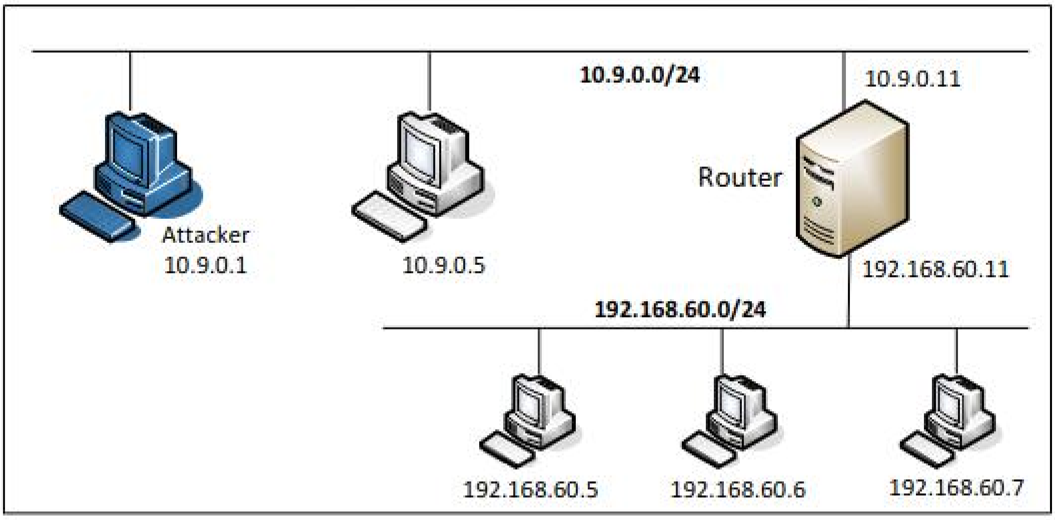
COMPUTER NETWORK SECURITY LABORATORY

NAME : PREM SAGAR J S	SRN : PES1UG20CS825	SEC : H

Firewall Exploration Lab

Lab Environment Setup



Task 1: Implementing a Simple Firewall

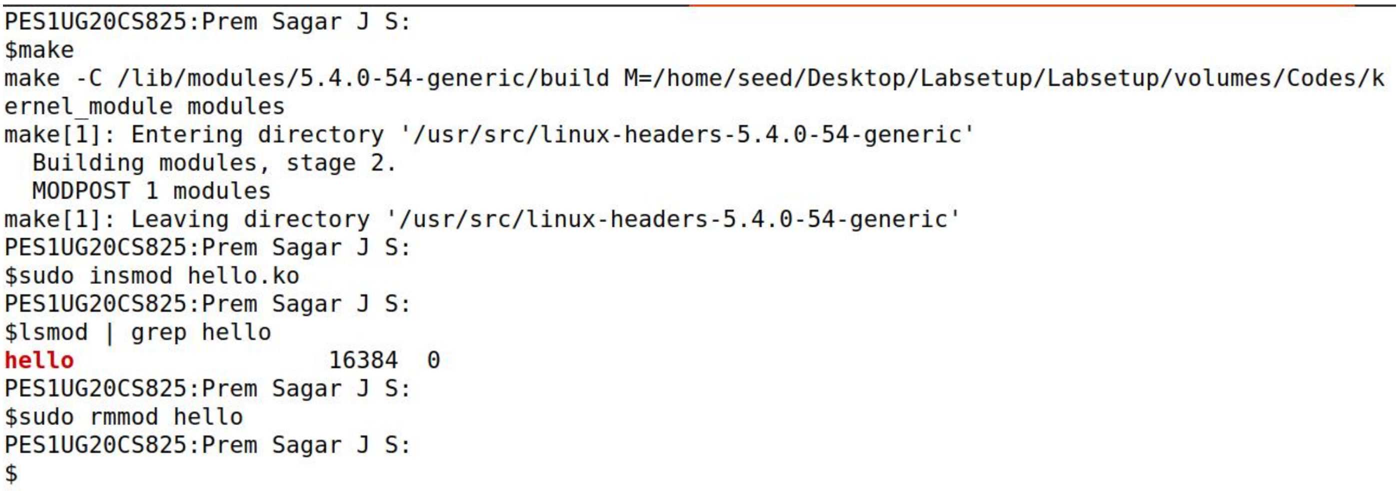
simple packet filtering type of firewall, which inspects each incoming and outgoing packet, and enforces the firewall policies set by the administrator. Since the packet processing is done within the kernel, the filtering must also be done within the kernel.

Task 1.A: Implement a Simple Kernel Module

- On the VM - Opening two Terminal Tabs, one to load the module and the other to view the
  messages.

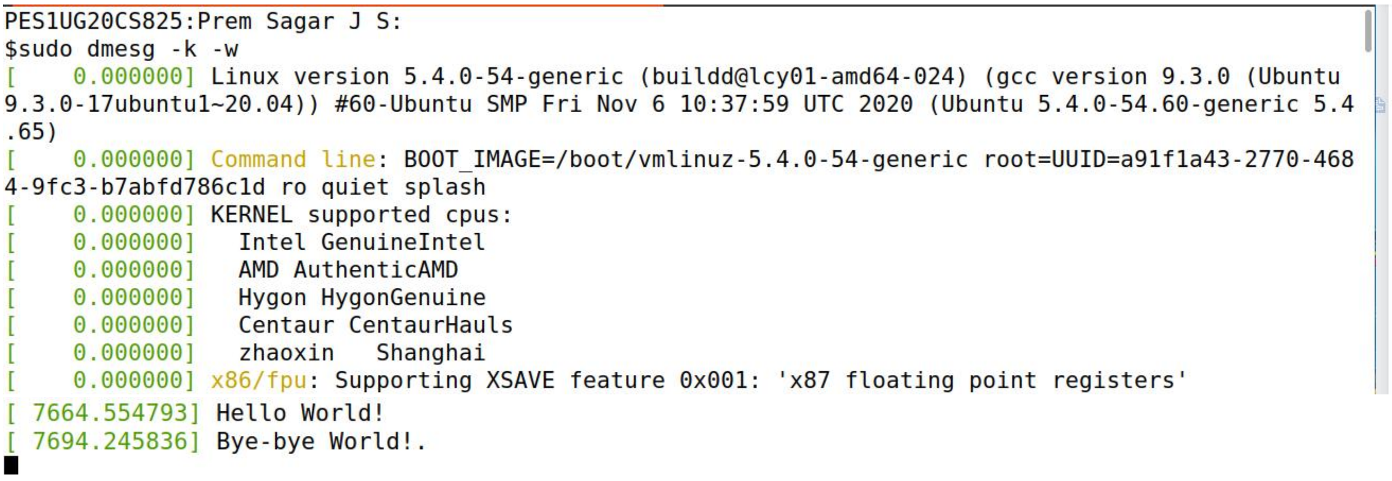
- The other to Load and Remove the Kernel
  - Command:

```
$ make
$ sudo insmod hello.ko (inserting a module)
$ lsmod | grep hello (list modules)
$ sudo rmmod hello
```



```
PES1UG20CS825:Prem Sagar J S:
$make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Desktop/Labsetup/Labsetup/volumes/Codes/k
ernel_module 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'
PES1UG20CS825:Prem Sagar J S:
$sudo insmod hello.ko
PES1UG20CS825:Prem Sagar J S:
$lsmod | grep hello
hello                  16384  0
PES1UG20CS825:Prem Sagar J S:
$sudo rmmod hello
PES1UG20CS825:Prem Sagar J S:
$
```

- Using one terminal window to view the messages
  - Command:
    $ sudo dmesg -k -w



```
PES1UG20CS825:Prem Sagar J S:
$sudo dmesg -k -w
[    0.000000] Linux version 5.4.0-54-generic (buildd@lcy01-amd64-024) (gcc version 9.3.0 (Ubuntu
9.3.0-17ubuntu1~20.04)) #60-Ubuntu SMP Fri Nov 6 10:37:59 UTC 2020 (Ubuntu 5.4.0-54.60-generic 5.4
.65)
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-54-generic root=UUID=a91f1a43-2770-468
4-9fc3-b7abfd786c1d ro quiet splash
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Hygon HygonGenuine
[    0.000000]   Centaur CentaurHauls
[    0.000000]   zhaoxin   Shanghai
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ 7664.554793] Hello World!
[ 7694.245836] Bye-bye World!.
```

Task 1.B: Implement a Simple Firewall Using Netfilter

Writing our packet filtering program as an LKM, and then insert it into the packet processing path inside the kernel.

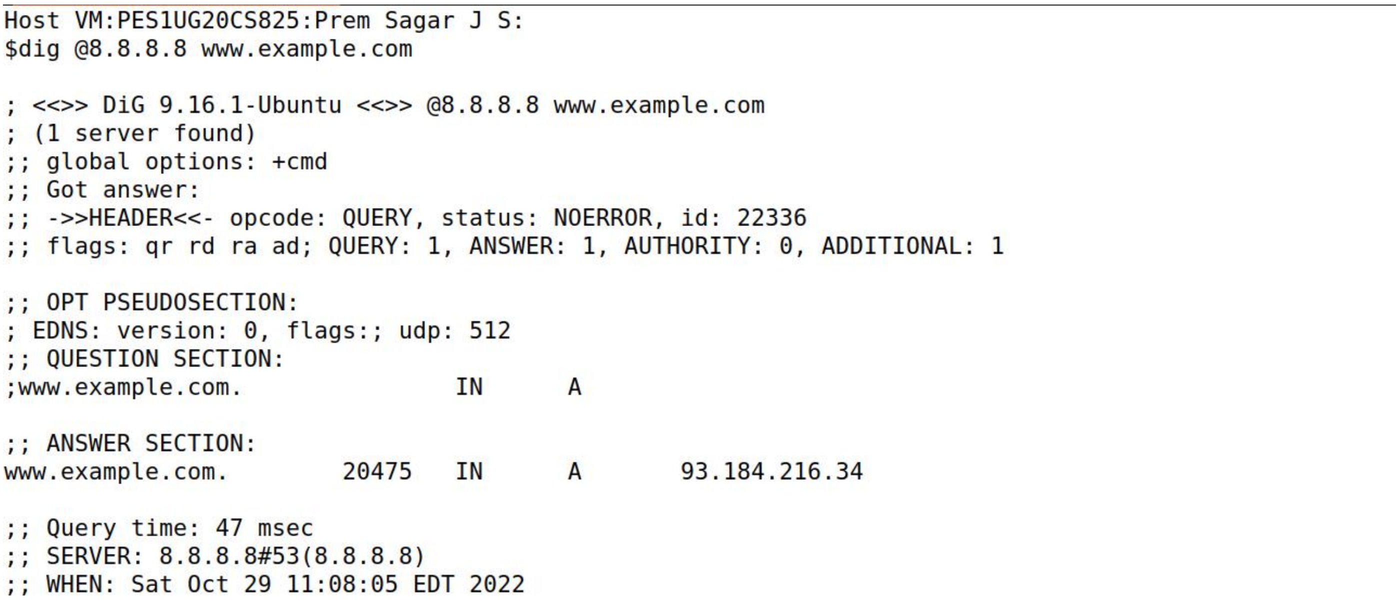
Tasks

- Compile the code using the provided Makefile. Load it into the kernel, and demonstrate that the firewall is working as expected.

- We are generate UDP packets to 8.8.8.8, which is Google’ s DNS server.

- making sure www.example.com is reachable.

Command:
$ dig @8.8.8.8 www.example.com



```
Host VM:PES1UG20CS825:Prem Sagar J S:
$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: 22336
;; 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.        20475   IN      A       93.184.216.34

;; Query time: 47 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sat Oct 29 11:08:05 EDT 2022
```

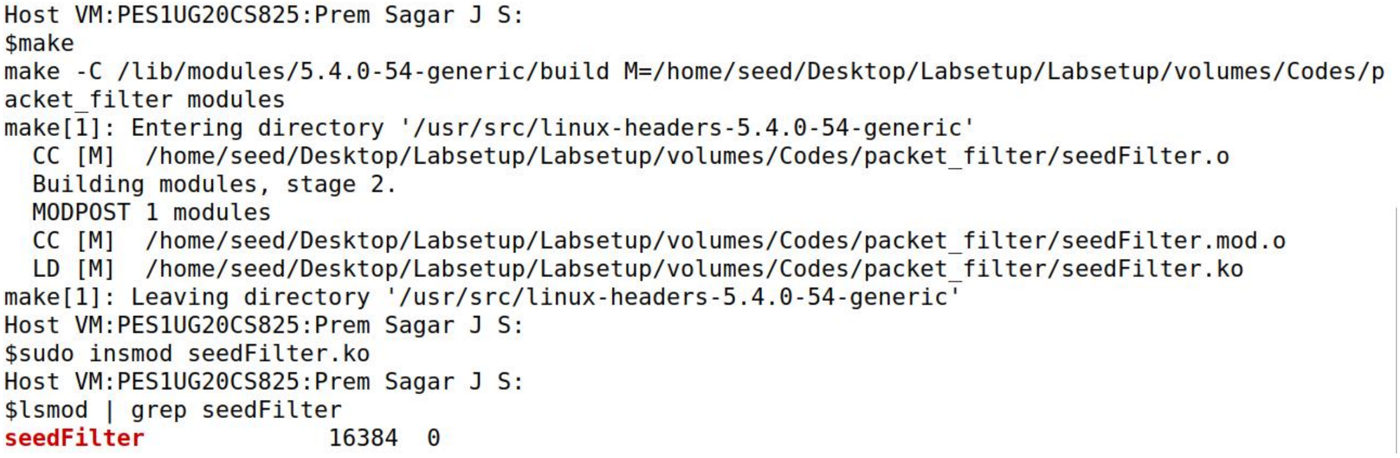
- I am able to reach www.example.com
- Inserting the kernel object.

