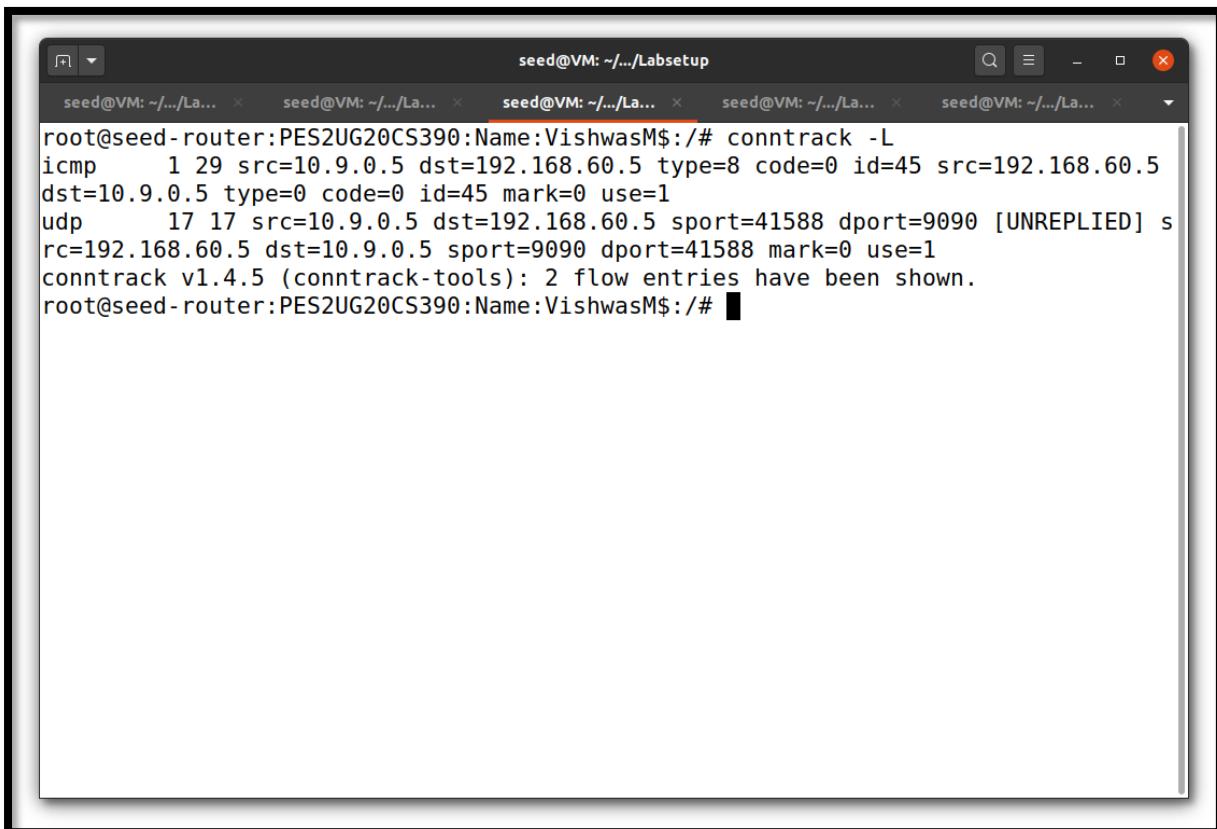
UDP Experiment:

First we are establishing a UDP connection between the external host(host A) and the internal host(host 1).

```
seed@VM: ~/.../Labsetup
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# nc -u 192.168.60.5 9090
Hello vishwas
CNS assignment is humungous
India won against Pakistan!!!
This sem is turning out to be a nightmare :')
Anyway I will finish off this assignment asap
byeee :))
^C
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:#
```

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../La... seed@VM: ~/.../La... seed@VM: ~/.../La... seed@VM: ~/.../La... seed@VM: ~/.../La...
root@seed-host1:PES2UG20CS390:Name:VishwasM$:# nc -lu 9090
Hello vishwas
CNS assignment is humungous
India won against Pakistan!!!
This sem is turning out to be a nightmare :')
Anyway I will finish off this assignment asap
byeee :))
```

After performing some conversations between the two hosts, we have to cut the connection.



The screenshot shows a terminal window titled "seed@VM: ~/Labsetup". It displays the output of the "conntrack -L" command. The output shows two entries: an ICMP connection (id=45) and a UDP connection (sport=41588, dport=9090). The UDP connection is labeled as "[UNREPLIED]". The message "conntrack v1.4.5 (conntrack-tools): 2 flow entries have been shown." is also present. The terminal window has multiple tabs at the top, all showing the same command prompt.

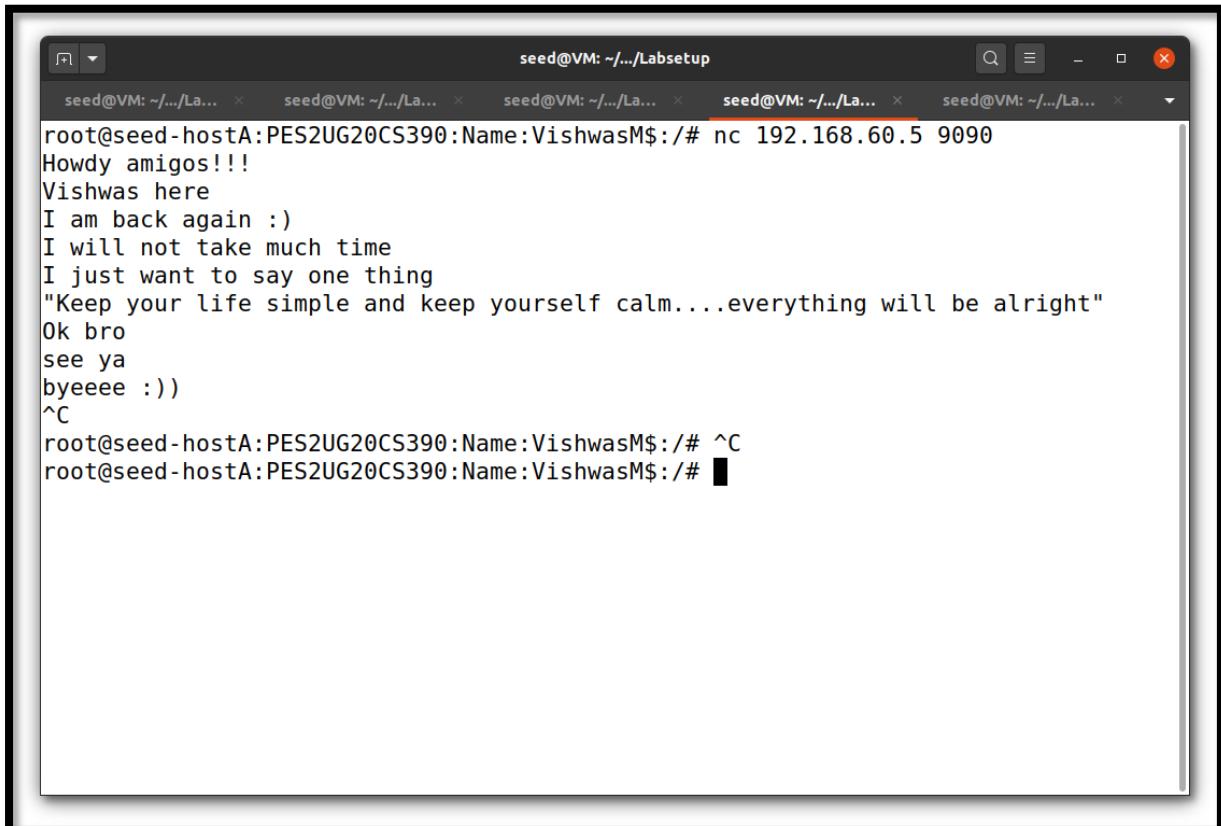
```
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# conntrack -L
icmp      1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=45 src=192.168.60.5
dst=10.9.0.5 type=0 code=0 id=45 mark=0 use=1
udp      17 17 src=10.9.0.5 dst=192.168.60.5 sport=41588 dport=9090 [UNREPLIED] s
rc=192.168.60.5 dst=10.9.0.5 sport=9090 dport=41588 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 2 flow entries have been shown.
root@seed-router:PES2UG20CS390:Name:VishwasM$:/#
```

After cutting the connection, we have to execute the above command to see the connection status inside the router. We can see that the UDP connection was established inside the router as shown in the above ss.

When we perform this experiment multiple times the time will reduce gradually.

TCP Experiment:

First we have to establish a TCP connection between the external host(host A) and the internal host(host 1).



The screenshot shows a terminal window with five tabs, all labeled "seed@VM: ~.../Labsetup". The fourth tab is active. The terminal output is as follows:

