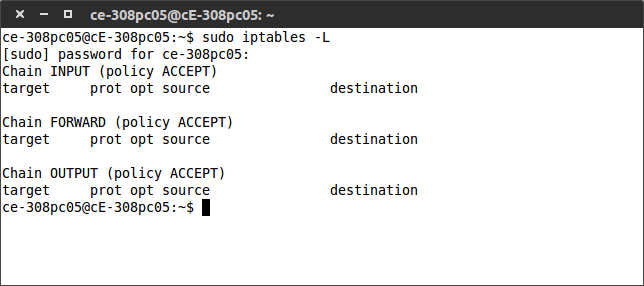
Experiment No. \_\_

#### **Aim:** To configure iptables in ubuntu.

**Theory:**

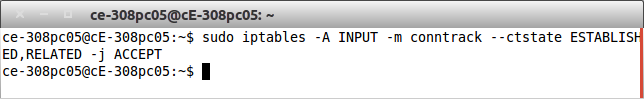
#### **Listing current rules**

Ubuntu servers do not implement any restrictions by default, but for future reference, check the current iptable rules using the following command

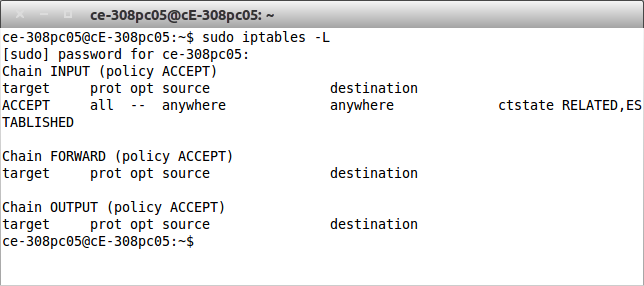


#### **Adding rules**

To begin using iptables, we should first add the rules for allowed inbound traffic for the services you require. Iptables can track the state of the connection, so use the command below to allow established connections to continue.

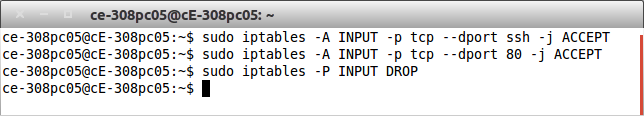


You can check that the rule was added using the same *sudo iptables -L* as before.



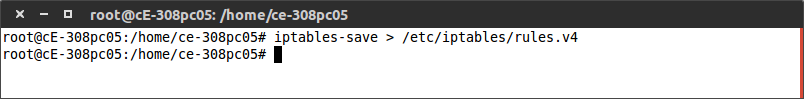
Next allow traffic to a specific port to enable SSH connections with the following.

After adding all the allowed rules you require, change the input policy to drop.

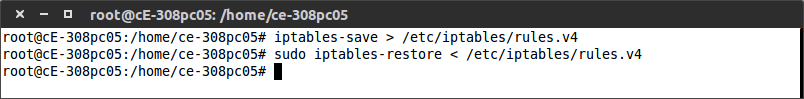


#### **Saving and restoring rules**

Now if you were to restart your cloud server all of these iptables configurations would be wiped. To prevent this, save the rules to a file.



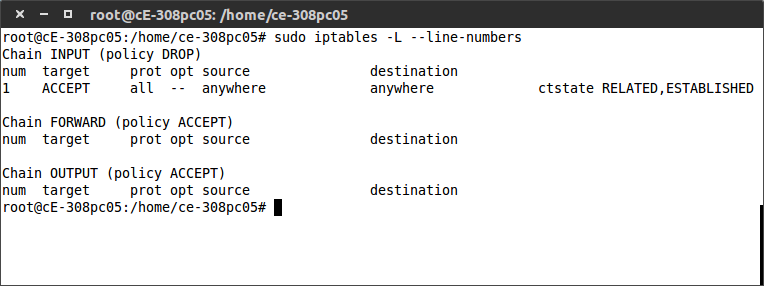
Now we can restore the saved rules by the following command.



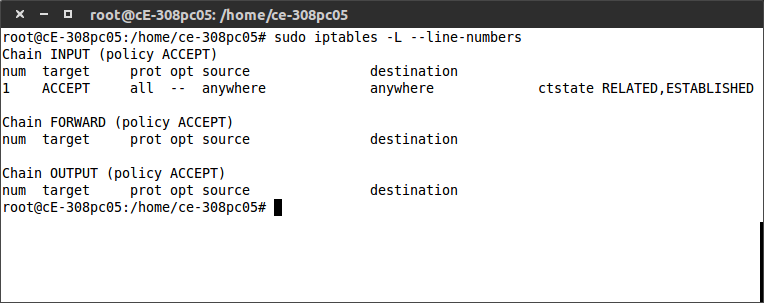
**Advanced rule setup**

Appending new rules adds them to the end of the list. We can add new rules to a specific position of the list by inserting them using iptables -I <index> -command, where the <index> is the order number you wish to insert the rule. To know which index number to enter, use the following command.

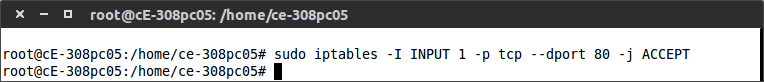
*For drop policy*



*For accept policy*

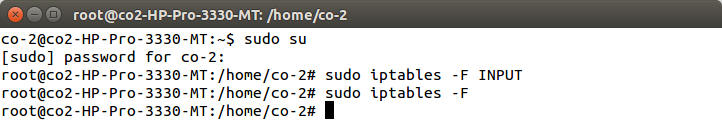


To insert a new rule above a specific existing rule, simply use the index number of that existing rule.



If you wish to remove an existing rule from a certain chain, use the delete command with the parameter -D.

It’s also possible to flush all rules of a specific chain or even the whole iptables using the -F -parameter.



**Conclusion :**

Hence we studied and configured the iptables in ubuntu.