**Lab-5**

* By Vikramadithya Ivaturi

**Task 1.A:**

Code:

#include <linux/module.h>

#include <linux/kernel.h>

int initialization(void)

{

printk(KERN\_INFO "Hello World!\n");

return 0;

}

void cleanup(void)

{

printk(KERN\_INFO "Bye-bye World!.\n");

}

module\_init(initialization);

module\_exit(cleanup);

MODULE\_LICENSE("GPL");

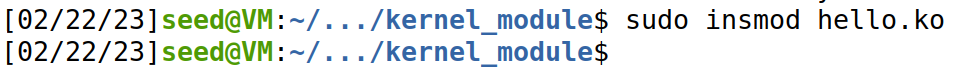
Output:

First, we need to execute the ‘make’ command in the directory to compile the program into a loadable kernel module.

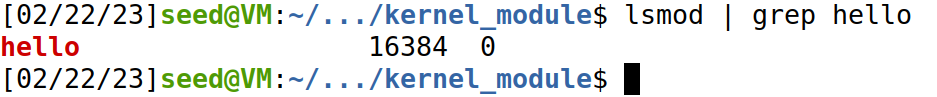
Text

Description automatically generated

After this, we need to load the ‘hello.ko’ file into the list of modules. For this, we use the command ‘sudo insmod hello.ko’.



The list of all kernel modules are below.



Now that we have inserted the module, we can verify the kernel log to check for the message ‘Hello World!’.



Now, we remove the kernel module from the list, and print another message to the log ‘Bye-bye World!’.

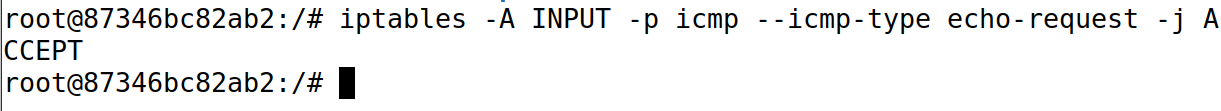
Text

Description automatically generated with medium confidence

**Task 2.A:**

Output:

Set a rule in the ‘filter’ table on the INPUT chain to accept incoming icmp packets into the router.



Now, try to ping the router from Host-A

Text, letter

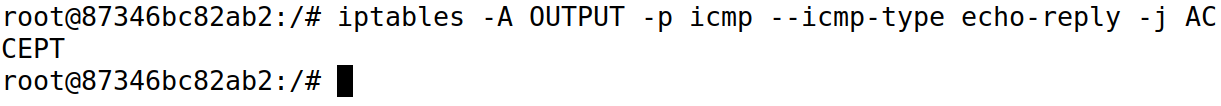
Description automatically generated

Now try connecting to the router via telnet.

Text, letter

Description automatically generated

Now, set another rule in the ‘filter’ table in the OUTPUT chain by executing the following command.



Now try pinging the router from host -A

Text

Description automatically generated

Now try telnet form host-A

Text, letter

Description automatically generated

Now, Set default rules for the ‘filter’ table for the INPUT chain and the OUTPUT chain to drop the packets in the router machine.

Timeline

Description automatically generated

Now, try to ping the machine.

Text, letter

Description automatically generated

Ping is working because there is an **exclusive rule to accept ‘icmp’ packets** in the INPUT and OUTPUT chains in the filter tables (highlighted above)

Now try telnetting to the machine.

A picture containing text

Description automatically generated

telnet is unsuccessful as it is a TCP connection, which means it is bocking all incoming packets except ‘icmp’.

To prove the outgoing rules, we can try establishing a telnet connection from the router to a host.

A picture containing chart

Description automatically generated

This means all the rules that are established have fulfilled their respective purposes.

**Task-2.B:**

Output:

Reset the filter table on the machine.

Text

Description automatically generated

1. **Outside hosts cannot ping internal hosts.**
2. **Outside hosts can ping the router.**

Execute the below command to insert a rule into the filter table on the FORWARD chain to drop any icmp packet that tries to ping the internal hosts.

Text

Description automatically generated with medium confidence

Now try to ping the router (for which you will succeed),

Text

Description automatically generated

then try to ping internal host1 (for which you will fail).