```
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# nc 192.168.60.5 9090
Howdy amigos!!!
Vishwas here
I am back again :)
I will not take much time
I just want to say one thing
"Keep your life simple and keep yourself calm....everything will be alright"
Ok bro
see ya
byeeee :))
^C
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# ^C
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:#
```

```
seed@VM: ~/.../Labsetup
root@seed-host1:PES2UG20CS390:Name:VishwasM$:# nc -l 9090
Howdy amigos!!!
Vishwas here
I am back again :)
I will not take much time
I just want to say one thing
"Keep your life simple and keep yourself calm....everything will be alright"
Ok bro
see ya
byeeee :))
root@seed-host1:PES2UG20CS390:Name:VishwasM$:# ^C
root@seed-host1:PES2UG20CS390:Name:VishwasM$:#
```

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../La... seed@VM: ~/.../La... seed@VM: ~/.../La... seed@VM: ~/.../La... seed@VM: ~/.../La...
root@seed-router:PES2UG20CS390:Name:VishwasM$:# conntrack -L
tcp      6 62 TIME_WAIT src=10.9.0.5 dst=192.168.60.5 sport=39724 dport=9090 src=
192.168.60.5 dst=10.9.0.5 sport=9090 dport=39724 [ASSURED] mark=0 use=1
icmp     1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=45 src=192.168.60.5
dst=10.9.0.5 type=0 code=0 id=45 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 2 flow entries have been shown.
root@seed-router:PES2UG20CS390:Name:VishwasM$:#
```

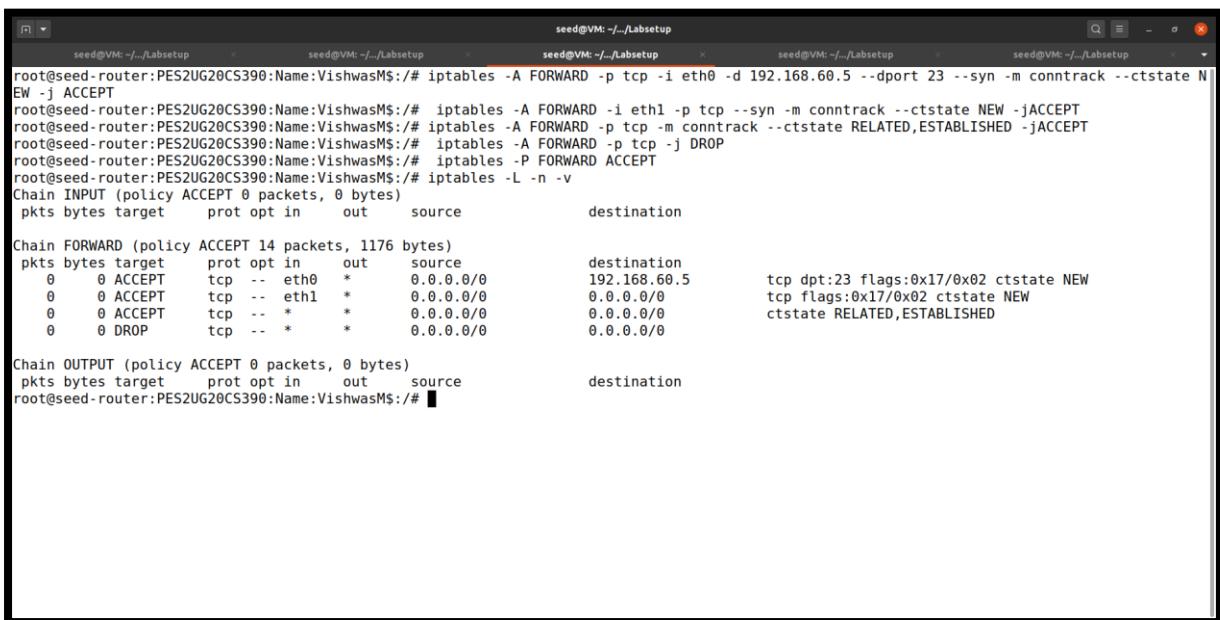
As we can see that TCP connection is shown in the router.

Difference:

In UDP, if we close the connection from host A(external) and still the host 1(internal) is still not closed, then only we can see the UDP connection inside the router. If we cut connection from both sides then the UDP connection will not be shown inside the router.

But in case of TCP, if we close the connection one side, then automatically it closes on the other side also. And we can see the TCP connection inside the router.

Task 3.B: Setting up a Stateful Firewall



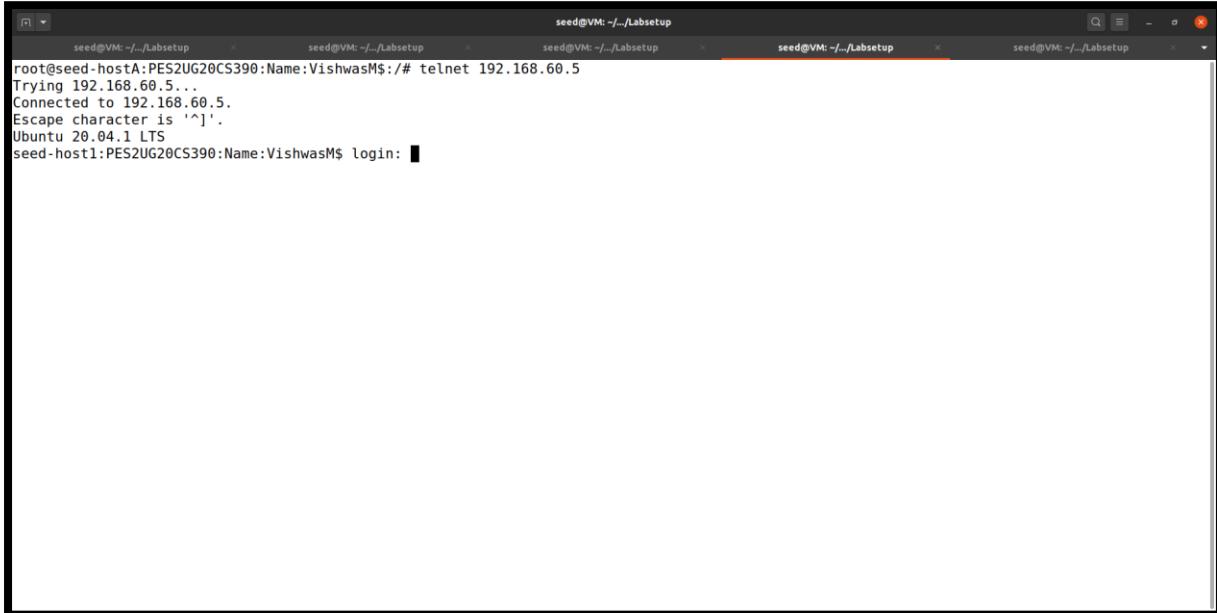
The screenshot shows a terminal window with five tabs, all titled 'seed@VM: .../Labsetup'. The terminal displays the following command and its output:

```
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -A FORWARD -p tcp -i eth0 -d 192.168.60.5 --dport 23 --syn -m conntrack --ctstate NEW -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -A FORWARD -i eth1 -p tcp --syn -m conntrack --ctstate NEW -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -A FORWARD -p tcp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -P FORWARD ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out    source          destination
Chain FORWARD (policy ACCEPT 14 packets, 1176 bytes)
 pkts bytes target     prot opt in     out    source          destination
  0     0 ACCEPT      tcp  --  eth0   *       0.0.0.0/0      192.168.60.5      tcp dpt:23 flags:0x17/0x02 ctstate NEW
  0     0 ACCEPT      tcp  --  eth1   *       0.0.0.0/0      0.0.0.0/0      tcp flags:0x17/0x02 ctstate NEW
  0     0 ACCEPT      tcp  --   *    *       0.0.0.0/0      0.0.0.0/0      ctstate RELATED,ESTABLISHED
  0     0 DROP        tcp  --   *    *       0.0.0.0/0      0.0.0.0/0
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out    source          destination
root@seed-router:PES2UG20CS390:Name:VishwasM$:/#
```

We added some rules to the firewall to set up some restrictions in the firewall for the firewall to work like stateful.