Command:

$ make

$ sudo insmod seedFilter.ko

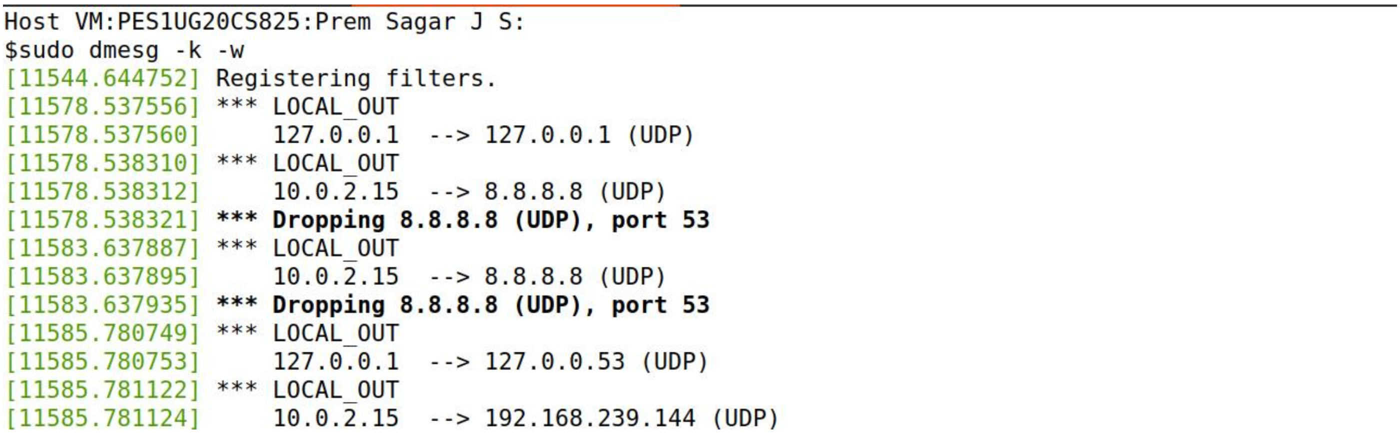
$ lsmod | grep seedFilter



```
Host VM:PES1UG20CS825:Prem Sagar J S:
$make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Desktop/Labsetup/Labsetup/volumes/Codes/p
acket_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/Desktop/Labsetup/Labsetup/volumes/Codes/packet_filter/seedFilter.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/seed/Desktop/Labsetup/Labsetup/volumes/Codes/packet_filter/seedFilter.mod.o
  LD [M]  /home/seed/Desktop/Labsetup/Labsetup/volumes/Codes/packet_filter/seedFilter.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
Host VM:PES1UG20CS825:Prem Sagar J S:
$sudo insmod seedFilter.ko
Host VM:PES1UG20CS825:Prem Sagar J S:
$lsmod | grep seedFilter
seedFilter             16384  0
```

Command:

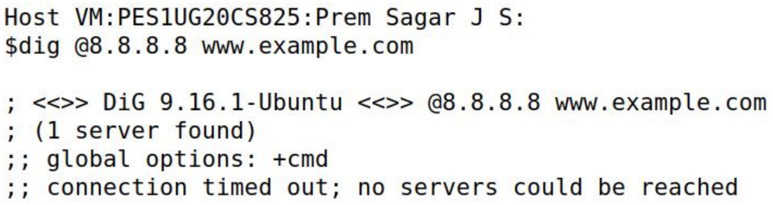
$ sudo dmesg -k -w



```
Host VM:PES1UG20CS825:Prem Sagar J S:
$sudo dmesg -k -w
[11544.644752] Registering filters.
[11578.537556] *** LOCAL_OUT
[11578.537560]     127.0.0.1  --> 127.0.0.1 (UDP)
[11578.538310] *** LOCAL_OUT
[11578.538312]     10.0.2.15  --> 8.8.8.8 (UDP)
[11578.538321] *** Dropping 8.8.8.8 (UDP), port 53
[11583.637887] *** LOCAL_OUT
[11583.637895]     10.0.2.15  --> 8.8.8.8 (UDP)
[11583.637935] *** Dropping 8.8.8.8 (UDP), port 53
[11585.780749] *** LOCAL_OUT
[11585.780753]     127.0.0.1  --> 127.0.0.53 (UDP)
[11585.781122] *** LOCAL_OUT
[11585.781124]     10.0.2.15  --> 192.168.239.144 (UDP)
```

- After inserting module executing the below commands to notice the difference.

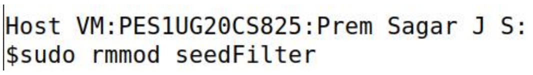
Command:
$ dig @8.8.8.8 www.example.com



```
Host VM:PES1UG20CS825:Prem Sagar J S:
$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
```

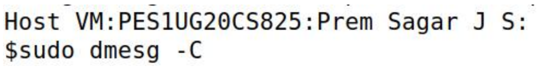
- After inserting the module im not able to reach www.example.com that indicates that our firewall is working fine.

- Removing the module.



```
Host VM:PES1UG20CS825:Prem Sagar J S:
$sudo rmmod seedFilter
```

- clearing the kernel messages.

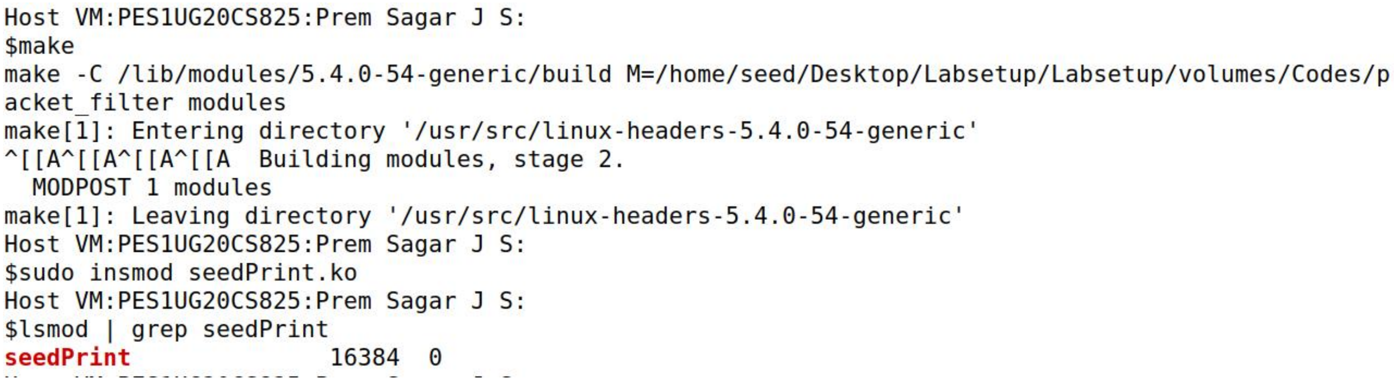


```
Host VM:PES1UG20CS825:Prem Sagar J S:
$sudo dmesg -C
```

2. Hook the printInfo function to all of the netfilter hooks. Here are the macros of the hook numbers. Using your experiment results to help explain at what condition each of the hook functions be invoked.

NF_INET_PRE_ROUTING
NF_INET_LOCAL_IN
NF_INET_FORWARD
NF_INET_LOCAL_OUT
NF_INET_POST_ROUTING

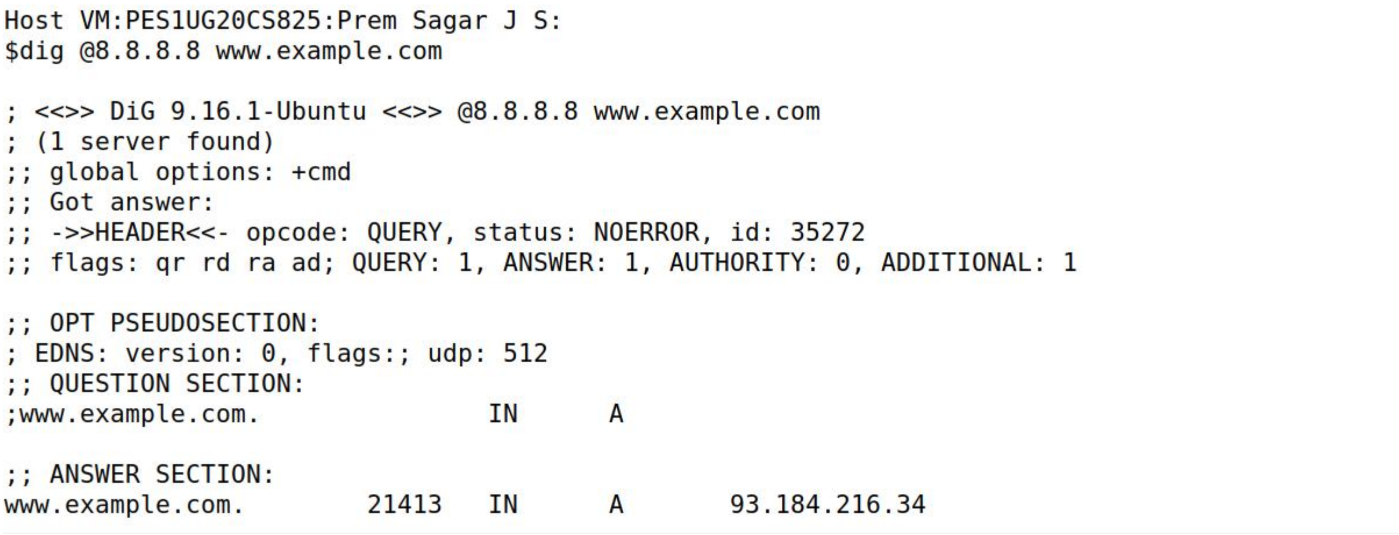
Command:
$ make
$ sudo insmod seedPrint.ko
$ lsmod | grep seedPrint



```
Host VM:PES1UG20CS825:Prem Sagar J S:
$make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Desktop/Labsetup/Labsetup/volumes/Codes/p
acket_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
^[[A^[[A^[[A^[[A  Building modules, stage 2.
  MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
Host VM:PES1UG20CS825:Prem Sagar J S:
$sudo insmod seedPrint.ko
Host VM:PES1UG20CS825:Prem Sagar J S:
$lsmod | grep seedPrint
seedPrint              16384  0
```

- After inserting executing the below commands to notice the difference.

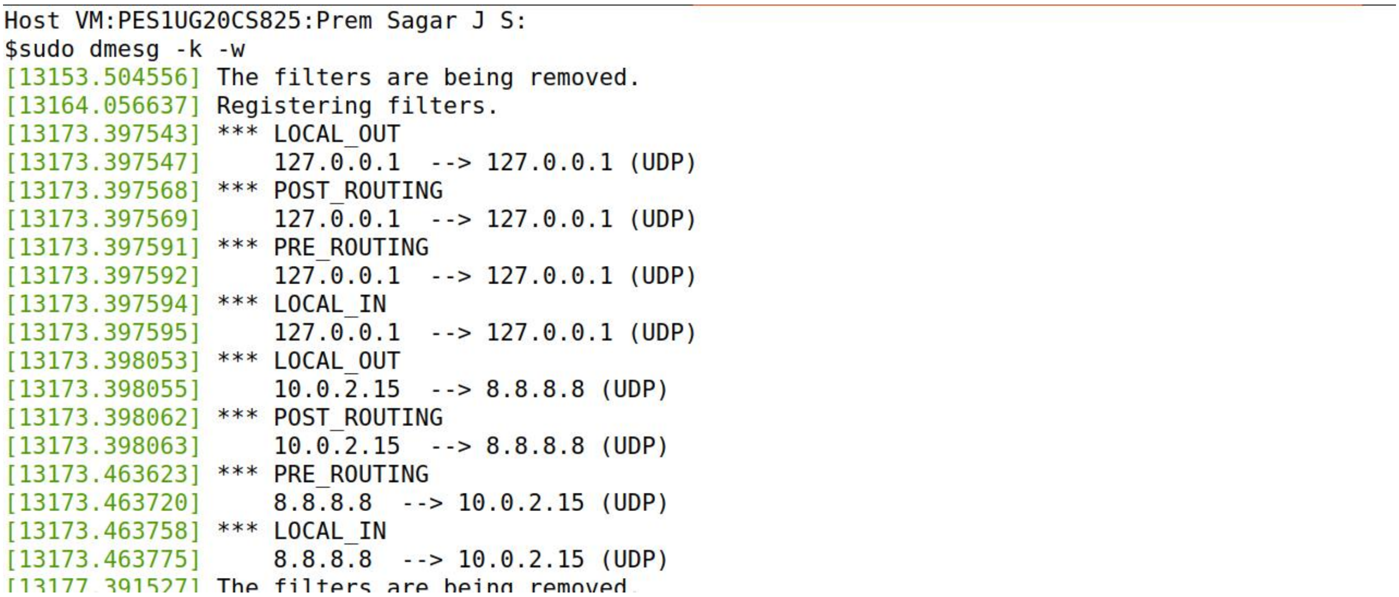
Command:
$ dig @8.8.8.8 www.example.com



```
Host VM:PES1UG20CS825:Prem Sagar J S:
$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: 35272
;; 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.        21413   IN      A       93.184.216.34
```

- On one window to view kernel messages.

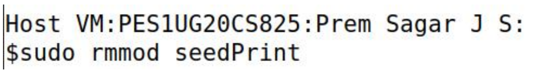
Command:
$ sudo dmesg -k -w



```
Host VM:PES1UG20CS825:Prem Sagar J S:
$sudo dmesg -k -w
[13153.504556] The filters are being removed.
[13164.056637] Registering filters.
[13173.397543] *** LOCAL_OUT
[13173.397547]     127.0.0.1  --> 127.0.0.1 (UDP)
[13173.397568] *** POST_ROUTING
[13173.397569]     127.0.0.1  --> 127.0.0.1 (UDP)
[13173.397591] *** PRE_ROUTING
[13173.397592]     127.0.0.1  --> 127.0.0.1 (UDP)
[13173.397594] *** LOCAL_IN
[13173.397595]     127.0.0.1  --> 127.0.0.1 (UDP)
[13173.398053] *** LOCAL_OUT
[13173.398055]     10.0.2.15  --> 8.8.8.8 (UDP)
[13173.398062] *** POST_ROUTING
[13173.398063]     10.0.2.15  --> 8.8.8.8 (UDP)
[13173.463623] *** PRE_ROUTING
[13173.463720]     8.8.8.8  --> 10.0.2.15 (UDP)
[13173.463758] *** LOCAL_IN
[13173.463775]     8.8.8.8  --> 10.0.2.15 (UDP)
[13177.391527] The filters are being removed.
```

- As you can see in the above screenshots I' m able access www.example.com using 8.8.8.8 and in the Kernel messeges I can see printed info of the all the netfilter hooks.

- Removing the module.



