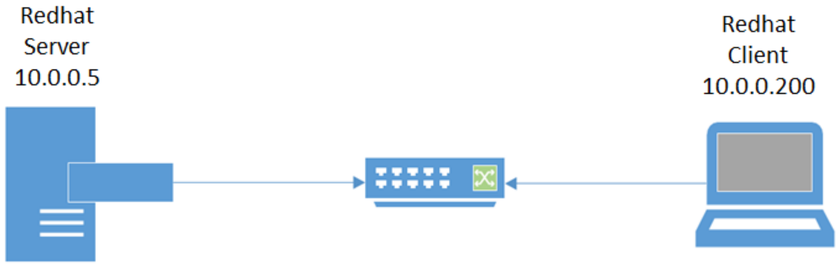
Practical 12: Netfilters – iptables

Lab 12.1 : Configuring a Basic IPv4 Firewall

Scenario: A host (server) requires protection by packet filtering. This host has only one network interface, so no packet forwarding is involved.



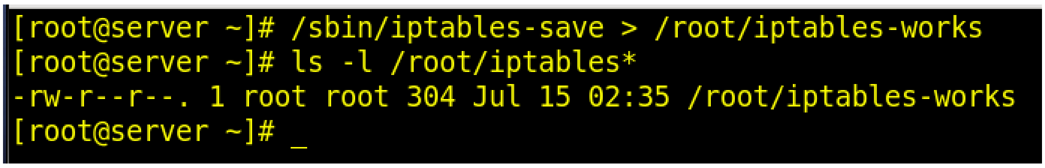
Objective: A set of Ipv4 packet filtering rules in a custom chain called CLASS-RULES, allowing icmp and cups packets, stateful (ESTABLISHED, RELATED) replies, NEW packets connecting to sshd and packets arriving on the loopback (lo) interface. All other packets will be LOGged before being REJECTed with icmp-host-prohibited.

Instructions:

1. Take a snapshot of your Server and name the snapshot as Server_Wk14.
2. Log in to the Client as student.
3. Remote log in from the Client to the Server via SSH and escalate your privileges to root with su-

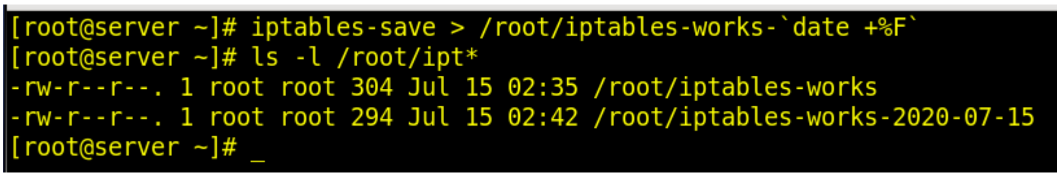
```
[student@client ~]$ ssh server
Password: redhat
[student@client ~]$ su –
Password: redhat
[root@server ~] #
```

4. Backup your iptables configuration first. The /sbin is optional.



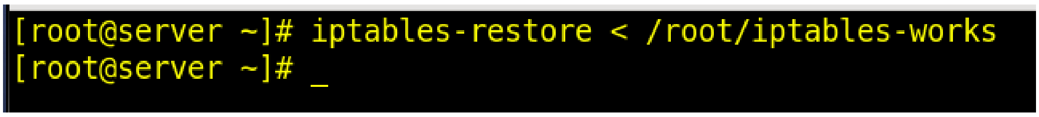
```
[root@server ~]# /sbin/iptables-save > /root/iptables-works
[root@server ~]# ls -l /root/iptables*
-rw-r--r--. 1 root root 304 Jul 15 02:35 /root/iptables-works
[root@server ~]# _
```

You can also backup it up with a date stamp as shown below:



```
[root@server ~]# iptables-save > /root/iptables-works-`date +%F`
[root@server ~]# ls -l /root/ipt*
-rw-r--r--. 1 root root 304 Jul 15 02:35 /root/iptables-works
-rw-r--r--. 1 root root 294 Jul 15 02:42 /root/iptables-works-2020-07-15
[root@server ~]# _
```

5. If you mess up the iptables, you can restore it with the following command:



```
[root@server ~]# iptables-restore < /root/iptables-works
[root@server ~]# _
```

6. Ensure that the firewall is stopped, processing no rules.

```
[root@server ~]# service iptables stop
Flushing firewall rules: -for flush memory    [OK]
Setting chains to policy ACCEPT: fliter [OK]
Unloading iptables modules:             [OK]
```