We will check if the restrictions are working or not:

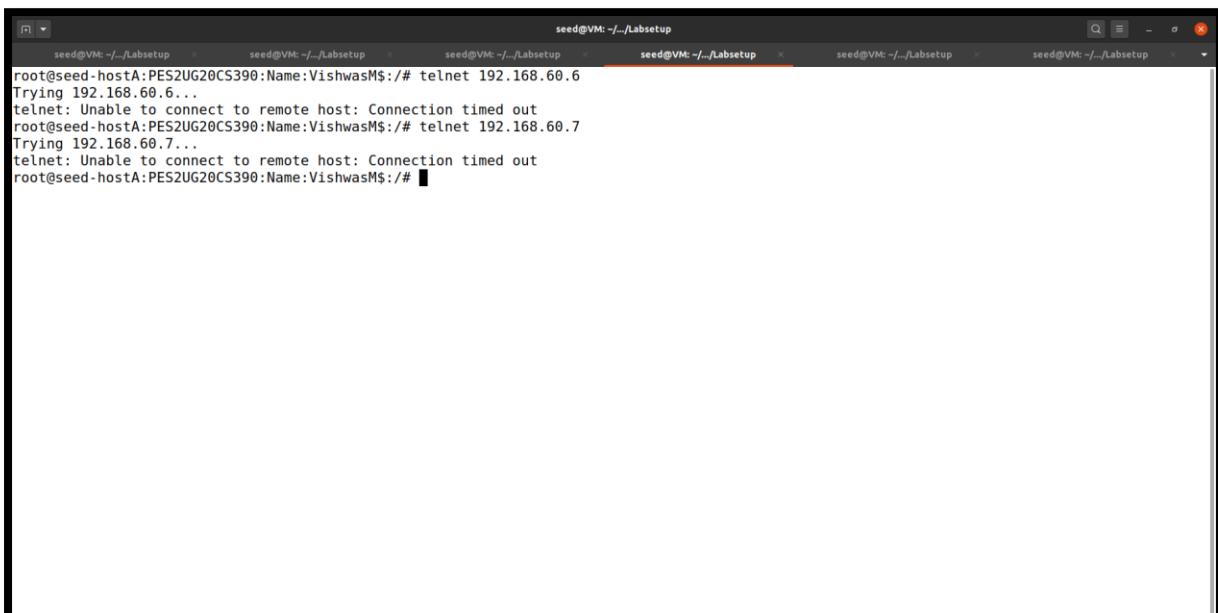
1. Outside hosts can only access the telnet server on 192.168.60.5, not the other internal hosts.



```
root@seed-hostA:PES2UG20CS390:Name:VishwasM$ telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^].
Ubuntu 20.04.1 LTS
seed-host1:PES2UG20CS390:Name:VishwasM$ login: [REDACTED]
```

We can clearly see that the connection is established between outside host and server 192.168.60.5

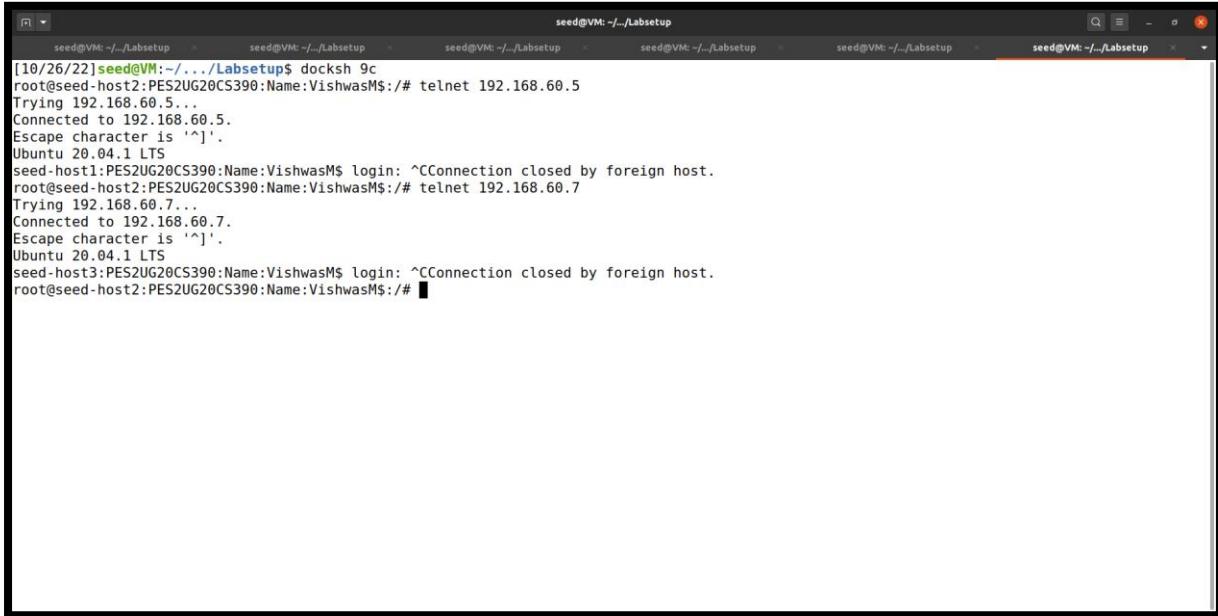
2. Outside hosts cannot access other internal servers



```
root@seed-hostA:PES2UG20CS390:Name:VishwasM$ telnet 192.168.60.6
Trying 192.168.60.6...
telnet: Unable to connect to remote host: Connection timed out
root@seed-hostA:PES2UG20CS390:Name:VishwasM$ telnet 192.168.60.7
Trying 192.168.60.7...
telnet: Unable to connect to remote host: Connection timed out
root@seed-hostA:PES2UG20CS390:Name:VishwasM$ [REDACTED]
```

We can see that the connection failed to establish between host and other internal servers.

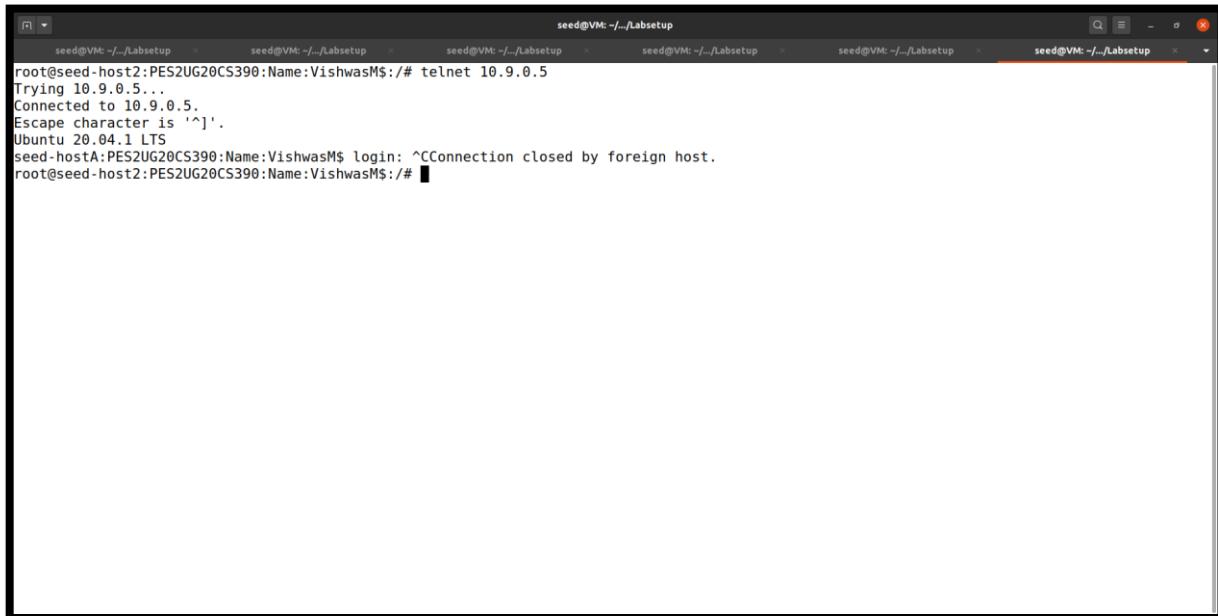
3. Internal hosts can access all the servers.



```
[10/26/22] seed@VM: ~/Labsetup$ docksh 9c
root@seed-host2:PES2UG20CS390:Name:VishwasM$:# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
seed-host1:PES2UG20CS390:Name:VishwasM$ login: ^CConnection closed by foreign host.
root@seed-host2:PES2UG20CS390:Name:VishwasM$:# telnet 192.168.60.7
Trying 192.168.60.7...
Connected to 192.168.60.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
seed-host3:PES2UG20CS390:Name:VishwasM$ login: ^CConnection closed by foreign host.
root@seed-host2:PES2UG20CS390:Name:VishwasM$:#
```

The connection is well established between internal hosts and all the servers.