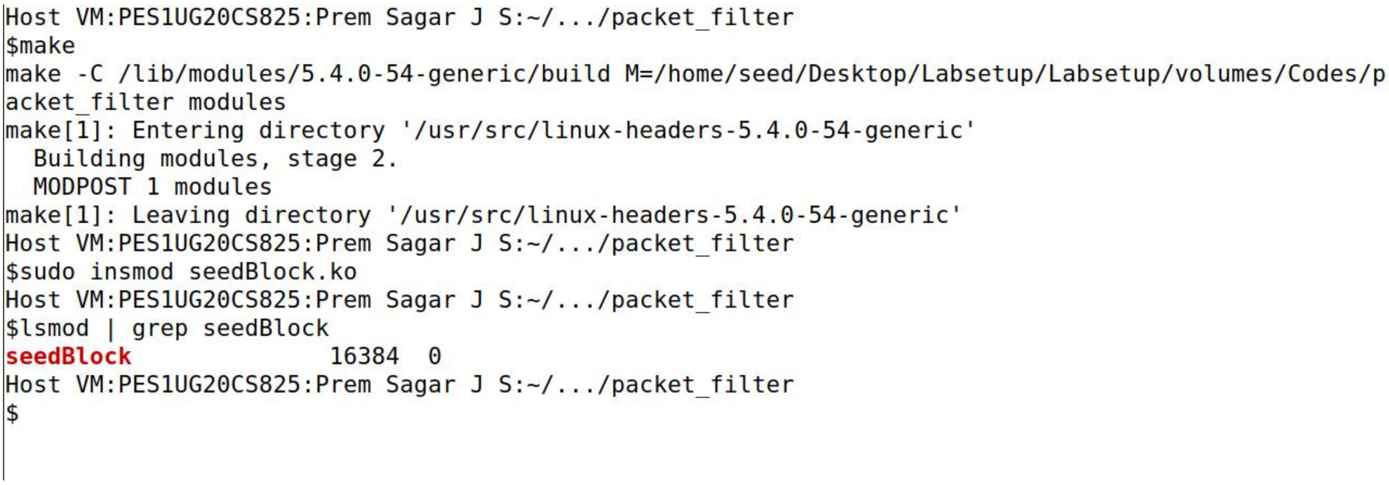
```
Host VM:PES1UG20CS825:Prem Sagar J S:
$sudo rmmod seedPrint
```

3. Implement two more hooks to achieve the following:

(1) preventing other computers to ping the VM, and 6
(2) preventing other computers from telnetting into the VM.

- On terminal window, insert the kernel module.

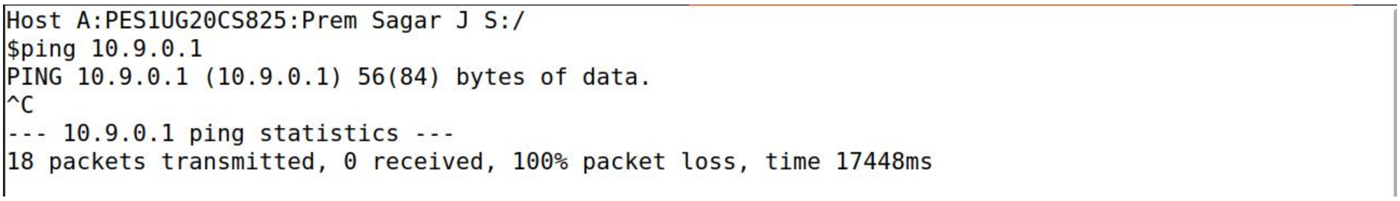
Command:
$ make
$ sudo insmod seedBlock.ko
$ lsmod | grep seedBlock



```
Host VM:PES1UG20CS825:Prem Sagar J S:~/.../packet_filter
$make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Desktop/Labsetup/Labsetup/volumes/Codes/p
acket_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'
Host VM:PES1UG20CS825:Prem Sagar J S:~/.../packet_filter
$sudo insmod seedBlock.ko
Host VM:PES1UG20CS825:Prem Sagar J S:~/.../packet_filter
$lsmod | grep seedBlock
seedBlock              16384  0
Host VM:PES1UG20CS825:Prem Sagar J S:~/.../packet_filter
$
```

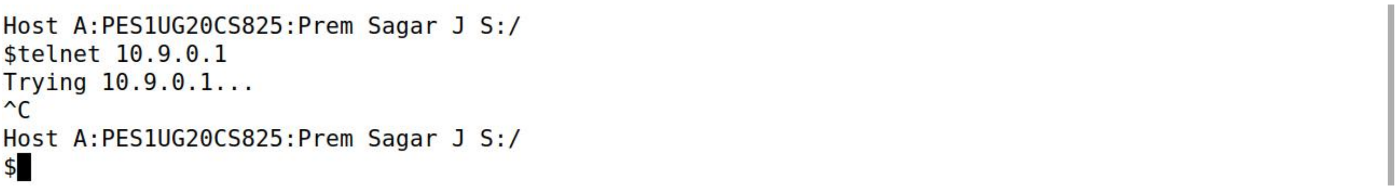
On the Host A - 10.9.0.5 terminal.

Command:
# ping 10.9.0.1



```
Host A:PES1UG20CS825:Prem Sagar J S:/
$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 ---
18 packets transmitted, 0 received, 100% packet loss, time 17448ms
```

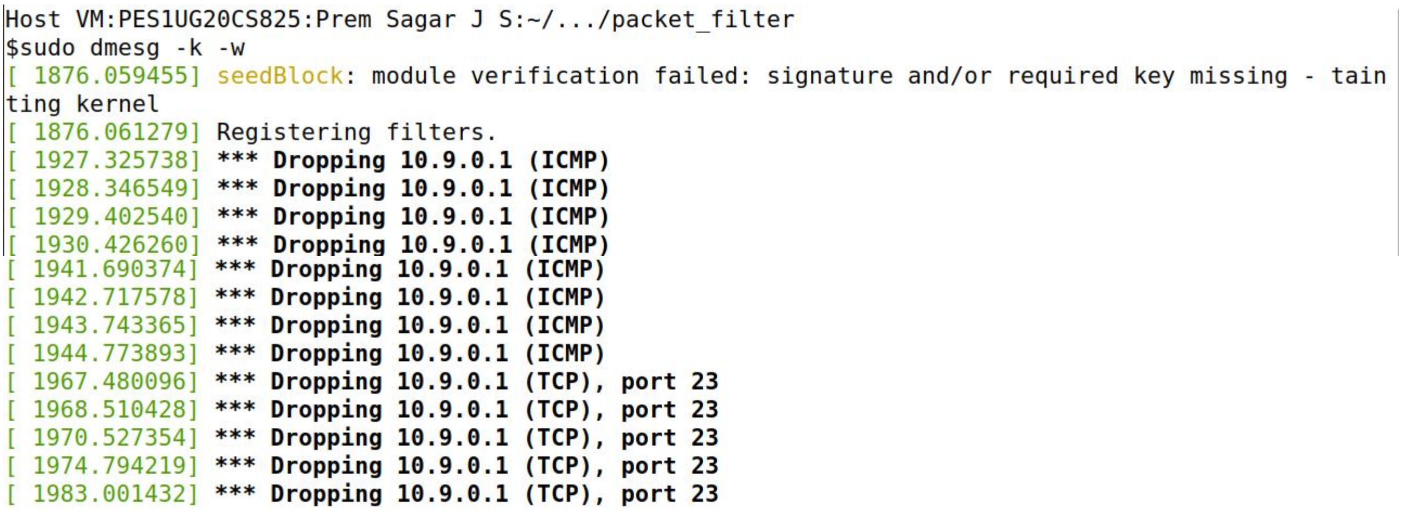
# telnet 10.9.0.1



```
Host A:PES1UG20CS825:Prem Sagar J S:/
$telnet 10.9.0.1
Trying 10.9.0.1...
^C
Host A:PES1UG20CS825:Prem Sagar J S:/
$
```

- On window to view kernel messages:

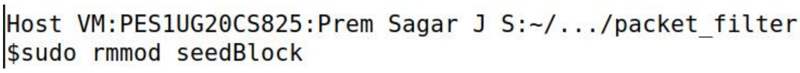
Command:
$ sudo dmesg -k -w



```
Host VM:PES1UG20CS825:Prem Sagar J S:~/.../packet_filter
$sudo dmesg -k -w
[ 1876.059455] seedBlock: module verification failed: signature and/or required key missing - tain
ting kernel
[ 1876.061279] Registering filters.
[ 1927.325738] *** Dropping 10.9.0.1 (ICMP)
[ 1928.346549] *** Dropping 10.9.0.1 (ICMP)
[ 1929.402540] *** Dropping 10.9.0.1 (ICMP)
[ 1930.426260] *** Dropping 10.9.0.1 (ICMP)
[ 1941.690374] *** Dropping 10.9.0.1 (ICMP)
[ 1942.717578] *** Dropping 10.9.0.1 (ICMP)
[ 1943.743365] *** Dropping 10.9.0.1 (ICMP)
[ 1944.773893] *** Dropping 10.9.0.1 (ICMP)
[ 1967.480096] *** Dropping 10.9.0.1 (TCP), port 23
[ 1968.510428] *** Dropping 10.9.0.1 (TCP), port 23
[ 1970.527354] *** Dropping 10.9.0.1 (TCP), port 23
[ 1974.794219] *** Dropping 10.9.0.1 (TCP), port 23
[ 1983.001432] *** Dropping 10.9.0.1 (TCP), port 23
```

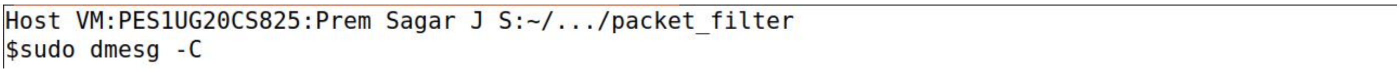
- After inserting the module im not able to ping the machine and not able establish the telnet connection to the machine, all the icmp packets and tcp packets on port 23 are being dropped.

- Removing the module.



```
Host VM:PES1UG20CS825:Prem Sagar J S:~/.../packet_filter
$sudo rmmod seedBlock
```

- clearing the kernel messages.



```
Host VM:PES1UG20CS825:Prem Sagar J S:~/.../packet_filter
$sudo dmesg -C
```

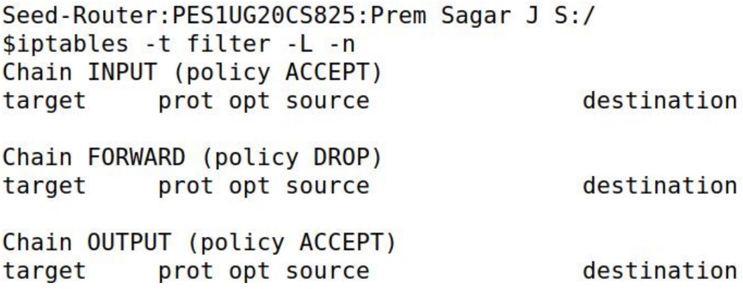
Task 2: Experimenting with Stateless Firewall Rules

Linux has a built-in firewall, also based on netfilter. This firewall is called iptables. Technically, the kernel part implementation of the firewall is called Xtables, while iptables is a user-space program to configure the firewall. However, iptables is often used to refer to both the kernel-part implementation and the user-space program.

Task 2.A: Protecting the Router

- Order to view the current policies running the below command on seed router.
-

Command:
# iptables -t filter -L -n

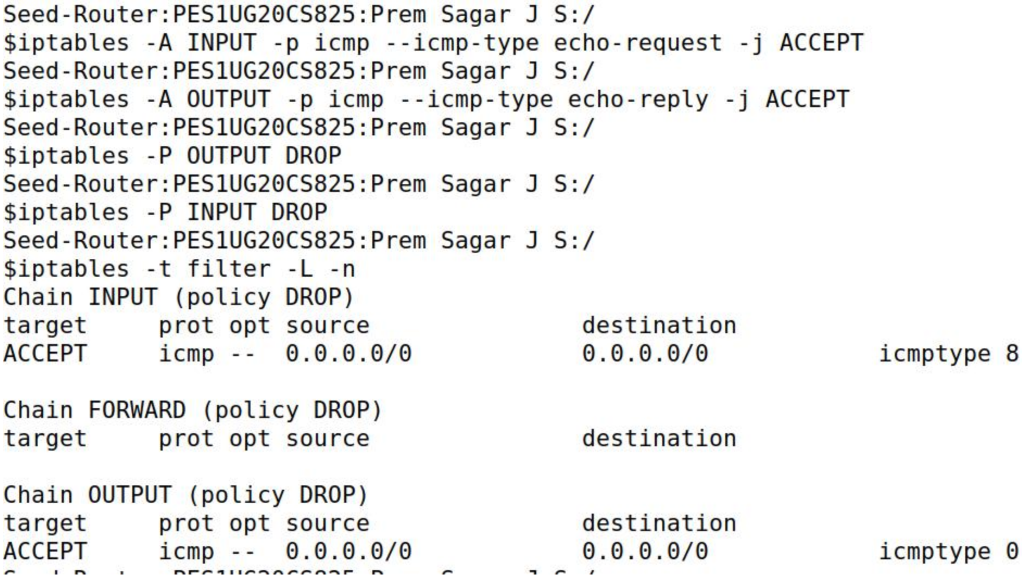


```
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -t filter -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- Adding the rules to the iptables on the seed router.

On seed-router run -
Command:

```
# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
# iptables -P OUTPUT DROP
# iptables -P INPUT DROP
# iptables -t filter -L -n
```

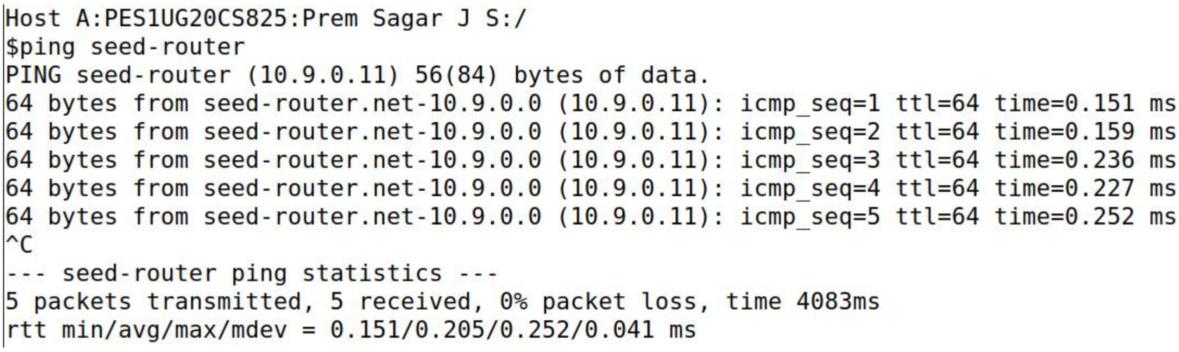


```
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -P OUTPUT DROP
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -P INPUT DROP
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$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 DROP)
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
```

- Now trying to access (ping and telnet) the router from Host A - 10.9.0.5

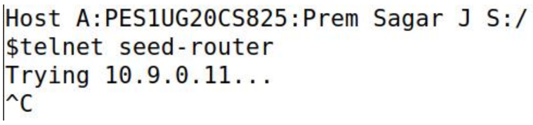
Command:

```
# ping seed-router
```



```
Host A:PES1UG20CS825:Prem Sagar J S:/
$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=0.151 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=2 ttl=64 time=0.159 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=3 ttl=64 time=0.236 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=4 ttl=64 time=0.227 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=5 ttl=64 time=0.252 ms
^C
--- seed-router ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4083ms
rtt min/avg/max/mdev = 0.151/0.205/0.252/0.041 ms
```