Text

Description automatically generated

1. **Internal hosts can ping outside hosts.**
2. **All other packets between the internal and external networks should be blocked.**

First, we set the default rule in the forward chain to drop the packets.

Text

Description automatically generated with medium confidence

Now, we allow icmp packets from internal hosts on interface eth1 to the outside hosts, and allow replies from the outside hosts on the interface eth0.

A picture containing text

Description automatically generated

Now, ping outside host from internal host1

Text

Description automatically generated

Noe, ping host1 from hostA

Text

Description automatically generated

To test if any other traffic is being allowed on the network, we can telnet from host1 to hostA, and back from hostA to host1.

Telnetting hostA (external) from host1(internal)

Chart

Description automatically generated

Telnetting host1 (internal) form hostA(external)

A picture containing text

Description automatically generated

The firewall is acting as expected.

**Task 2.C:**

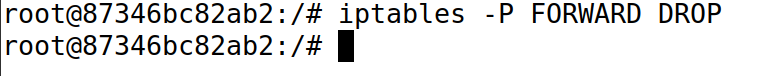
Output:

Reset all the filters.

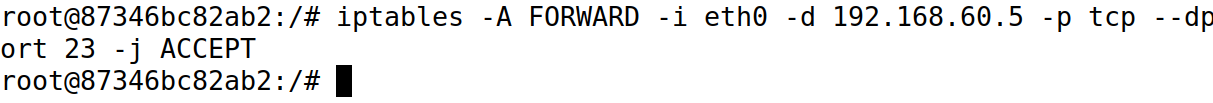
Text

Description automatically generated

First we make the default policy for FORWARDING to be DROP



Then accept TCP packets from external hosts to internal host1.



Then accept the TCP packet forwarding from host1 to external hosts.



Text

Description automatically generated with medium confidence

1. **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.**

Now, try telnetting host1 from external hostA

Text

Description automatically generated

The connection is successful.

Now try telnetting to another internal machine.

A picture containing chart

Description automatically generated

TCP packets are getting dropped, so there is no response.

1. **Internal hosts can access all the internal servers.**

Try telnetting host2 from host1

Text

Description automatically generated

1. **Internal hosts cannot access external servers.**

Listen on the hostA(external) on port 9090.

A picture containing diagram

Description automatically generated

Sending TCP packets to hostA(external) from host1(internal)

A picture containing text

Description automatically generated

Receiver’s side

A picture containing text

Description automatically generated

The packets have not been received.

So, the firewall rules are working as expected.

**Task 3.A:**

Output:

1. **ICMP experiment**

First ping the internal network host host1 from hostA.

Text

Description automatically generated

Now run connection tracking on the router continuously and check for the time the status changes.

A picture containing text, newspaper, document

Description automatically generated

.

.

.

.

A screenshot of a computer

Description automatically generated with low confidence

It took about 30 seconds for the state to get refreshed.

1. **UDP experiment**

First listen on host1 on port 9090

A picture containing text

Description automatically generated

Now post a message from hostA to host1



You can monitor the connection on the router

Text, letter

Description automatically generated

This explains that the state time was 30 seconds.

1. **TCP experiment**

First listen on host1 on port 9090



Now, post a message from hostA

A picture containing logo

Description automatically generated

Monitor the connection on the router

Text

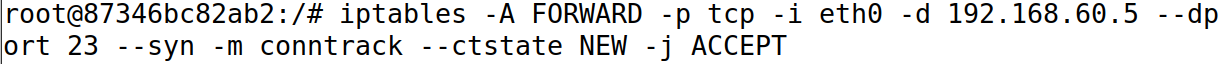
Description automatically generated

This means the connection state is going to be maintained for 432000 seconds for a TCP connection.

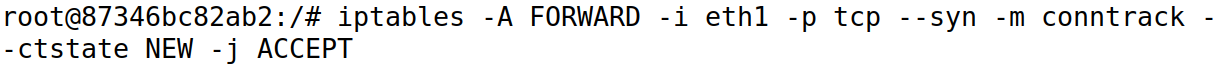
**Task 3.B:**

Output:

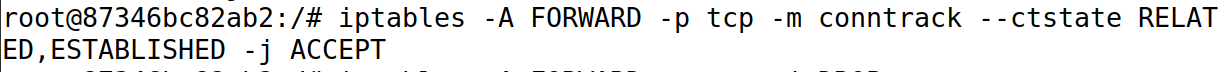
First establish the rule to accept tcp packets from external hosts to internal host1.