4. Internal hosts can access the external servers.

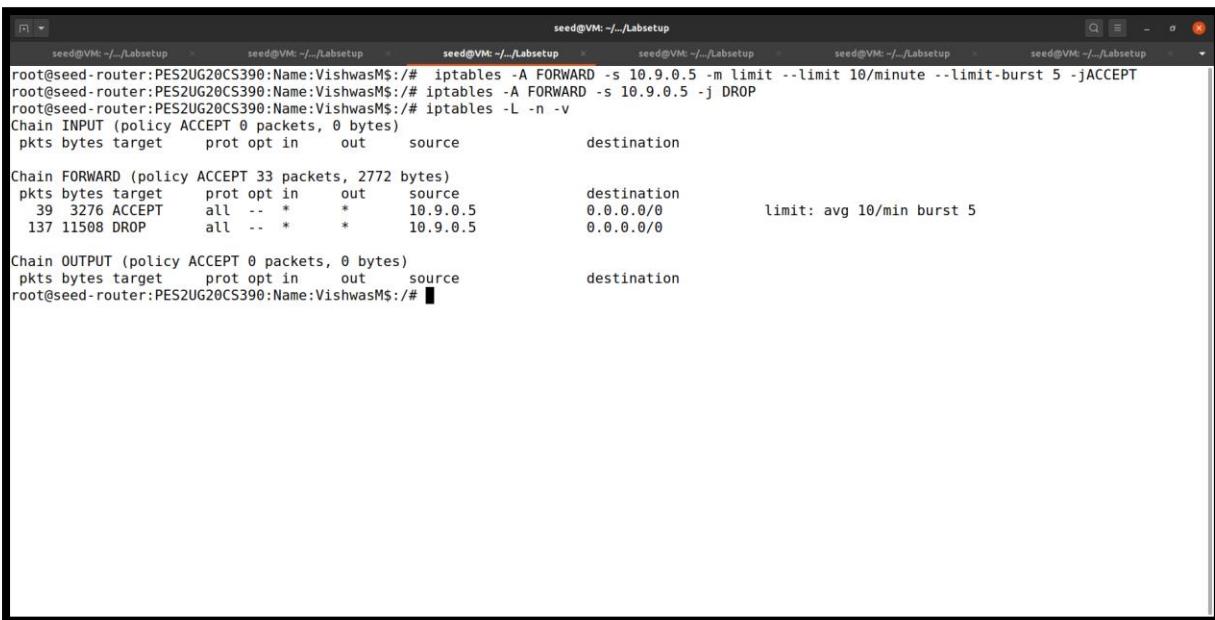


```
root@seed-host2:PES2UG20CS390:Name:VishwasM$:# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
seed-hostA:PES2UG20CS390:Name:VishwasM$ login: ^CConnection closed by foreign host.
root@seed-host2:PES2UG20CS390:Name:VishwasM$:#
```

Connection is well established between internal hosts and external servers.

Task 4: Limiting Network Traffic

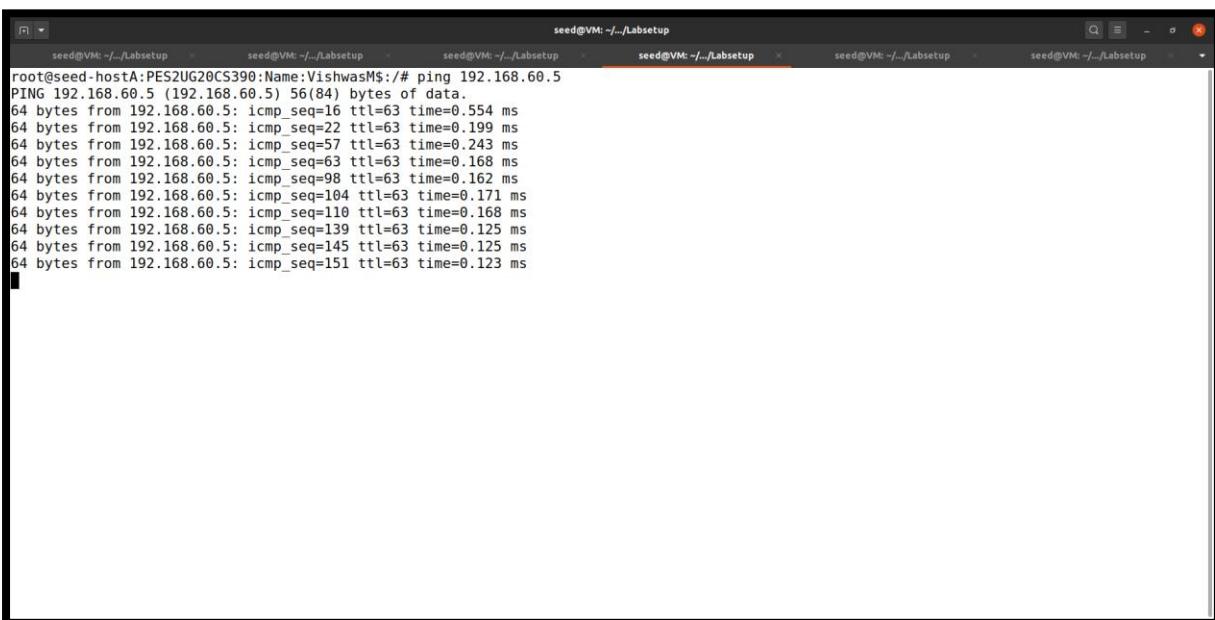
a) With dropping packets:



The screenshot shows a terminal window with six tabs, all titled 'seed@VM: ~/Labsetup'. The active tab displays the following command and its output:

```
root@seed-router:PES2UG20CS390:Name:VishwasM$: # iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -j ACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$: # iptables -A FORWARD -s 10.9.0.5 -j DROP
root@seed-router:PES2UG20CS390:Name:VishwasM$: # iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out      source          destination
Chain FORWARD (policy ACCEPT 33 packets, 2772 bytes)
pkts bytes target     prot opt in     out      source          destination
  39  3276 ACCEPT    all   --  *       10.9.0.5        0.0.0.0/0      limit: avg 10/min burst 5
 137 11508 DROP      all   --  *       10.9.0.5        0.0.0.0/0
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out      source          destination
root@seed-router:PES2UG20CS390:Name:VishwasM$:/#
```

We have set the rules for the firewall here to drop the extra packets.