7. Create a custom chain called CLASS-RULES and insert a rule at the top of input that jumps all packets to it. Save the firewall configuration when you are done.

```
[root@server ~]# iptables -N CLASS-RULES
[root@server ~]# iptables -A INPUT -j CLASS-RULES
[root@server ~]# service iptables save
Saving firewall rules to /etc/sysconfig/iptables:
[OK]
```

8. Populate the CLASS-RULES by editing **/etc/sysconfig/iptables** directly.

   Add rules that do the following:

   - ACCEPT all traffic arriving on the loopback interface(lo)
   - ACCEPT all packets that use the icmp protocol.
   - ACCEPT all packets destined for both udp and tcp ports 631(cups) (Hint: You will need two rules for this)
   - ACCEPT packets with the ESTABLISHED or RELATED state
   - ACCEPT packets destined for tcp port 22 (ssh)
   - LOG and REJECT (with icmp-host-prohibited) all packets not matched by one of the above rules
     (Hint: You will need two rules for this)

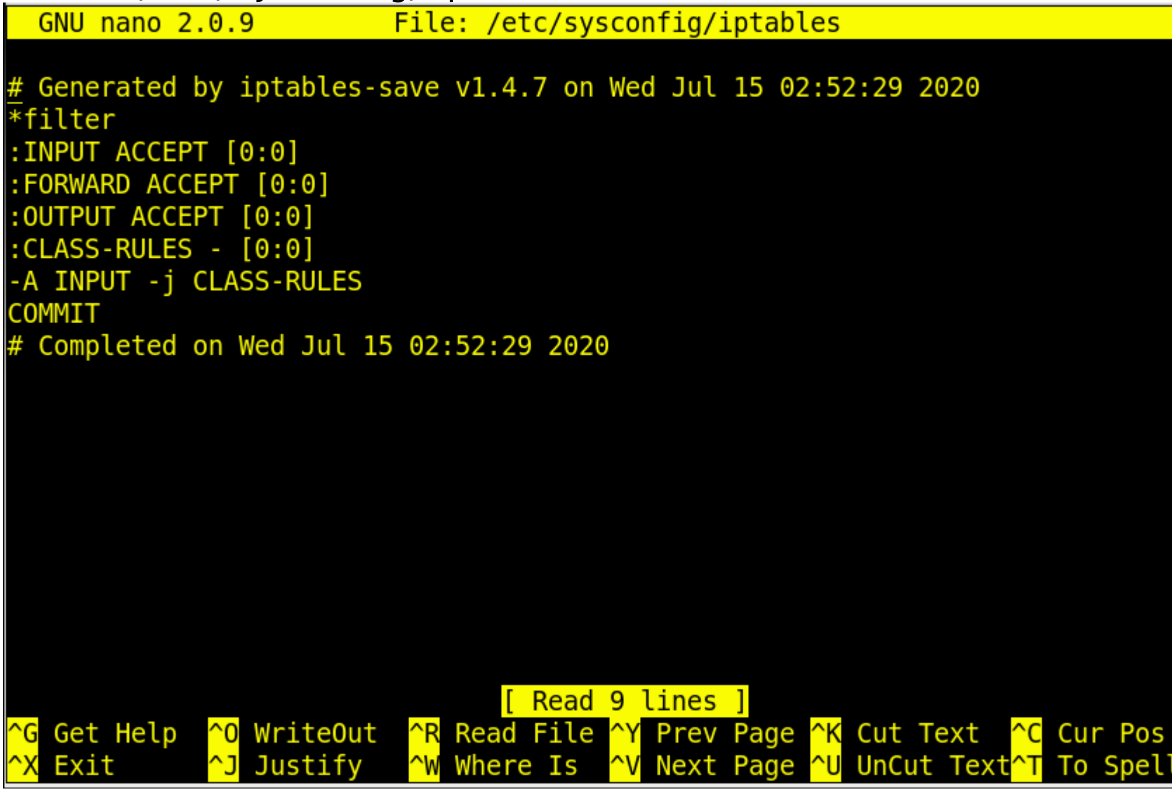
   Add the following lines to /etc/sysconfig/ iptables. Remember that you can save yourself some typing by cut-and-pasting similar lines. Add the following text just below the –A INPUT lines and above the COMMIT line.

```
-A CLASS-RULES  -i lo -j ACCEPT
-A CLASS-RULES  -p icmp -j ACCEPT
-A CLASS-RULES  -p udp --dport 631 -j ACCEPT
-A CLASS-RULES  -p tcp  --dport 631 -j ACCEPT
-A CLASS-RULES -m state --state ESTABLISHED,
RELATED -j ACCEPT
-A CLASS-RULES  -p tcp  --dport 22 -j ACCEPT
-A CLASS-RULES  -j LOG
-A CLASS-RULES  -j REJECT --reject-with icmp-
host-prohibited
```