Then accept TCP packets on the internal interface for the 3 way handshake



Then accept tcp packets of any established connections between internal and external machines.



Set a default rule for the tcp packets that comes after the above rules (in sequence) dropping any other tcp packets.



Set default ACCEPT rule for the FORWARD chain



Now, telnet to the internal machine from external machine.

Text, letter

Description automatically generated

telnet from host1 to external machine.

Text

Description automatically generated

Now, try to telnet from external machine to the other internal machine.

A picture containing diagram

Description automatically generated

Now, telnet from other internal machine to external machine.

Text

Description automatically generated

**Task 4:**

Output:

First we clean the tables, and place the 1st rule in the tables.

Text

Description automatically generated

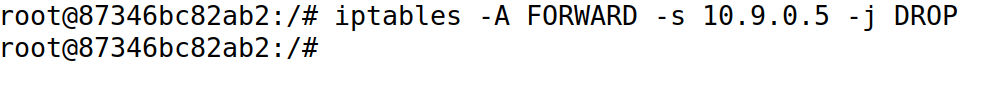
Now, we try to ping hostA (external) from host1 (internal).

Text

Description automatically generated

This rule is not working as expected, there is no effect of this rule on the network traffic.

Implementing the rule 2.



Again, we try to ping hostA (external) from host1 (internal).

Table

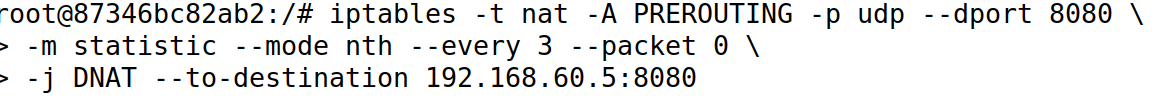
Description automatically generated

Now, we see response packets form the external host are dropped for every 6 packets sent.

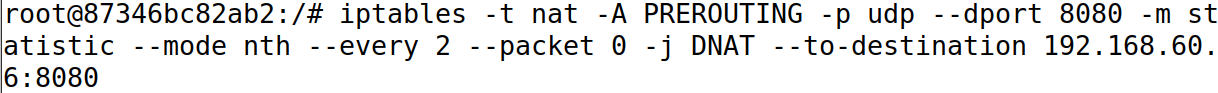
**Task 5:**

1. **Round robin**

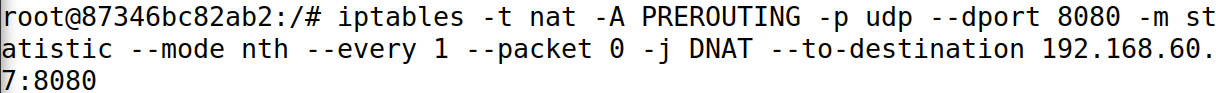
First apply the give rule to the NAT table while forwarding the packets.



The above rule sends every 3rd packet to host1



The above rule sends every 2nd packet to host2



Above rule sends the remaining packet to host3

Now, listen for UDP packets on all the 3 hosts







Now, send messages from hostA (external) to the router at port 8080.

Text

Description automatically generated

Checking the internal hosts for messages,

Host1

A picture containing text

Description automatically generated

Host2

A picture containing text

Description automatically generated

Host3

A picture containing text

Description automatically generated

1. **Random mode**

Set the probability rule in the router giving each internal host 0.33333 as the probability.

Text

Description automatically generated

Now, listen on all the internal hosts

A picture containing shape

Description automatically generated

A picture containing text

Description automatically generated

A picture containing logo

Description automatically generated

Now send messages from external host to the router

Text, letter

Description automatically generated

Now, check for the messages in the internal servers

Host1

A picture containing text

Description automatically generated

Host2

A picture containing text

Description automatically generated

Host3

A picture containing text

Description automatically generated

Hence the load balancing is working perfectly.