```
# telnet seed-router
```



```
Host A:PES1UG20CS825:Prem Sagar J S:/
$telnet seed-router
Trying 10.9.0.11...
^C
```

Questions :

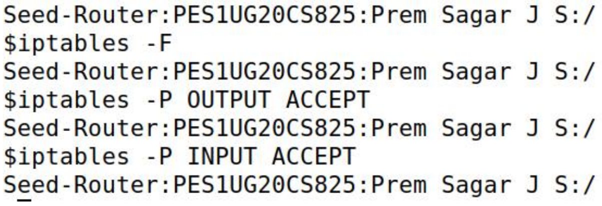
(1) Can you ping the router?

=> Yes, Im able to ping the router.

(2) Can you telnet into the router (a telnet server is running on all the containers; an account called seed was created on them with a password dees).

=> No, Im not able to telnet into the router as you can see the above screenshot.

Cleanup



```
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -F
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -P OUTPUT ACCEPT
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -P INPUT ACCEPT
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
```

Task 2.B: Protecting the Internal Network

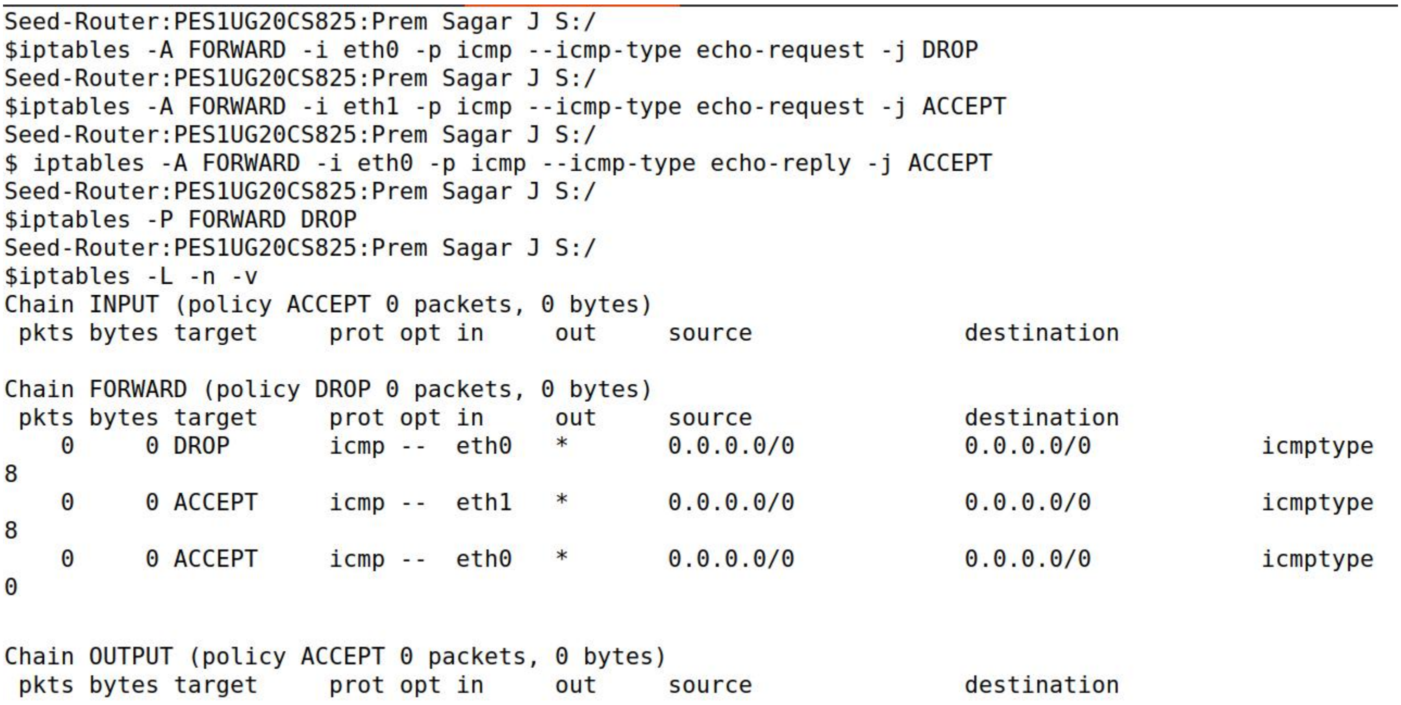
In this task, we want to implement a firewall to protect the internal network. we need to enforce the following restrictions on the ICMP traffic:

1. Outside hosts cannot ping internal hosts.
2. Outside hosts can ping the router.
3. Internal hosts can ping outside hosts.
4. All other packets between the internal and external networks should be blocked.

- Executing the following iptables commands on the **seed-router container.**

Commands:

```
# iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-request -j DROP
# iptables -A FORWARD -i eth1 -p icmp --icmp-type echo-request -j ACCEPT
# iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-reply -j ACCEPT
# iptables -P FORWARD DROP
# iptables -L -n -v
```

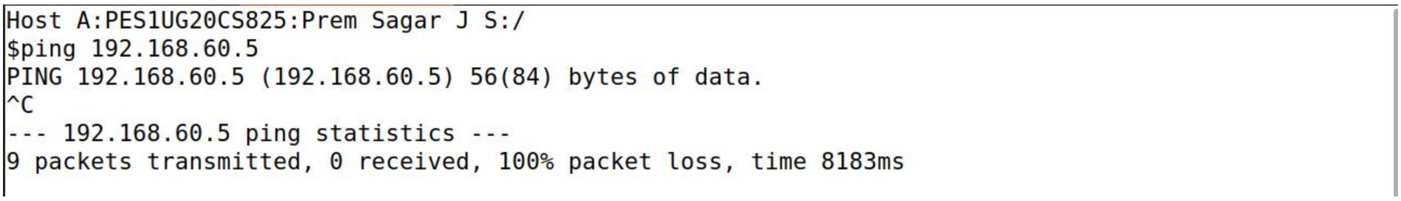


```
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-request -j DROP
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -A FORWARD -i eth1 -p icmp --icmp-type echo-request -j ACCEPT
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$ iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-reply -j ACCEPT
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -P FORWARD DROP
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$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            icmptype
8
    0     0 ACCEPT     icmp --  eth1   *       0.0.0.0/0            0.0.0.0/0            icmptype
8
    0     0 ACCEPT     icmp --  eth0   *       0.0.0.0/0            0.0.0.0/0            icmptype
0

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

- Now we will see if these **restrictions have been enforced** in the network.

1. Outside hosts cannot ping internal hosts.

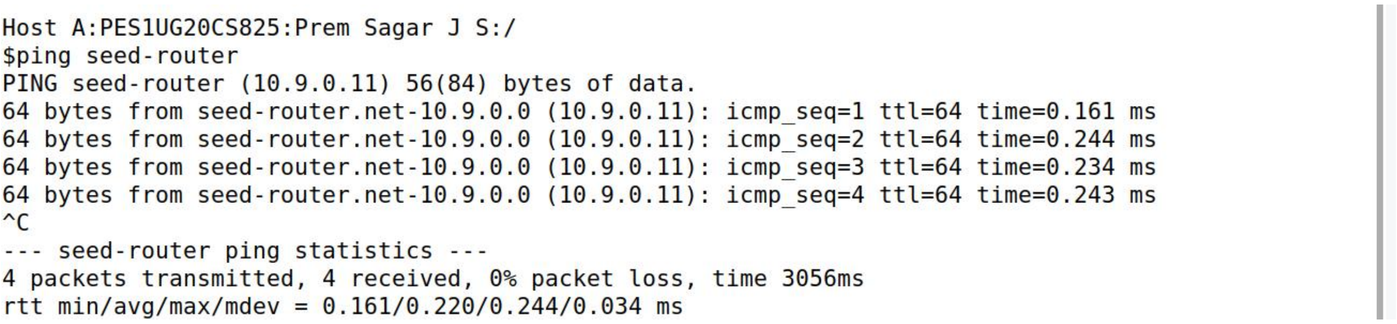
On Host A - 10.9.0.5 execute
Command:
# ping 192.168.60.5



```
Host A:PES1UG20CS825:Prem Sagar J S:/
$ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8183ms
```

2. Outside hosts can ping the router.

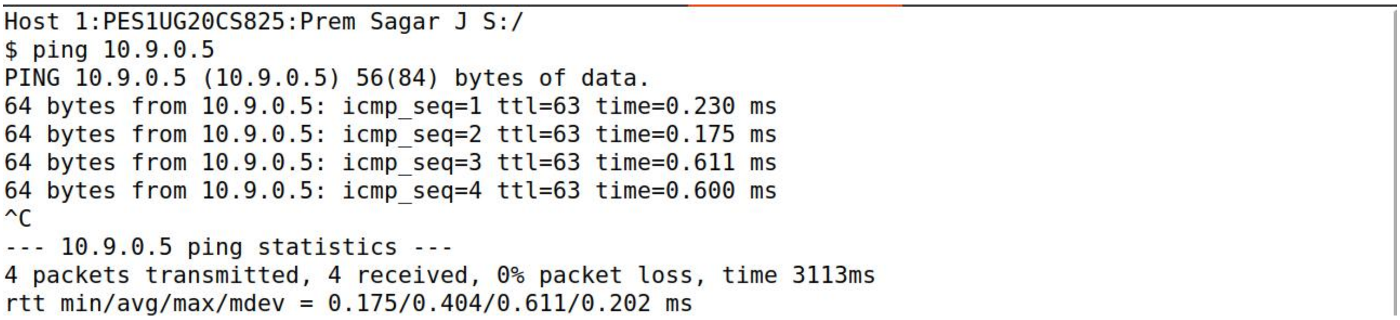
On Host A - 10.9.0.5 executing
Command:
# ping seed-router



```
Host A:PES1UG20CS825:Prem Sagar J S:/
$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=0.161 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=2 ttl=64 time=0.244 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=3 ttl=64 time=0.234 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=4 ttl=64 time=0.243 ms
^C
--- seed-router ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3056ms
rtt min/avg/max/mdev = 0.161/0.220/0.244/0.034 ms
```

3. Internal hosts can ping Outside Hosts.

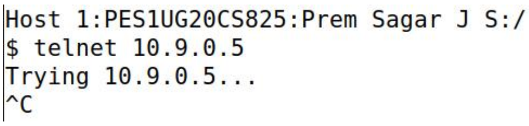
On host1-192.168.60.5 executing
Command:
# ping 10.9.0.5



```
Host 1:PES1UG20CS825:Prem Sagar J S:/
$ 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=0.230 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.175 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.611 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.600 ms
^C
--- 10.9.0.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3113ms
rtt min/avg/max/mdev = 0.175/0.404/0.611/0.202 ms
```

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

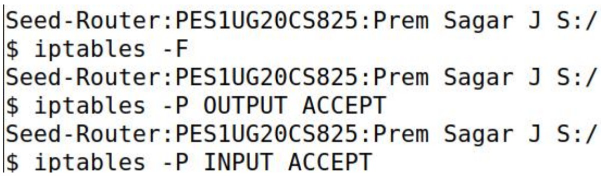
On host1-192.168.60.5 executing
Command:
# telnet 10.9.0.5



```
Host 1:PES1UG20CS825:Prem Sagar J S:/
$ telnet 10.9.0.5
Trying 10.9.0.5...
^C
```

- All the rules on our firewall are enforced and firewall is working fine.

Cleanup



```
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$ iptables -F
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$ iptables -P OUTPUT ACCEPT
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$ iptables -P INPUT ACCEPT
```

Task 2.C: Protecting Internal Servers

In this task, we want to protect the TCP servers inside the internal network (192.168.60.0/24).

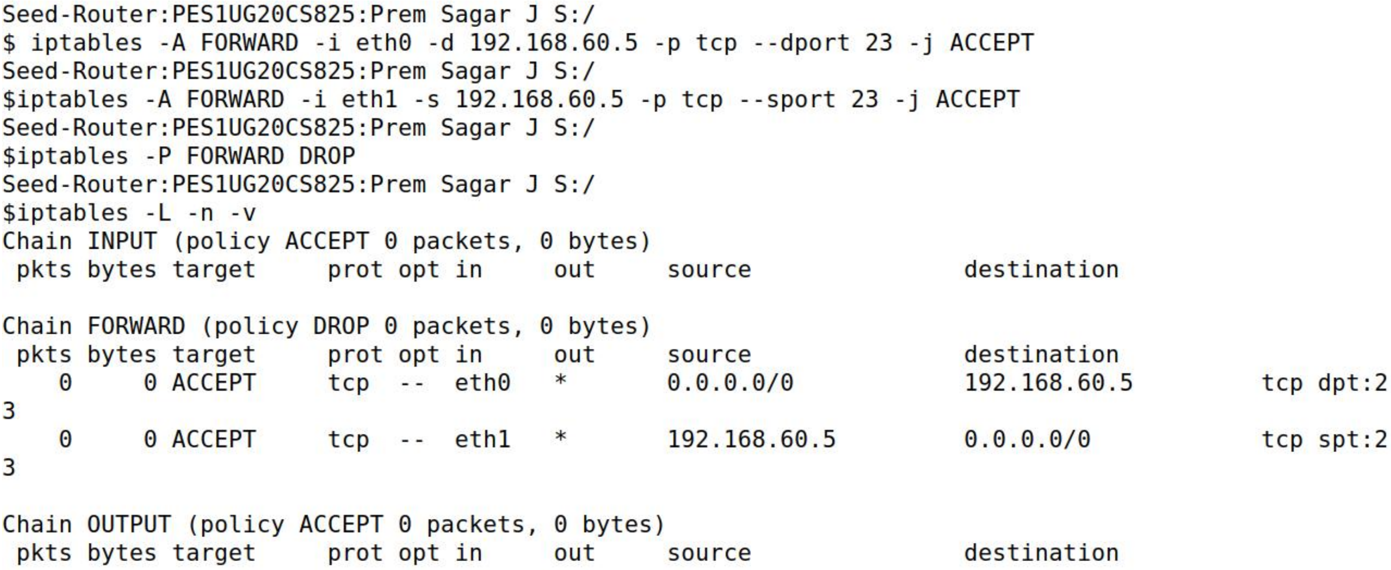
More specifically,

1. All the internal hosts run a telnet server (listening to port 23). Outside hosts can only access the telnet server on 192.168.60.5, not the other internal hosts.
2. Outside hosts cannot access other internal servers.
3. Internal hosts can access all the internal servers.
4. Internal hosts cannot access external servers.