(note: no space after comma ie. ESTABLISHED,RELATED)

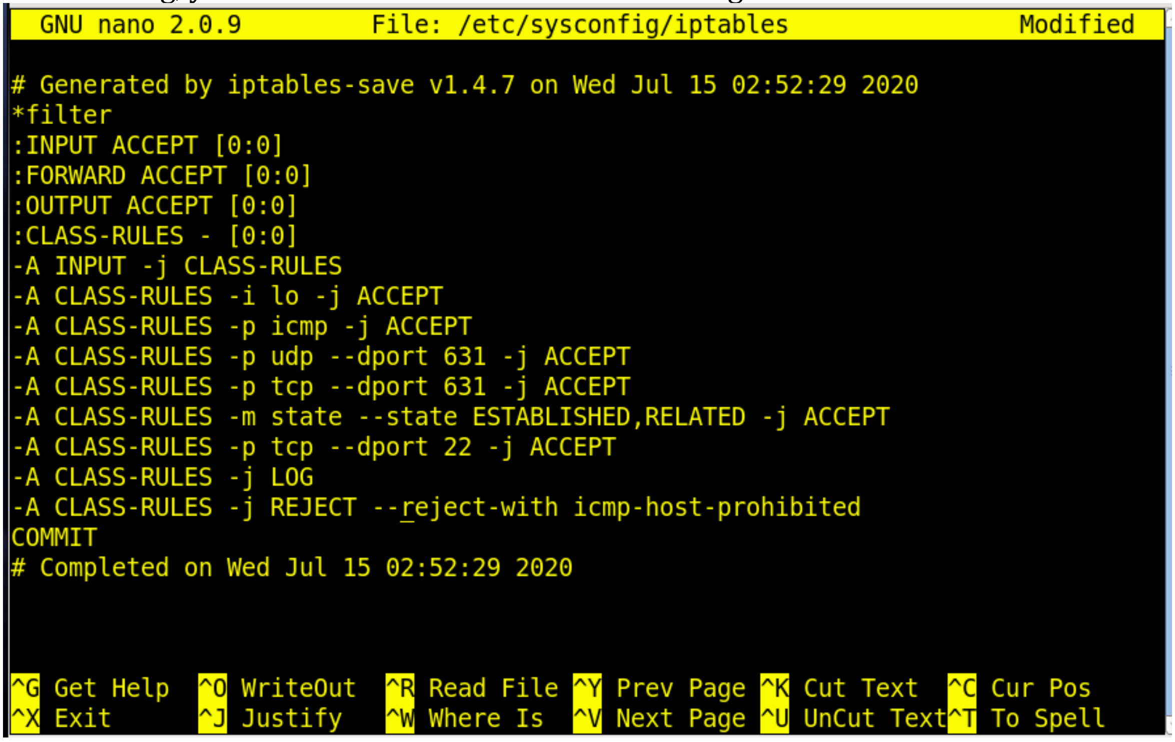
You can use any editor to edit the /etc/sysconfig/iptables file. The screen below uses the nano editor.

```
# nano /etc/sysconfig/iptables
```



```
# Generated by iptables-save v1.4.7 on Wed Jul 15 02:52:29 2020
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:CLASS-RULES - [0:0]
-A INPUT -j CLASS-RULES
COMMIT
# Completed on Wed Jul 15 02:52:29 2020
```