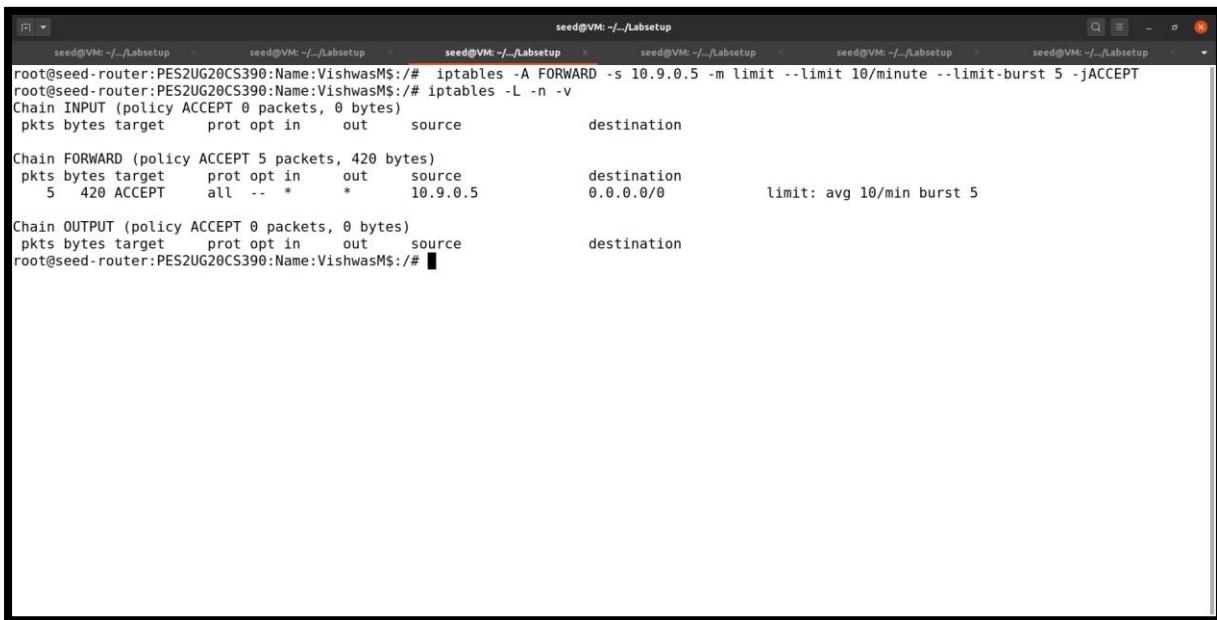


The screenshot shows a terminal window with six tabs, all titled 'seed@VM: ~/Labsetup'. The active tab displays the following command and its output:

```
root@seed-hostA:PES2UG20CS390:Name:VishwasM$: # 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=16 ttl=63 time=0.554 ms
64 bytes from 192.168.60.5: icmp_seq=22 ttl=63 time=0.199 ms
64 bytes from 192.168.60.5: icmp_seq=57 ttl=63 time=0.243 ms
64 bytes from 192.168.60.5: icmp_seq=63 ttl=63 time=0.168 ms
64 bytes from 192.168.60.5: icmp_seq=98 ttl=63 time=0.162 ms
64 bytes from 192.168.60.5: icmp_seq=104 ttl=63 time=0.171 ms
64 bytes from 192.168.60.5: icmp_seq=110 ttl=63 time=0.168 ms
64 bytes from 192.168.60.5: icmp_seq=139 ttl=63 time=0.125 ms
64 bytes from 192.168.60.5: icmp_seq=145 ttl=63 time=0.125 ms
64 bytes from 192.168.60.5: icmp_seq=151 ttl=63 time=0.123 ms
```

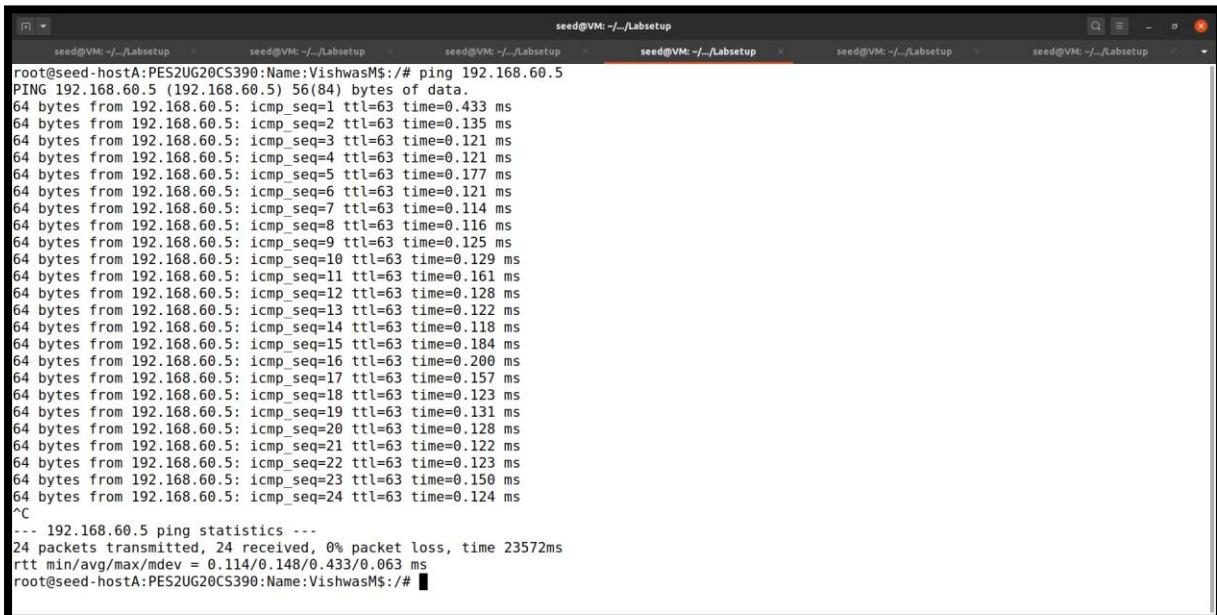
Only 10 packets are allowed to transmit through the firewall per minute, rest of the packets are dropped.

b) Without dropping the packets



```
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -jACCEPT
root@seed-router:PES2UG20CS390:Name:VishwasM$:# iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in     out    source          destination
Chain FORWARD (policy ACCEPT 5 packets, 420 bytes)
pkts bytes target prot opt in     out    source          destination
      5   420 ACCEPT  all   --  *       10.9.0.5           0.0.0.0/0        limit: avg 10/min burst 5
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in     out    source          destination
root@seed-router:PES2UG20CS390:Name:VishwasM$:#
```

The rule which tells to drop the packets is not uploaded to the firewall.

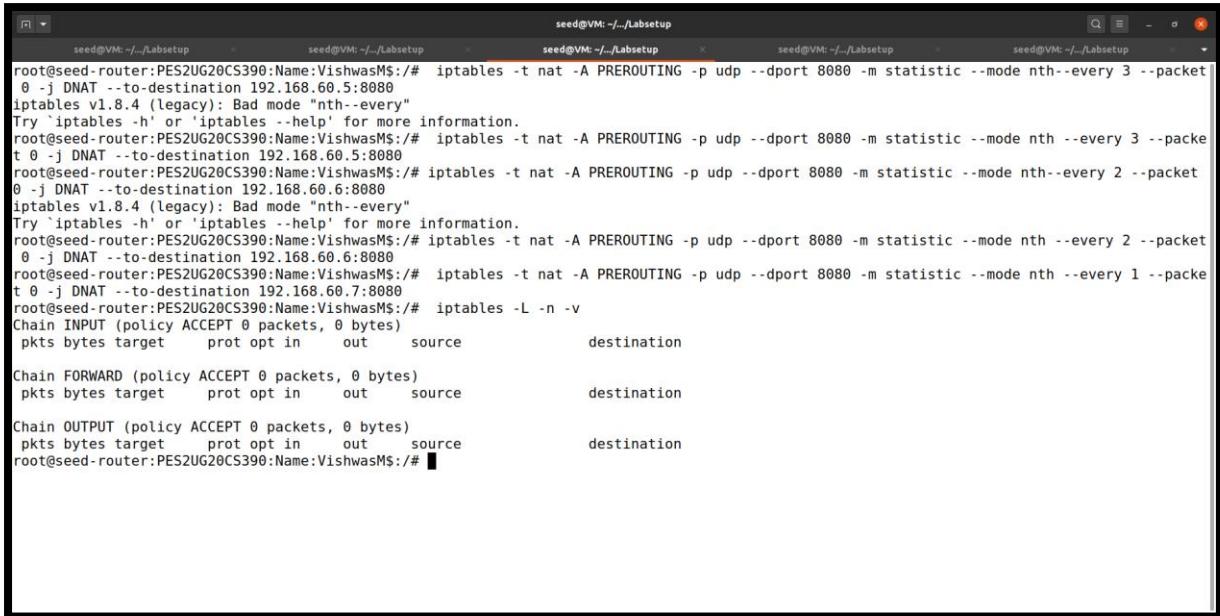


```
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=0.433 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.135 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.121 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.121 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.177 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.121 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.114 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.116 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.125 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.129 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.161 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.128 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.122 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.184 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.200 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.157 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.123 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.131 ms
64 bytes from 192.168.60.5: icmp_seq=20 ttl=63 time=0.128 ms
64 bytes from 192.168.60.5: icmp_seq=21 ttl=63 time=0.122 ms
64 bytes from 192.168.60.5: icmp_seq=22 ttl=63 time=0.123 ms
64 bytes from 192.168.60.5: icmp_seq=23 ttl=63 time=0.150 ms
64 bytes from 192.168.60.5: icmp_seq=24 ttl=63 time=0.124 ms
^C
--- 192.168.60.5 ping statistics ---
24 packets transmitted, 24 received, 0% packet loss, time 23572ms
rtt min/avg/max/mdev = 0.114/0.148/0.433/0.063 ms
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:#
```

The packets will not drop the packets which will result in sending the packets to destination server.