- Executing the following iptables commands on the **seed-router container.**

Commands:

```
# iptables -A FORWARD -i eth0 -d 192.168.60.5 -p tcp --dport 23 -j ACCEPT
# iptables -A FORWARD -i eth1 -s 192.168.60.5 -p tcp --sport 23 -j ACCEPT
# iptables -P FORWARD DROP
# iptables -L -n -v
```

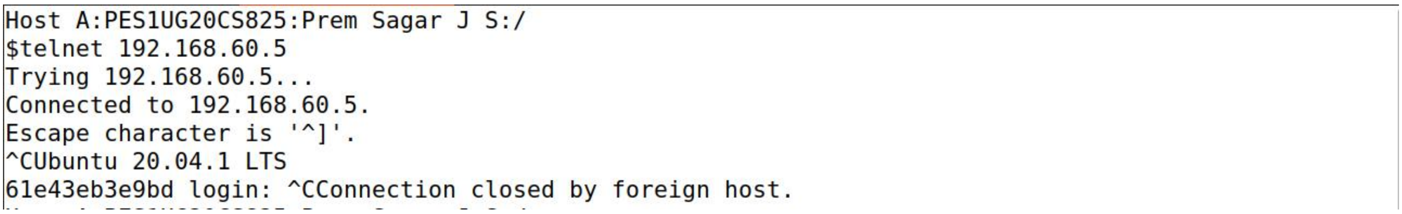


```
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$ iptables -A FORWARD -i eth0 -d 192.168.60.5 -p tcp --dport 23 -j ACCEPT
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -A FORWARD -i eth1 -s 192.168.60.5 -p tcp --sport 23 -j ACCEPT
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$iptables -P FORWARD DROP
Seed-Router:PES1UG20CS825:Prem Sagar J S:/
$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         tcp dpt:2
3
    0     0 ACCEPT     tcp  --  eth1   *       192.168.60.5         0.0.0.0/0            tcp spt:2
3

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

- After adding all the rules to the iptables firewall, Now we will see if these **restrictions have been enforced** in the network.

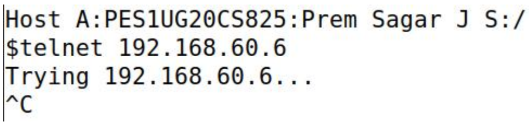
1. All the internal hosts run a telnet server (listening to port 23). Outside hosts can only access the telnet server on 192.168.60.5, not the other internal hosts.

On host A - 10.9.0.5
Command:
# telnet 192.168.60.5



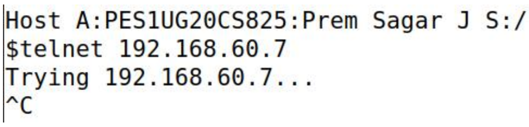
```
Host A:PES1UG20CS825:Prem Sagar J S:/
$telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
^CUbuntu 20.04.1 LTS
61e43eb3e9bd login: ^CConnection closed by foreign host.
```

2. Outside hosts cannot access other internal servers
On host A - 10.9.0.5
Command:
# telnet 192.168.60.6



```
Host A:PES1UG20CS825:Prem Sagar J S:/
$telnet 192.168.60.6
Trying 192.168.60.6...
^C
```

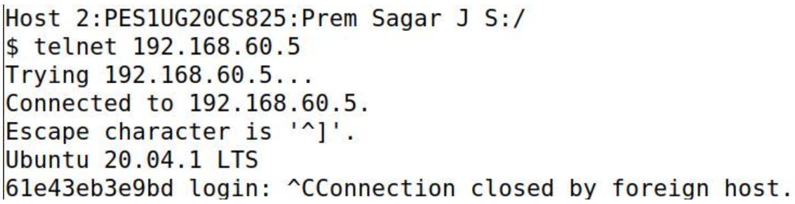
# telnet 192.168.60.7



```
Host A:PES1UG20CS825:Prem Sagar J S:/
$telnet 192.168.60.7
Trying 192.168.60.7...
^C
```

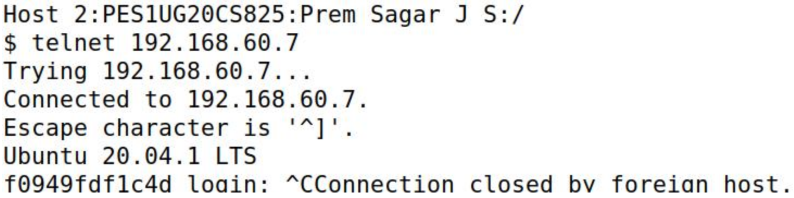
4. Internal hosts can access all the internal servers.

On host2 - 192.168.60.6
Command:
# telnet 192.168.60.5



```
Host 2:PES1UG20CS825:Prem Sagar J S:/
$ telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
61e43eb3e9bd login: ^CConnection closed by foreign host.
```

# telnet 192.168.60.7



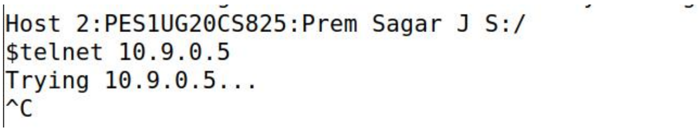
```
Host 2:PES1UG20CS825:Prem Sagar J S:/
$ telnet 192.168.60.7
Trying 192.168.60.7...
Connected to 192.168.60.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f0949fdf1c4d login: ^CConnection closed by foreign host.
```

4. Internal hosts cannot access external servers.

On host2 - 192.168.60.6

Command:

# telnet 10.9.0.5



```
Host 2:PES1UG20CS825:Prem Sagar J S:/
$telnet 10.9.0.5
Trying 10.9.0.5...
^C
```

- As you can see in the above screenshots that we were able to protect the TCP servers inside the internal network (192.168.60.0/24).
- All the rules were enforced perfectly and tested.
- Clean up has been performed before executing the Task 3.

Task 3: Connection Tracking and Stateful Firewall

Task 3.A: Experiment with the Connection Tracking.

- To support stateful firewalls, we need to be able to track connections. This is achieved by the conntrack mechanism inside the kernel.

- This can be done using the following command **conntrack -L**.

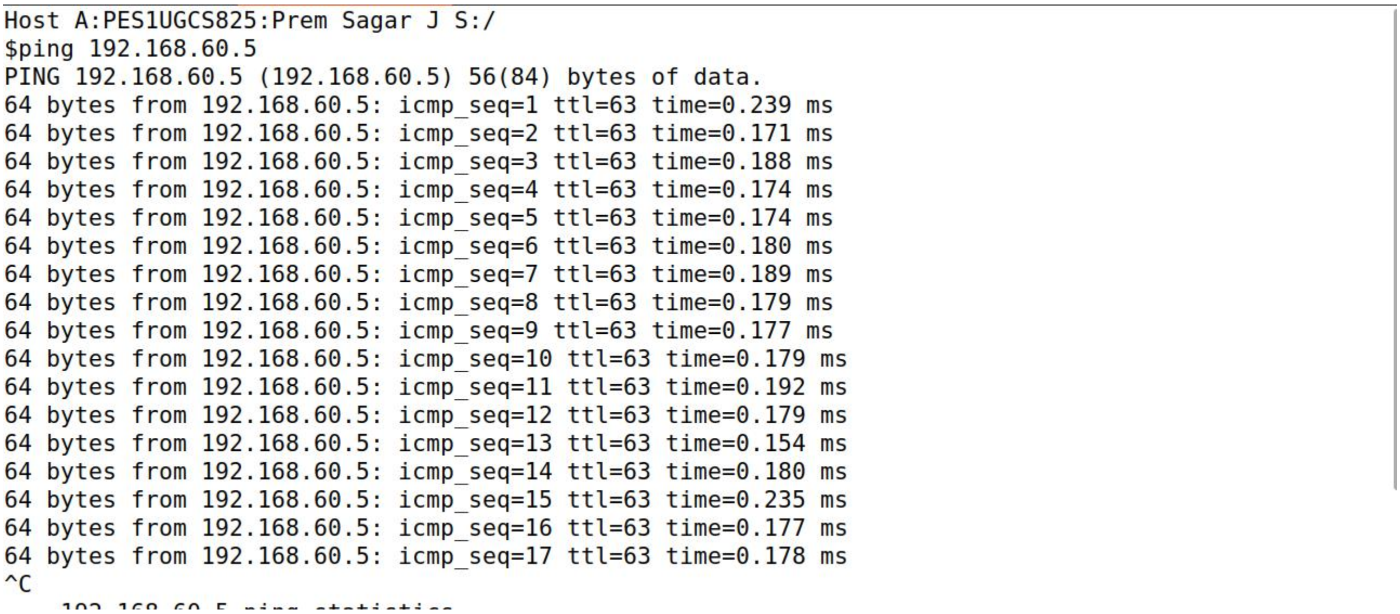
ICMP experiment:

- Running command and checking the connection tracking information on the router.

On host A - 10.9.0.5

Command:

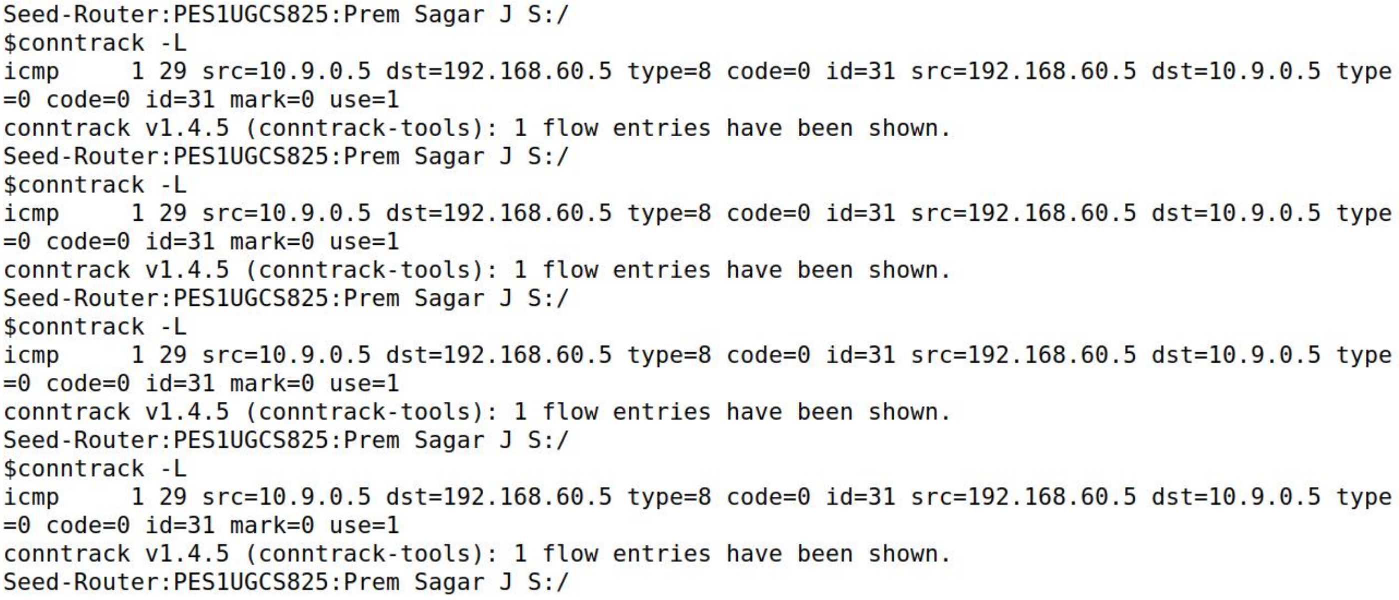
# ping 192.168.60.5



```
Host A:PES1UGCS825:Prem Sagar J S:/
$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=0.239 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.171 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.188 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.174 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.174 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.180 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.189 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.179 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.177 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.179 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.192 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.179 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.154 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.180 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.235 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.177 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.178 ms
^C
    192.168.60.5 ping statistics
```

- Immediately moving to the seed router and running.

# conntrack -L



```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$conntrack -L
icmp     1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=31 src=192.168.60.5 dst=10.9.0.5 type
=0 code=0 id=31 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$conntrack -L
icmp     1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=31 src=192.168.60.5 dst=10.9.0.5 type
=0 code=0 id=31 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$conntrack -L
icmp     1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=31 src=192.168.60.5 dst=10.9.0.5 type
=0 code=0 id=31 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$conntrack -L
icmp     1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=31 src=192.168.60.5 dst=10.9.0.5 type
=0 code=0 id=31 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
Seed-Router:PES1UGCS825:Prem Sagar J S:/
```

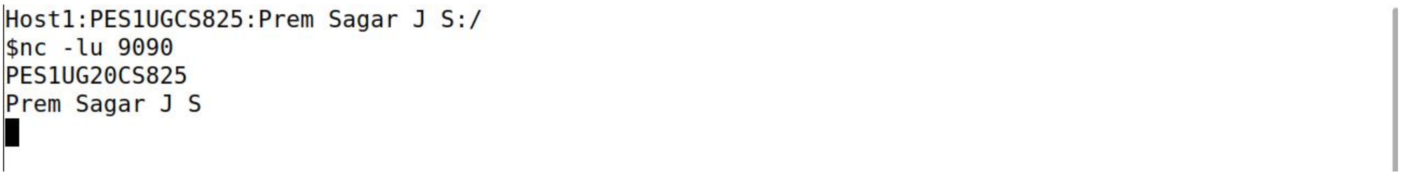
UDP experiment:

- Running the command and check the connection tracking information on the router.

On host 1 - 192.168.60.5

Command:

# nc -lu 9090

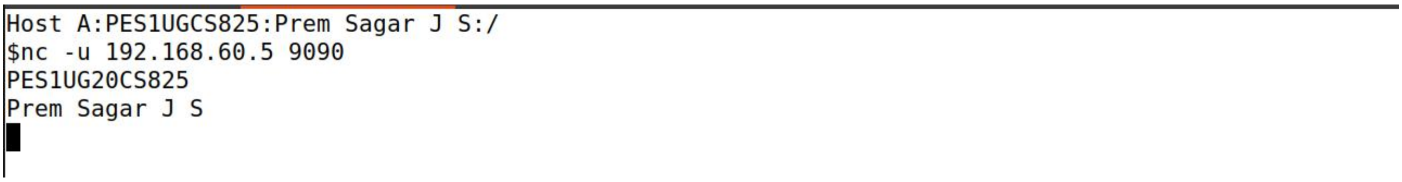


```
Host1:PES1UGCS825:Prem Sagar J S:/
$nc -lu 9090
PES1UG20CS825
Prem Sagar J S
█
```

On host A - 10.9.0.5

Command:

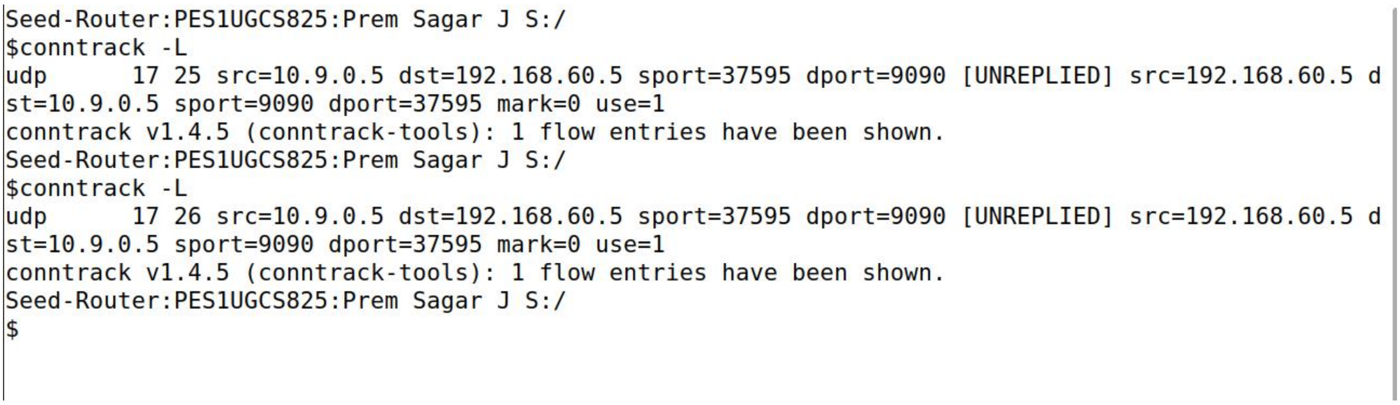
# nc -u 192.168.60.5 9090



```
Host A:PES1UGCS825:Prem Sagar J S:/
$nc -u 192.168.60.5 9090
PES1UG20CS825
Prem Sagar J S
█
```

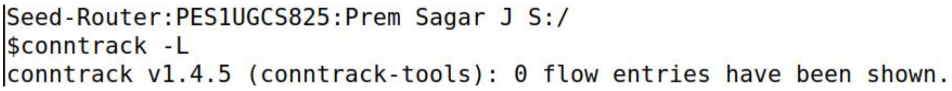
On seed router

# conntrack -L



```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$conntrack -L
udp      17 25 src=10.9.0.5 dst=192.168.60.5 sport=37595 dport=9090 [UNREPLIED] src=192.168.60.5 d
st=10.9.0.5 sport=9090 dport=37595 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$conntrack -L
udp      17 26 src=10.9.0.5 dst=192.168.60.5 sport=37595 dport=9090 [UNREPLIED] src=192.168.60.5 d
st=10.9.0.5 sport=9090 dport=37595 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$
```

- Closing the UDP connection and executing conntrack -L on the router container.
- Closing closing the UDP Connection there were no flow entries when the **conntrack -L** command was executed.



```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$conntrack -L
conntrack v1.4.5 (conntrack-tools): 0 flow entries have been shown.
```

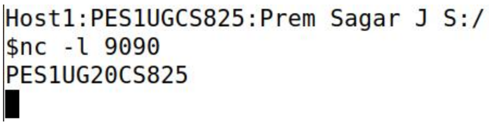
TCP experiment:

- Running the command and check the connection tracking information on the router.

On host 1 - 192.168.60.5

Command:

# nc -l 9090

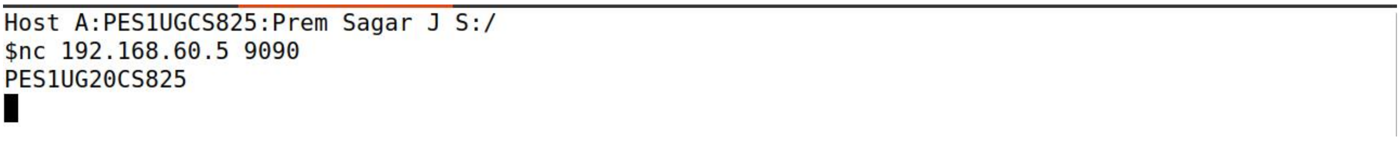


```
Host1:PES1UGCS825:Prem Sagar J S:/
$nc -l 9090
PES1UG20CS825
█
```

On host A - 10.9.0.5

Command:

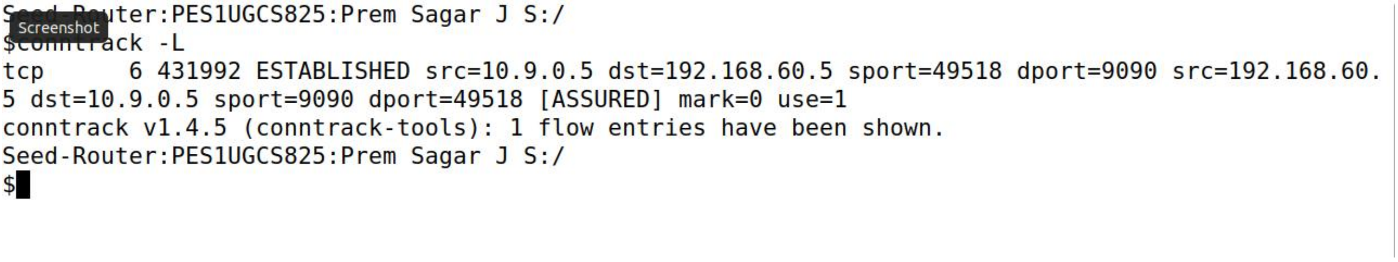
# nc 192.168.60.5 9090



```
Host A:PES1UGCS825:Prem Sagar J S:/
$nc 192.168.60.5 9090
PES1UG20CS825
█
```

On seed router

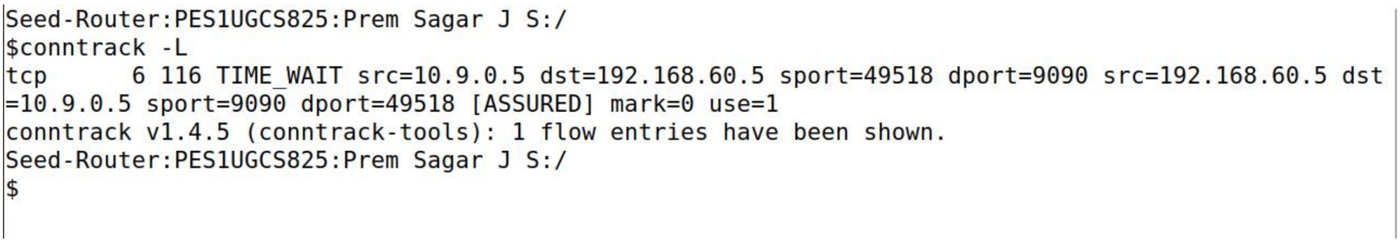
# conntrack -L



```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$conntrack -L
tcp      6 431992 ESTABLISHED src=10.9.0.5 dst=192.168.60.5 sport=49518 dport=9090 src=192.168.60.
5 dst=10.9.0.5 sport=9090 dport=49518 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$█
```

- As you can see in the above screenshot the TCP connection has been established from respective source and destination IP’ s and Ports.
- And there was only one flow entry in the Output.

- Closing the TCP connection and executing conntrack -L on the router container.
- Closing the TCP connection conntrack -L has 1 flow entry the label is changed from Established to Time_Wait.



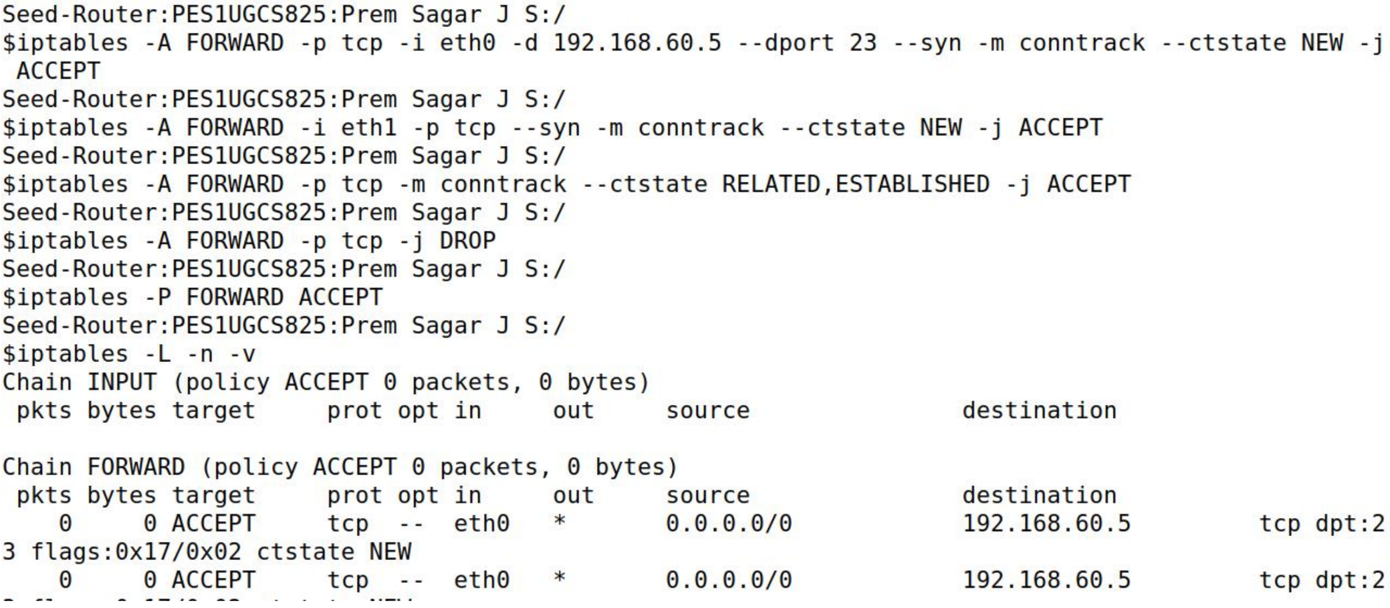
```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$conntrack -L
tcp      6 116 TIME_WAIT src=10.9.0.5 dst=192.168.60.5 sport=49518 dport=9090 src=192.168.60.5 dst
=10.9.0.5 sport=9090 dport=49518 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$
```

Task 3.B: Setting Up a Stateful Firewall

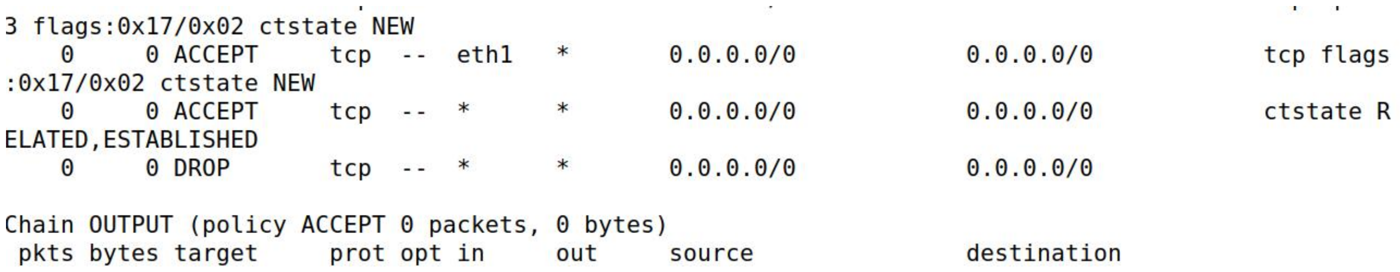
- For this task we have to rewrite the firewall rules in Task 2.C, but this time, we will add a rule allowing internal hosts to visit any external server (this was not allowed in Task 2.C).

On seed-router execute
Commands:

```
# iptables -A FORWARD -p tcp -i eth0 -d 192.168.60.5 --dport 23 --syn -m conntrack
--ctstate NEW -j ACCEPT
# iptables -A FORWARD -i eth1 -p tcp --syn -m conntrack --ctstate NEW -j
ACCEPT
# iptables -A FORWARD -p tcp -m conntrack --ctstate RELATED,ESTABLISHED
-j ACCEPT
# iptables -A FORWARD -p tcp -j DROP
# iptables -P FORWARD ACCEPT
# iptables -L -n -v
```



```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -A FORWARD -p tcp -i eth0 -d 192.168.60.5 --dport 23 --syn -m conntrack --ctstate NEW -j
 ACCEPT
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -A FORWARD -i eth1 -p tcp --syn -m conntrack --ctstate NEW -j ACCEPT
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -A FORWARD -p tcp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -A FORWARD -p tcp -j DROP
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -P FORWARD ACCEPT
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$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
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            192.168.60.5         tcp dpt:2
3 flags:0x17/0x02 ctstate NEW
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            192.168.60.5         tcp dpt:2
```



```
3 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 R
ELATED,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
```

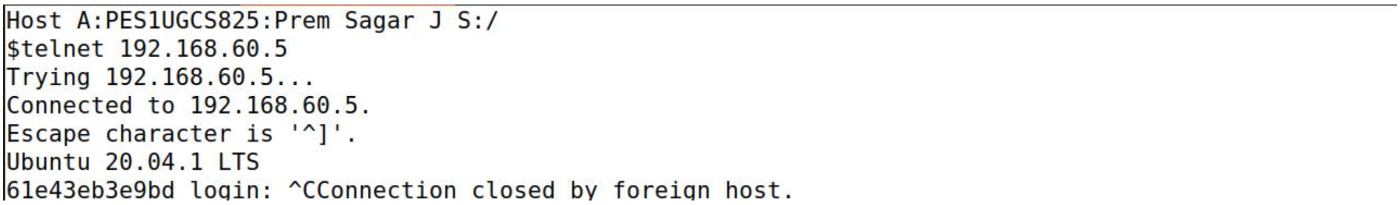
- Now we will see if these **restrictions have been enforced** in the network.

1. All the internal hosts run a telnet server (listening to port 23). Outside hosts can only access the telnet server on 192.168.60.5, not the other internal hosts.

On host A - 10.9.0.5

Command:

# telnet 192.168.60.5



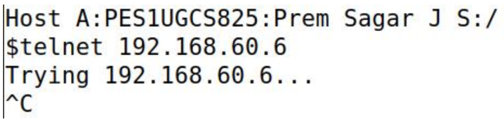
```
Host A:PES1UGCS825:Prem Sagar J S:/
$telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
61e43eb3e9bd login: ^CConnection closed by foreign host.
```

2. Outside hosts cannot access other internal servers.

On host A - 10.9.0.5

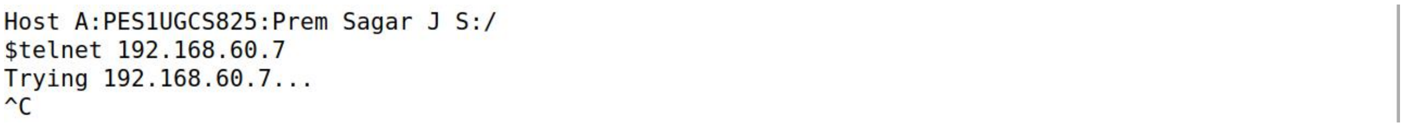
Command:

# telnet 192.168.60.6