Task 5: Load Balancing

a. Using the nth mode (round-robin):



```
seed@VM:~/Labsetup
root@seed-router:PES2UG20CS390:Name:VishwasMS:# 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
iptables v1.8.4 (legacy): Bad mode "nth--every"
Try `iptables -h` or `iptables --help` for more information.
root@seed-router:PES2UG20CS390:Name:VishwasMS:# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth --every 3 --packe
t 0 -j DNAT --to-destination 192.168.60.5:8080
root@seed-router:PES2UG20CS390:Name:VishwasMS:# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth--every 2 --packet
0 -j DNAT --to-destination 192.168.60.5:8080
iptables v1.8.4 (legacy): Bad mode "nth--every"
Try `iptables -h` or `iptables --help` for more information.
root@seed-router:PES2UG20CS390:Name:VishwasMS:# 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@seed-router:PES2UG20CS390:Name:VishwasMS:# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth --every 1 --packet
0 -j DNAT --to-destination 192.168.60.7:8080
root@seed-router:PES2UG20CS390:Name:VishwasMS:# iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source         destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source         destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source         destination
root@seed-router:PES2UG20CS390:Name:VishwasMS:#
```

We are updating the rules in the firewall.

Host A:

```
root@seed-hostA:PES2UG20CS390:Name:VishwasM$:/# nc -u 10.9.0.11 8080
vishwas
Manjunath
PES2UG20CS390
```

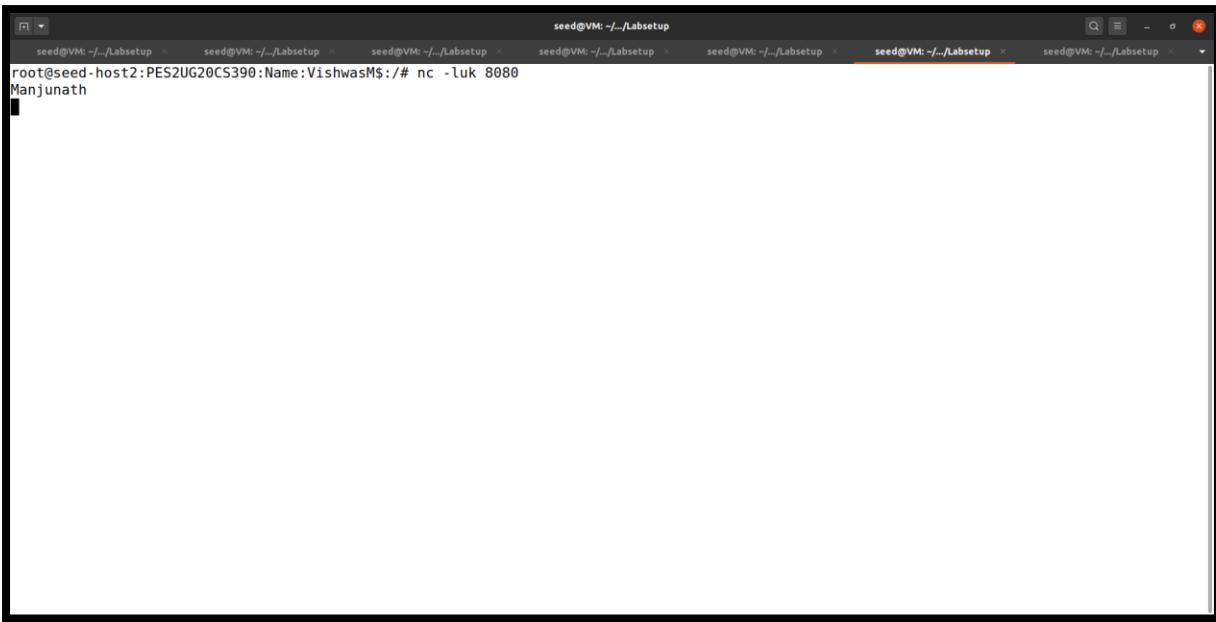
We have leave 30 sec gap between each word else all the 3 words will be considered as a single packet and go to the same host.

Host 1:

```
root@seed-host1:PES2UG20CS390:Name:VishwasM$:/# nc -luk 8080
vishwas
```

We can see that the first word is coming to host 1.

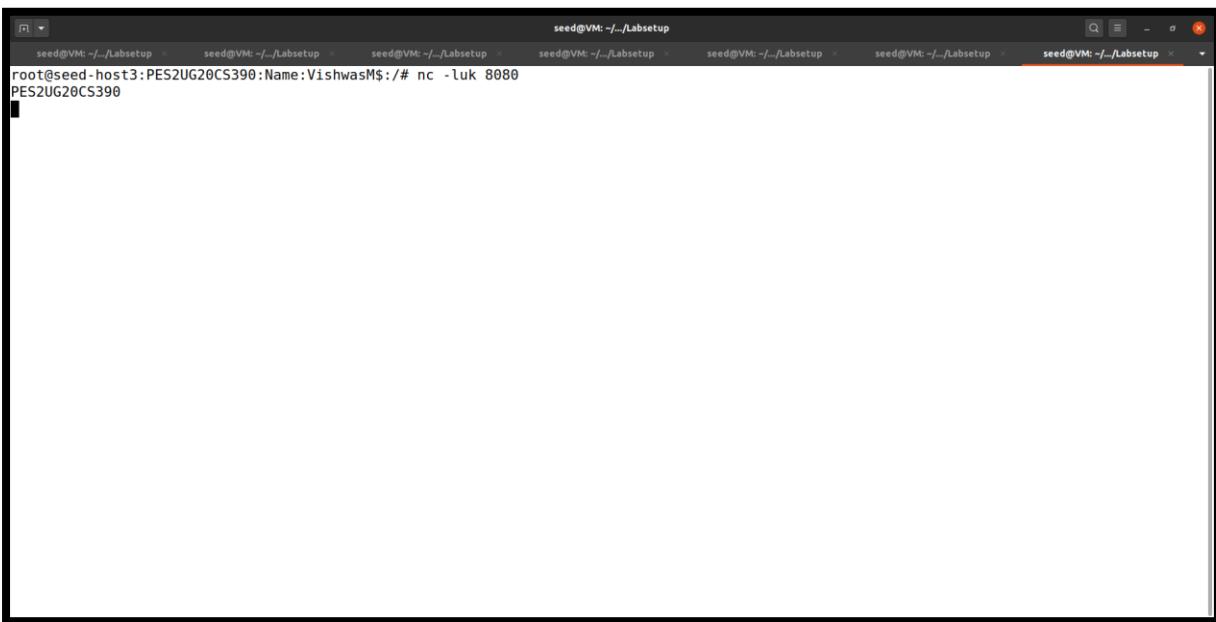
Host 2:



```
seed@VM: ~/Labsetup
root@seed-host2:PES2UG20CS390:Name:VishwasM$:/# nc -luk 8080
Manjunath
```

We can see that the 2nd word is coming to host 2

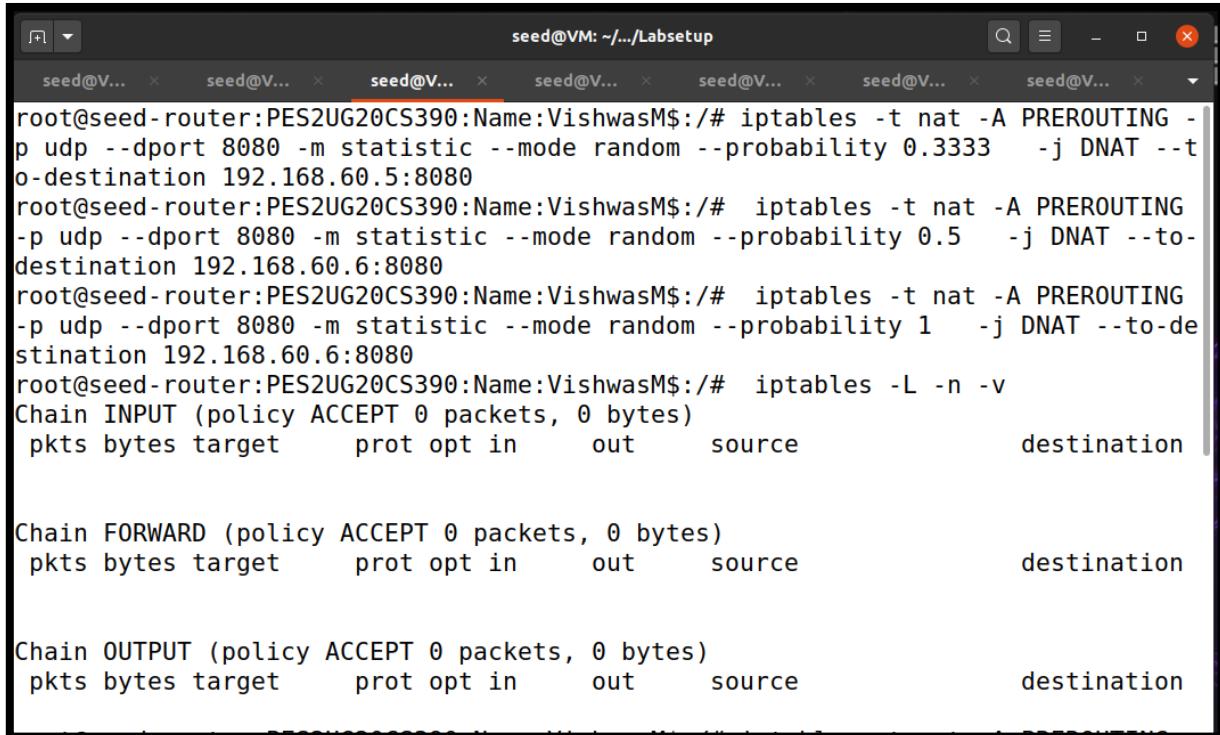
Host 3:



```
seed@VM: ~/Labsetup
root@seed-host3:PES2UG20CS390:Name:VishwasM$:/# nc -luk 8080
PES2UG20CS390
```