```
Host A:PES1UGCS825:Prem Sagar J S:/
$telnet 192.168.60.6
Trying 192.168.60.6...
^C
```

# telnet 192.168.60.7



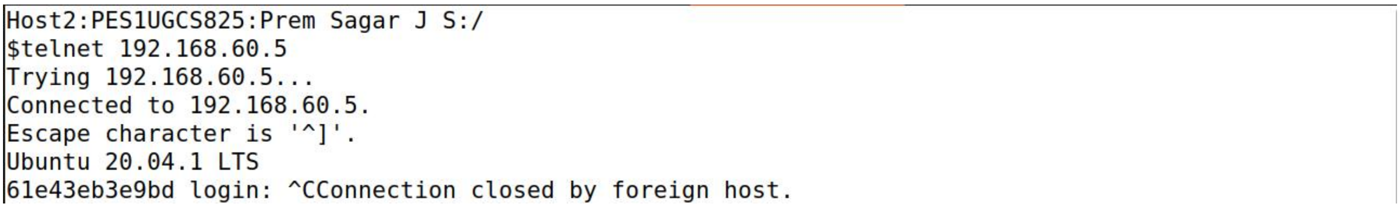
```
Host A:PES1UGCS825:Prem Sagar J S:/
$telnet 192.168.60.7
Trying 192.168.60.7...
^C
```

3. Internal hosts can access all the internal servers.

On host2 - 192.168.60.6

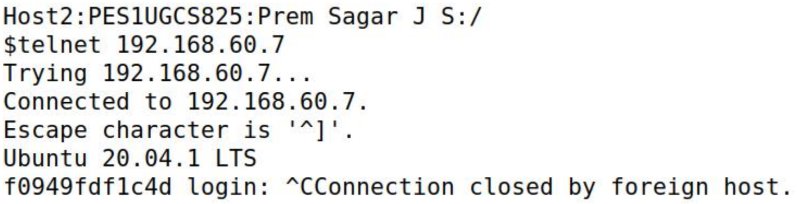
Command:

# telnet 192.168.60.5



```
Host2:PES1UGCS825:Prem Sagar J S:/
$telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
61e43eb3e9bd login: ^CConnection closed by foreign host.
```

# telnet 192.168.60.7



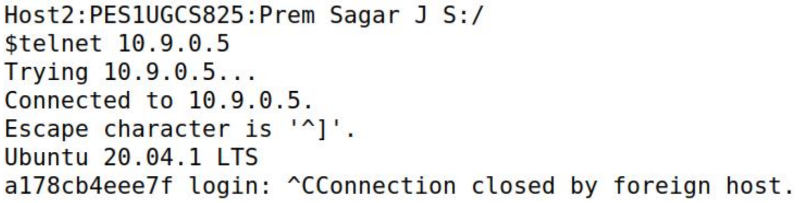
```
Host2:PES1UGCS825:Prem Sagar J S:/
$telnet 192.168.60.7
Trying 192.168.60.7...
Connected to 192.168.60.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f0949fdf1c4d login: ^CConnection closed by foreign host.
```

4. Internal hosts can access external servers.

On host2 - 192.168.60.6

Command:

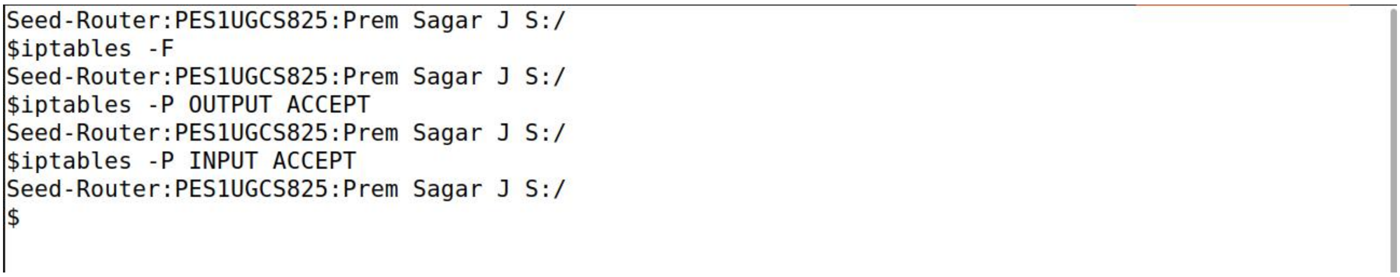
# telnet 10.9.0.5



```
Host2:PES1UGCS825:Prem Sagar J S:/
$telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
a178cb4eee7f login: ^CConnection closed by foreign host.
```

- All the Rules that are being added are enforced and working fine.
- Tested with respective hosts to ensure that our firewall is working are working.

Cleanup



```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -F
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -P OUTPUT ACCEPT
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -P INPUT ACCEPT
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$
```

Task 4: Limiting Network Traffic

In addition to blocking packets, we can also limit the number of packets that can pass through the
firewall. This can be done using the limit module of iptables.

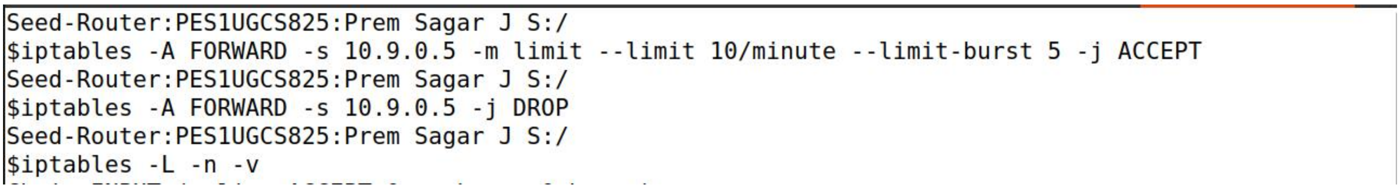
On seed router execute -

Command:

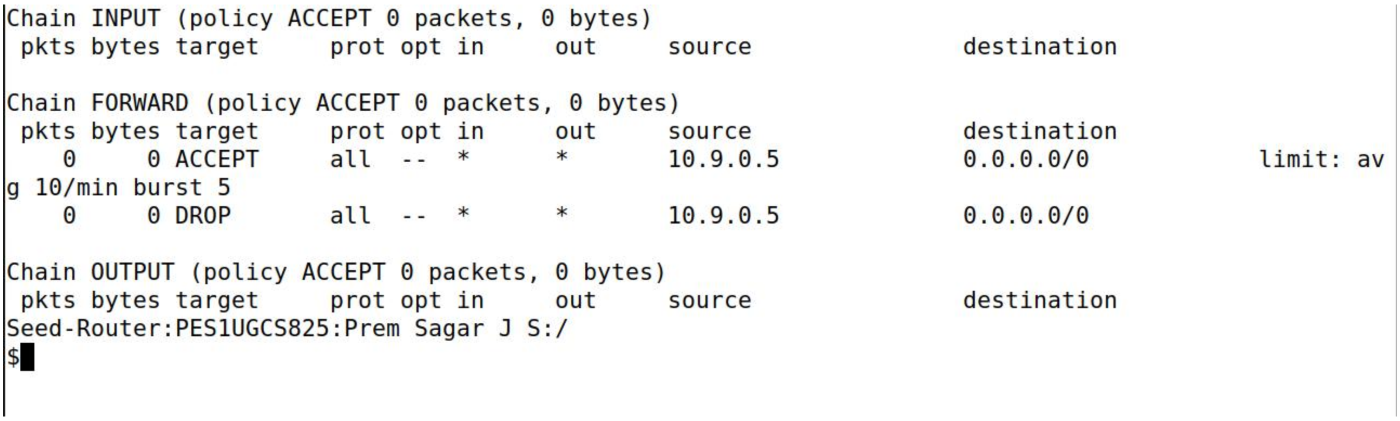
# iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -j ACCEPT

# iptables -A FORWARD -s 10.9.0.5 -j DROP

# iptables -L -n -v



```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -j ACCEPT
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -A FORWARD -s 10.9.0.5 -j DROP
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -L -n -v
```

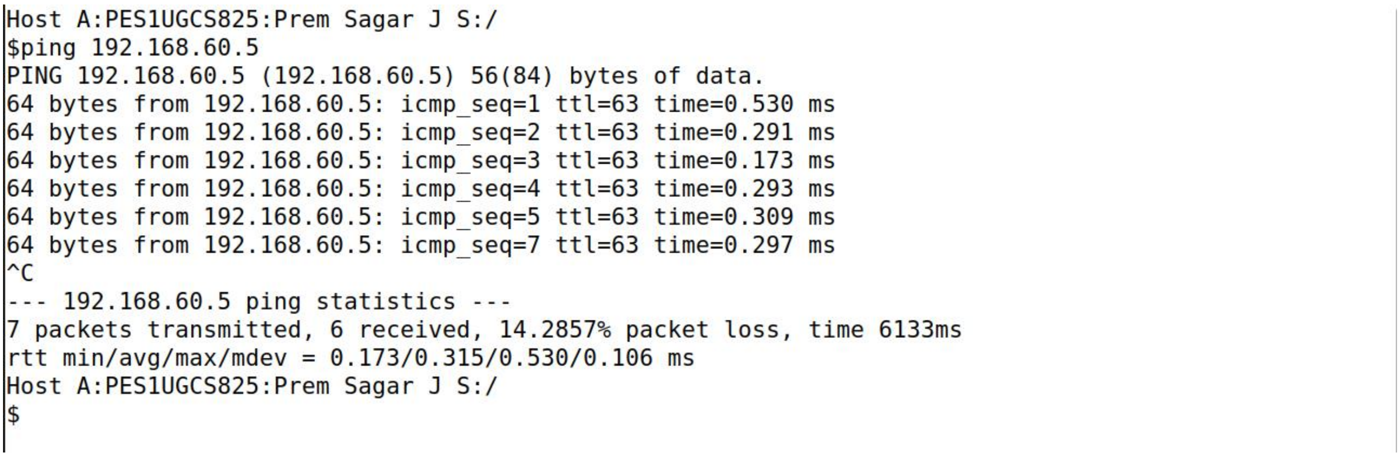


```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       10.9.0.5             0.0.0.0/0            limit: av
g 10/min burst 5
    0     0 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
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$
```

On host A - 10.9.0.5

Command:

# ping 192.168.60.5



```
Host A:PES1UGCS825:Prem Sagar J S:/
$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=0.530 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.291 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.173 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.293 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.309 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.297 ms
^C
--- 192.168.60.5 ping statistics ---
7 packets transmitted, 6 received, 14.2857% packet loss, time 6133ms
rtt min/avg/max/mdev = 0.173/0.315/0.530/0.106 ms
Host A:PES1UGCS825:Prem Sagar J S:/
$
```

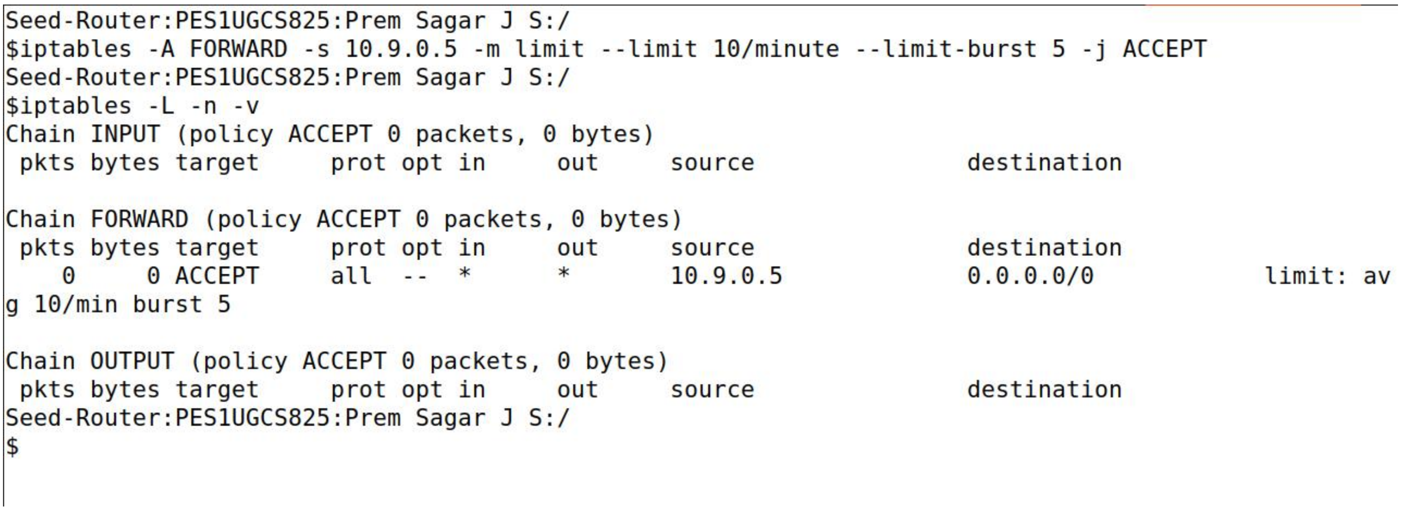
Cleaned the rules and now performing the same task without the second rule -

On seed router execute -

Command:

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

# iptables -L -n -v

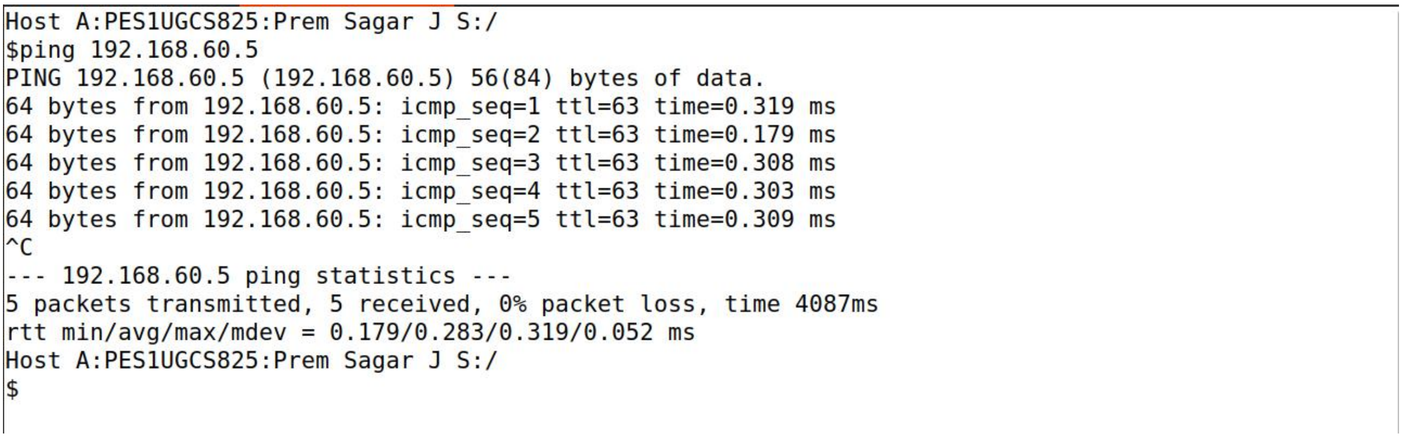


```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -j ACCEPT
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       10.9.0.5             0.0.0.0/0            limit: av
g 10/min burst 5

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$
```

On host A - 10.9.0.5

Command:

```
# ping 192.168.60.5
```



```
Host A:PES1UGCS825:Prem Sagar J S:/
$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=0.319 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.179 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.308 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.303 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.309 ms
^C
--- 192.168.60.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4087ms
rtt min/avg/max/mdev = 0.179/0.283/0.319/0.052 ms
Host A:PES1UGCS825:Prem Sagar J S:/
$
```

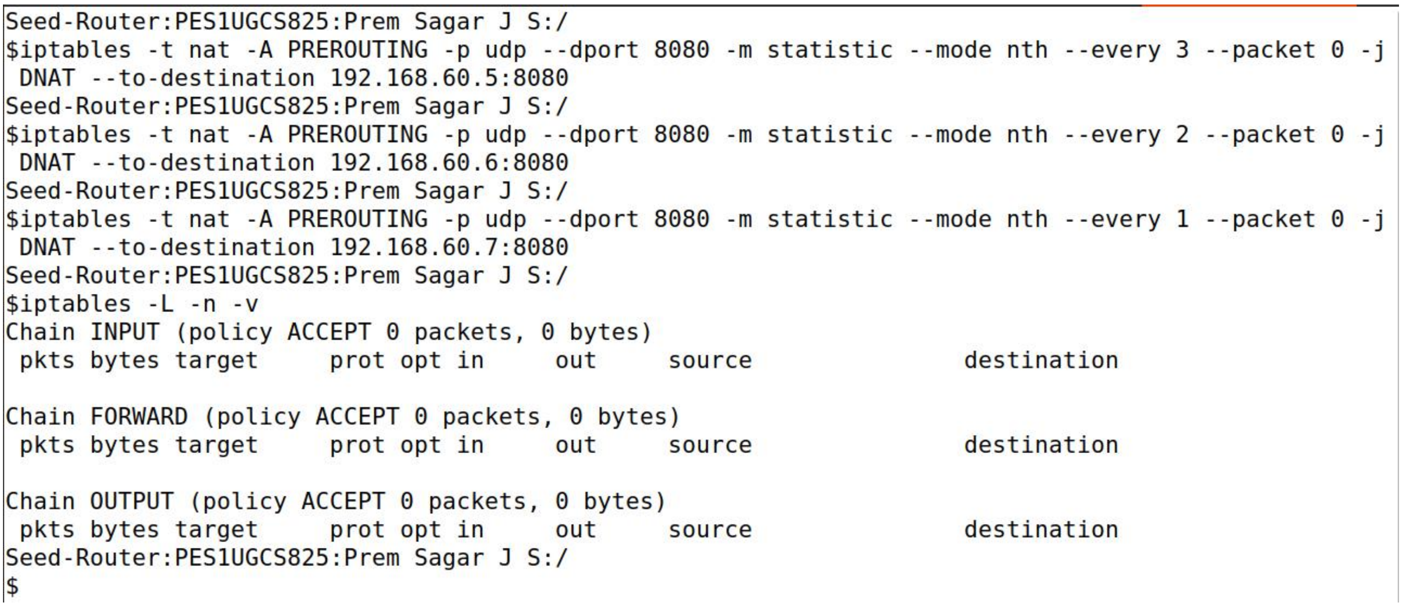
- After executing or adding both the rules there is no need to include the second rule, even without the second rule we are able perform the task

Task 5: Load Balancing

On the seed-router container:

Command:

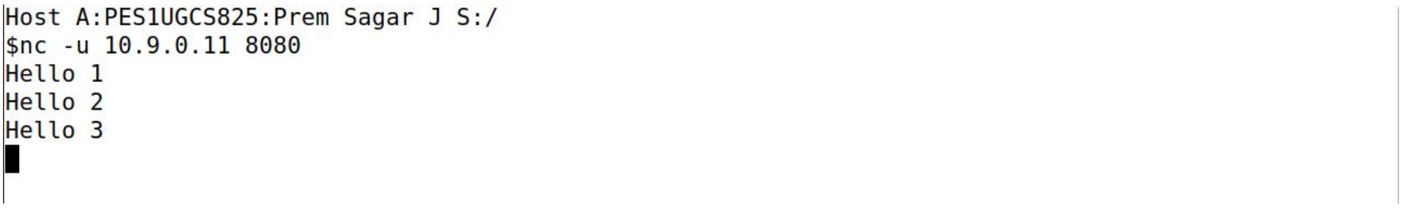
```
# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth
--every 3 --packet 0 -j DNAT --to-destination 192.168.60.5:8080
# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth
--every 2 --packet 0 -j DNAT --to-destination 192.168.60.6:8080
# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth
--every 1 --packet 0 -j DNAT --to-destination 192.168.60.7:8080
# iptables -L -n -v
```



```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$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
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$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
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$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
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$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
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$
```

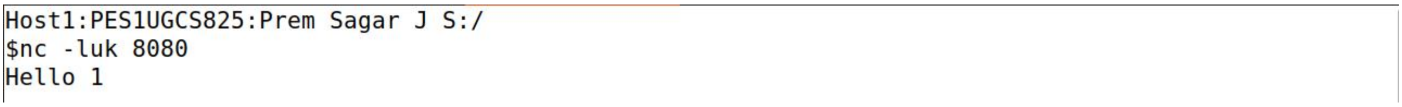
- On host1 - 192.168.60.5, host2 - 192.168.60.6 and host3 - 192.168.60.7 started the server using Command `nc -luk 8080`

On host A - 10.9.0.5
Command:
# nc -u 10.9.0.11 8080



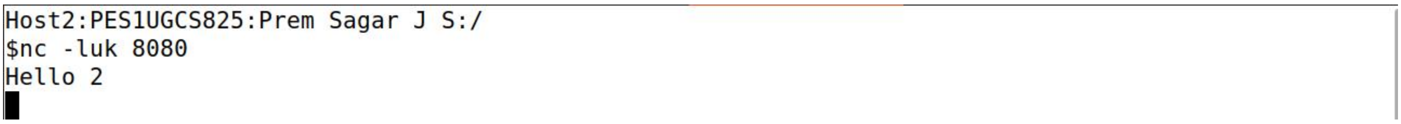
```
Host A:PES1UGCS825:Prem Sagar J S:/
$nc -u 10.9.0.11 8080
Hello 1
Hello 2
Hello 3
```

On host1 - 192.168.60.5



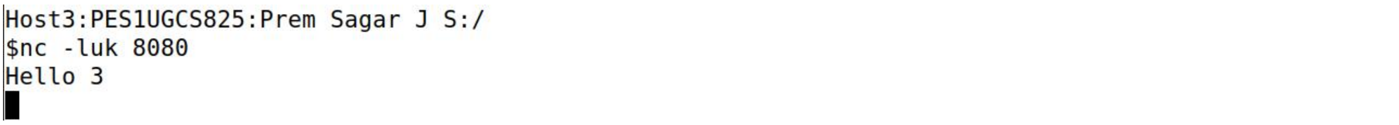
```
Host1:PES1UGCS825:Prem Sagar J S:/
$nc -luk 8080
Hello 1
```

On host2 - 192.168.60.6



```
Host2:PES1UGCS825:Prem Sagar J S:/
$nc -luk 8080
Hello 2
```

On host3 - 192.168.60.7



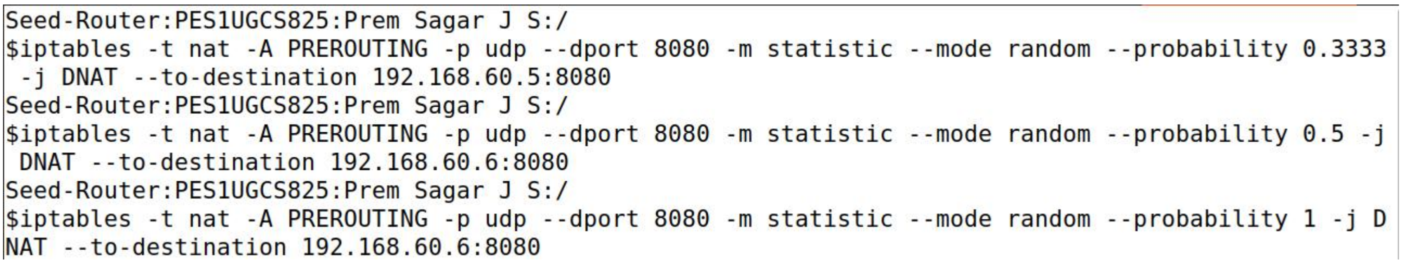
```
Host3:PES1UGCS825:Prem Sagar J S:/
$nc -luk 8080
Hello 3
```

- ‘Hello 1’ appears in the host 1 terminal, ‘Hello 2’ appears in the host 2 terminal etc.

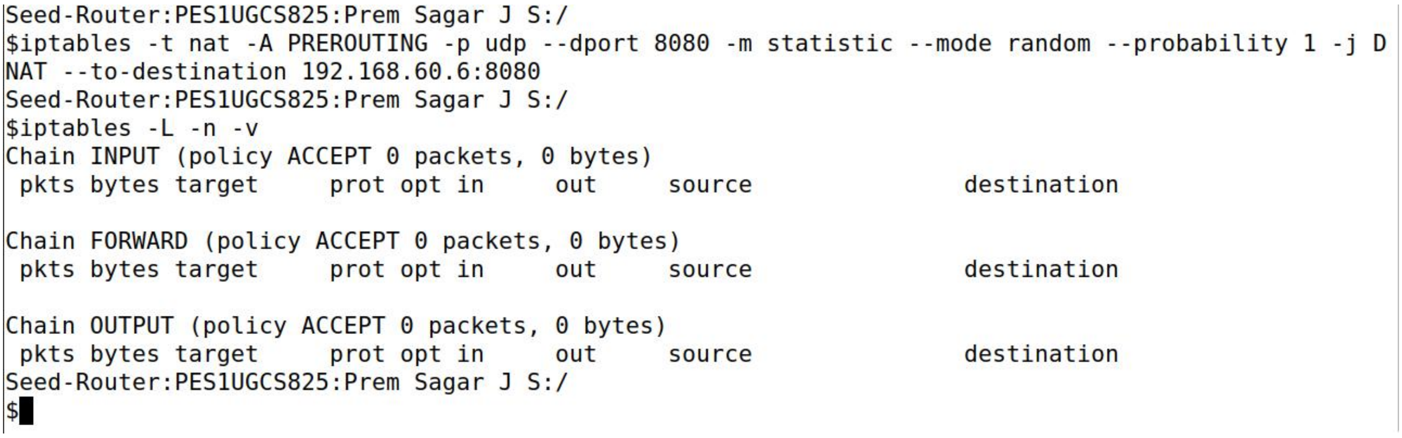
Using the random mode - Let’s use a different mode to achieve load balancing. The following rule will select a matching packet with the probability P. You need to replace P with a probability number.

On the seed-router container:
Command:

```
# 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
# 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
# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random
--probability 1 -j DNAT --to-destination 192.168.60.6:8080
# iptables -L -n -v
```



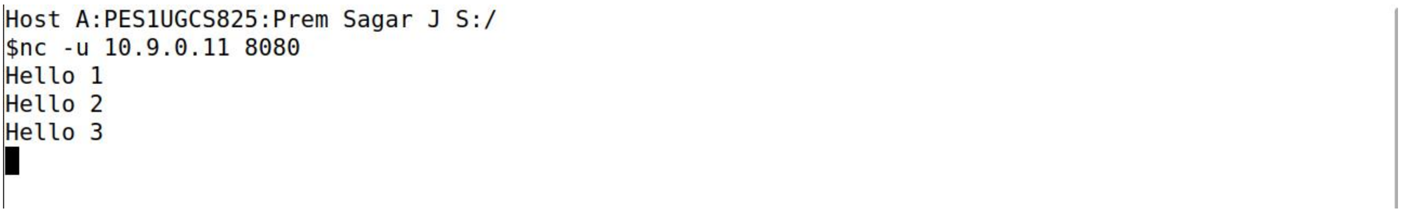
```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 0.3333
 -j DNAT --to-destination 192.168.60.5:8080
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 0.5 -j
 DNAT --to-destination 192.168.60.6:8080
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 1 -j D
NAT --to-destination 192.168.60.6:8080
```



```
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 1 -j D
NAT --to-destination 192.168.60.6:8080
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$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
Seed-Router:PES1UGCS825:Prem Sagar J S:/
$
```

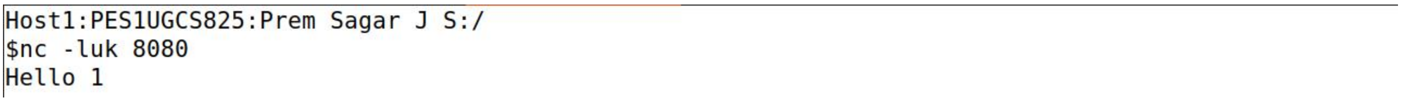
- On host1 - 192.168.60.5, host2 - 192.168.60.6, host3 - 192.168.60.7 started the server using Command: nc -luk 8080

On host A - 10.9.0.5
Command:
# nc -u 10.9.0.11 8080



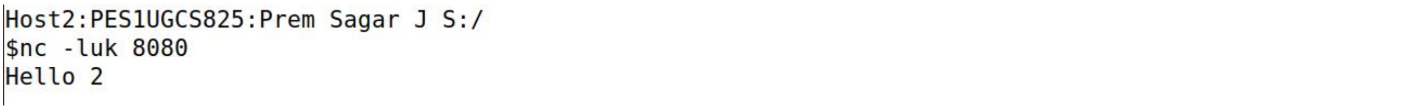
```
Host A:PES1UGCS825:Prem Sagar J S:/
$nc -u 10.9.0.11 8080
Hello 1
Hello 2
Hello 3
```

On host1 - 192.168.60.5



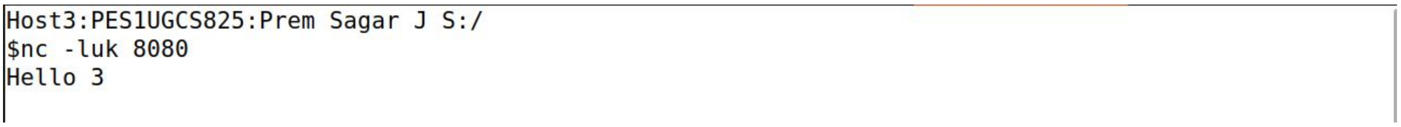
```
Host1:PES1UGCS825:Prem Sagar J S:/
$nc -luk 8080
Hello 1
```

On host2 - 192.168.60.6



```
Host2:PES1UGCS825:Prem Sagar J S:/
$nc -luk 8080
Hello 2
```

On host3 - 192.168.60.7



```
Host3:PES1UGCS825:Prem Sagar J S:/
$nc -luk 8080
Hello 3
```