We can see that 3rd word is coming to host 3.

b. Using random mode: The following rule will select a matching packet with the probability P. You need to replace P with a probability number.



A screenshot of a terminal window titled "seed@VM: ~.../Labsetup". The window contains several tabs, all labeled "seed@V...". The main pane displays the following command output:

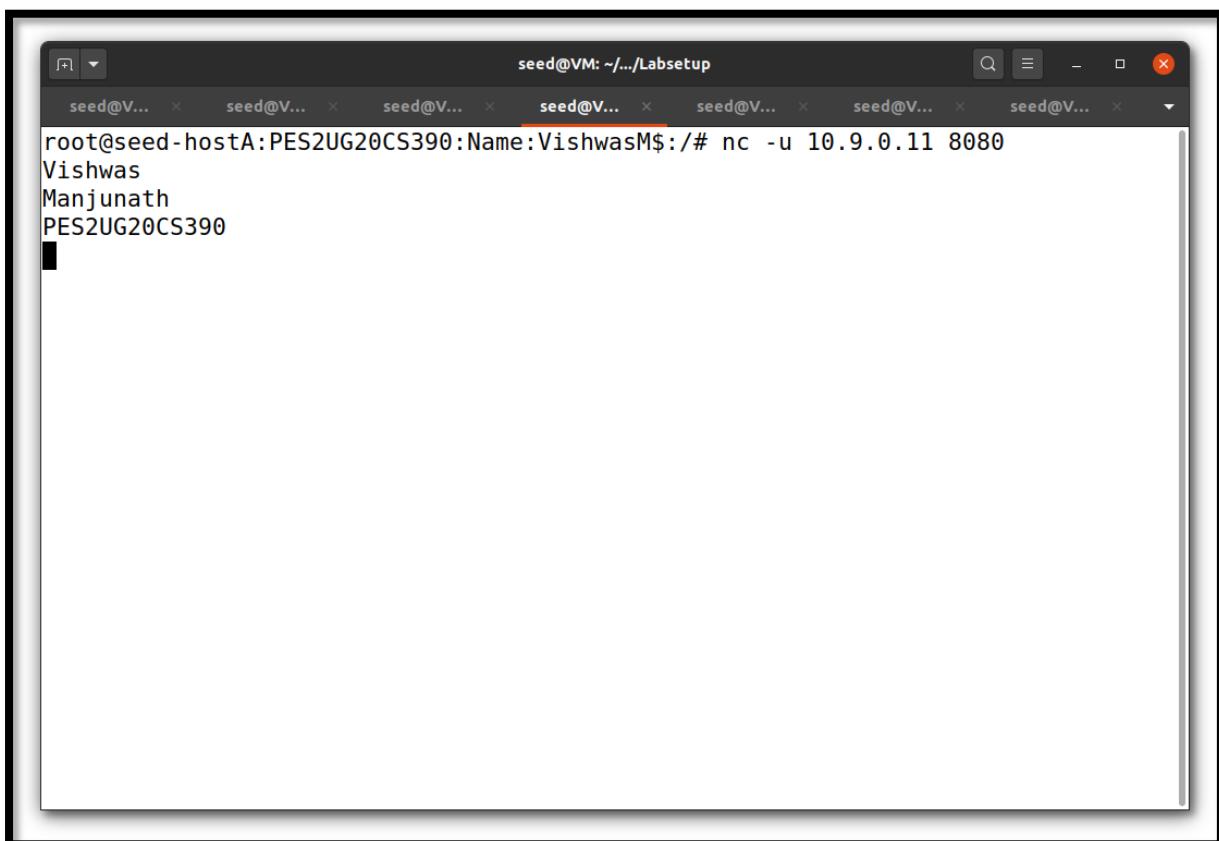
```
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 0.3333 -j DNAT --to-destination 192.168.60.5:8080
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 0.5 -j DNAT --to-destination 192.168.60.6:8080
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 1 -j DNAT --to-destination 192.168.60.6:8080
root@seed-router:PES2UG20CS390:Name:VishwasM$:/# iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in      out      source          destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in      out      source          destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in      out      source          destination
```

We have updated the rules of the server.

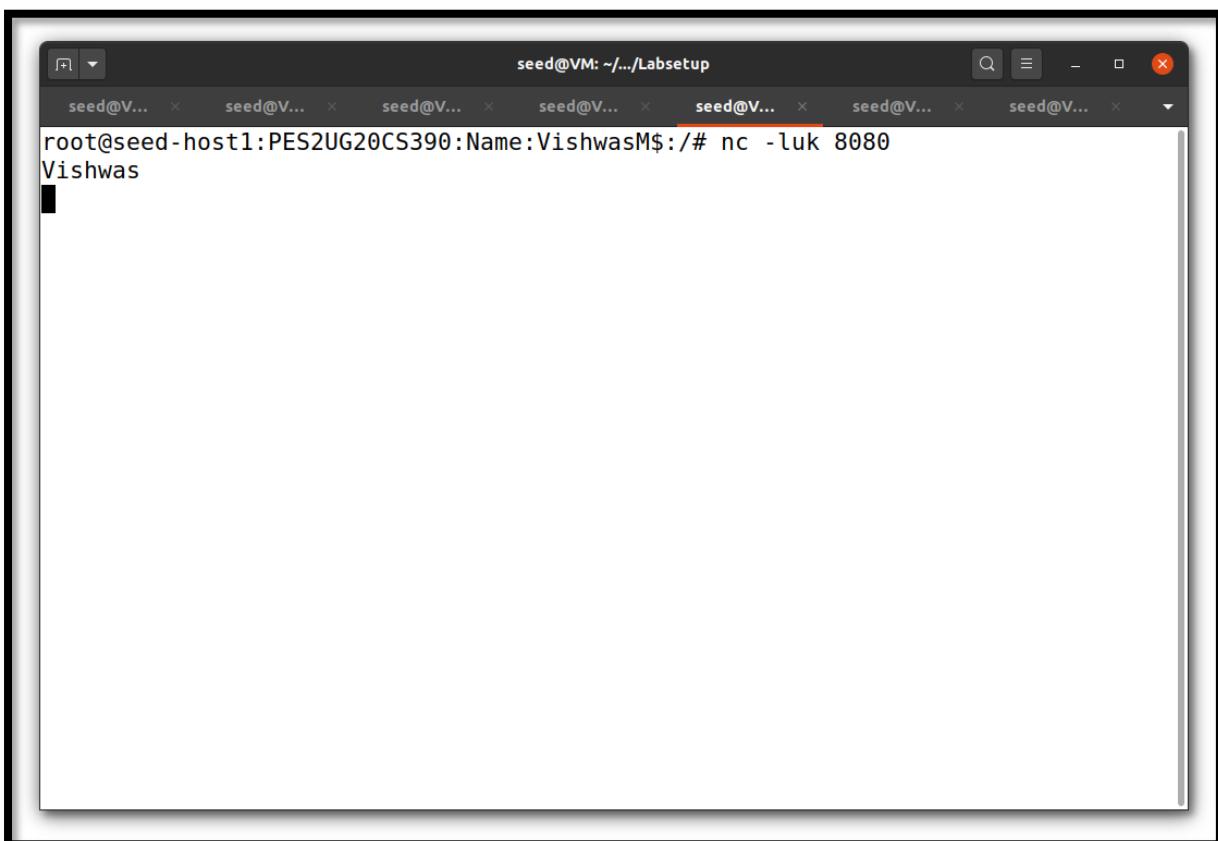
Host A:



A screenshot of a terminal window titled "seed@VM: ~/.../Labsetup". The window has multiple tabs, with the active tab labeled "seed@V...". The command entered is "root@seed-hostA:PES2UG20CS390:Name:VishwasM\$ nc -u 10.9.0.11 8080". The output shows three lines of text being sent: "Vishwas", "Manjunath", and "PES2UG20CS390".

We have leave 30 sec gap between each word else all the 3 words will be considered as a single packet and go to the same host.

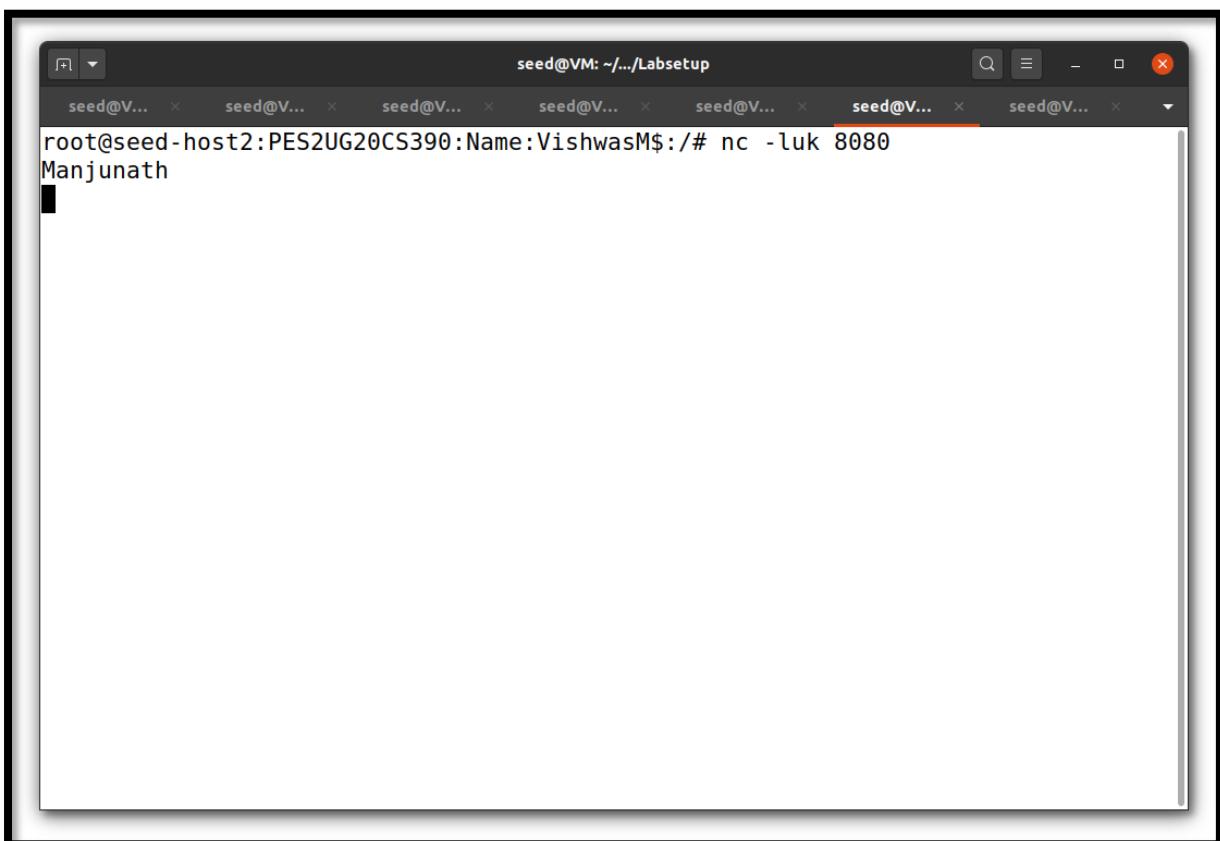
Host 1:



A screenshot of a terminal window titled "seed@VM: ~/.../Labsetup". The window contains several tabs, all labeled "seed@V...". The active tab shows the command "root@seed-host1:PES2UG20CS390:Name:VishwasM\$:/# nc -luk 8080" followed by the output "Vishwas".

We can see that the first word is coming to host 1.

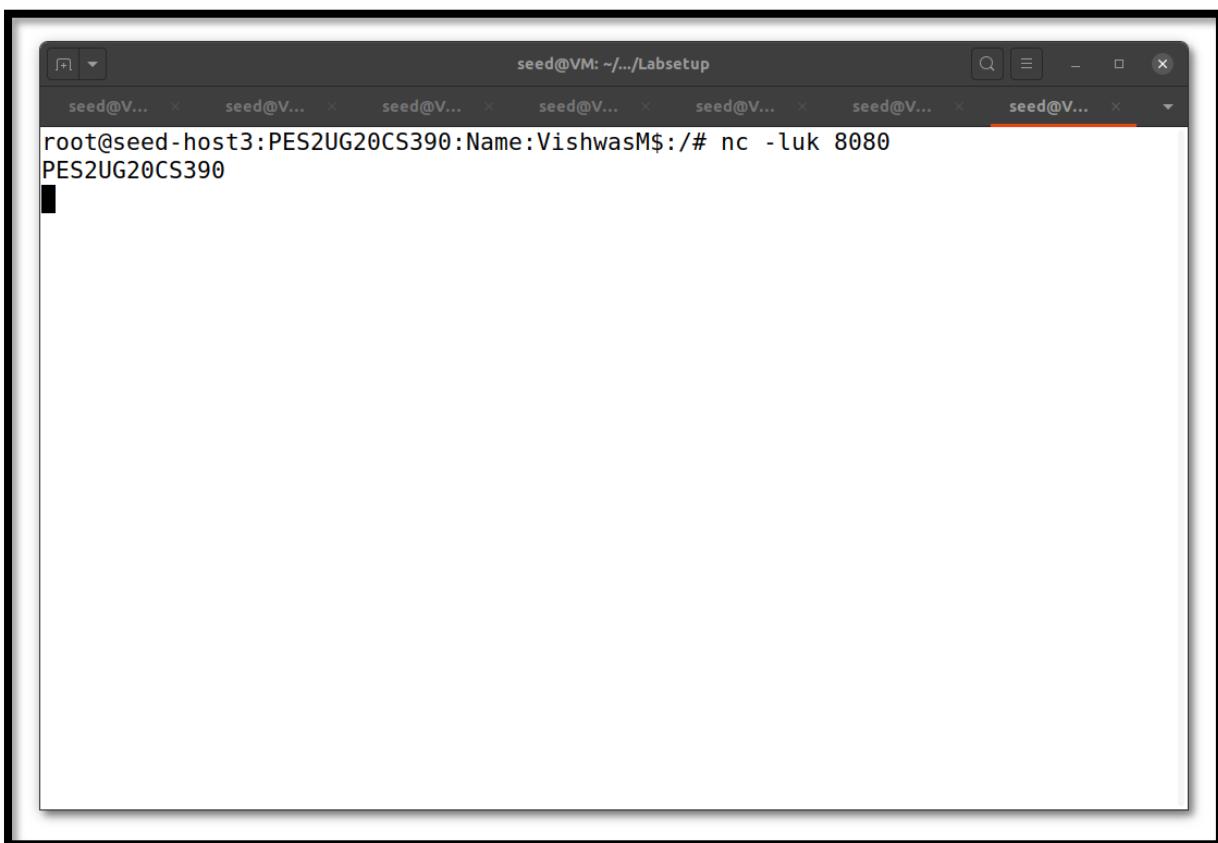
Host 2:



A screenshot of a terminal window titled "seed@VM: ~/.../Labsetup". The window contains several tabs, all labeled "seed@V...". The active tab shows the command "root@seed-host2:PES2UG20CS390:Name:VishwasM\$ nc -luk 8080" followed by the output "Manjunath".

We can see that the second word is coming to host 2.

Host 3:



A screenshot of a terminal window titled "seed@VM: ~/.../Labsetup". The window contains several tabs, with the rightmost tab, "seed@V...", highlighted by a red underline. The terminal output shows the command "root@seed-host3:PES2UG20CS390:Name:VishwasM\$ nc -luk 8080" followed by "PES2UG20CS390".

We can see that the third word is coming to host 3.