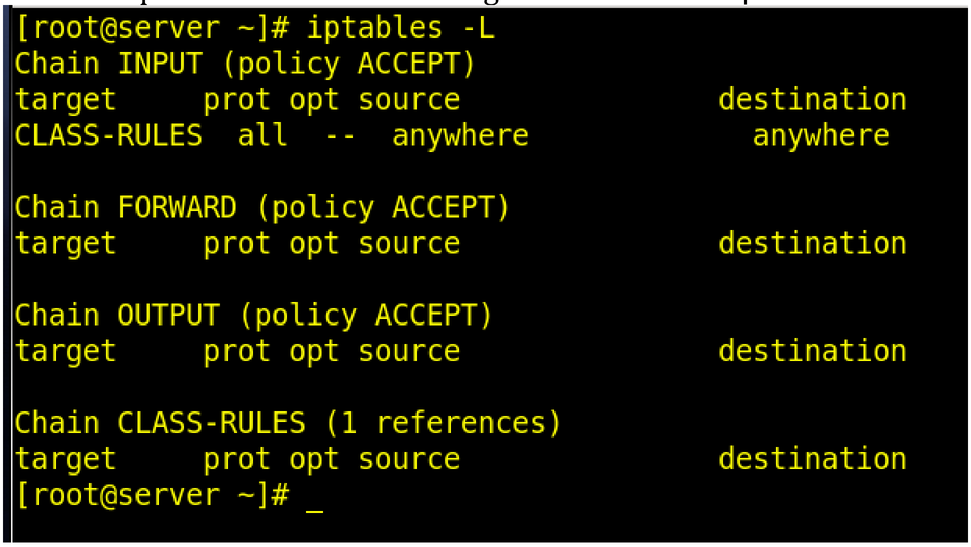
After editing, your nano screen should look something like this:



```
# Generated by iptables-save v1.4.7 on Wed Jul 15 02:52:29 2020
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:CLASS-RULES - [0:0]
-A INPUT -j CLASS-RULES
-A CLASS-RULES -i lo -j ACCEPT
-A CLASS-RULES -p icmp -j ACCEPT
-A CLASS-RULES -p udp --dport 631 -j ACCEPT
-A CLASS-RULES -p tcp --dport 631 -j ACCEPT
-A CLASS-RULES -m state --state ESTABLISHED,RELATED -j ACCEPT
-A CLASS-RULES -p tcp --dport 22 -j ACCEPT
-A CLASS-RULES -j LOG
-A CLASS-RULES -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Wed Jul 15 02:52:29 2020
```

Press Ctrl-o to save your file and Ctrl-x to exit.

View the iptables before committing the new rules – `iptables -L`

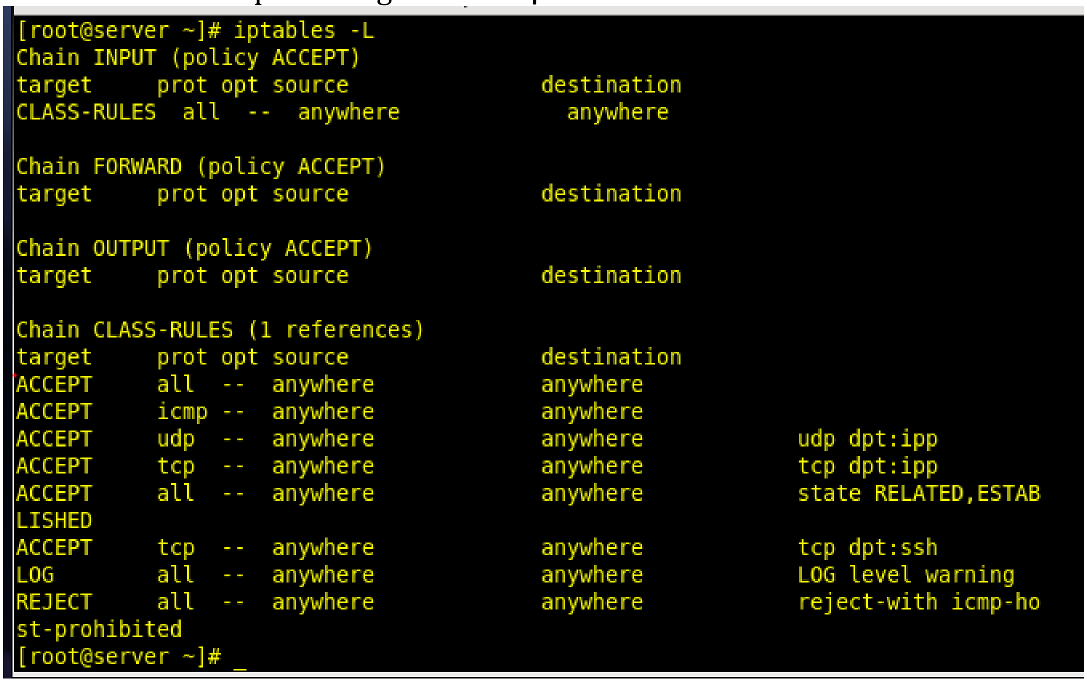


```
[root@server ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
CLASS-RULES  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain CLASS-RULES (1 references)
target     prot opt source               destination
[root@server ~]# _
```

9. Load your new rules.

```
[root@server ~] # service iptables restart
Flushing firewall rules:          [OK] Setting
chains to policy ACCEPT: filter   [OK] Unloading
iptables modules:                 [OK] Applying
iptables firewalls rules:         [OK]
```

10. View the iptables again with `iptables -L`



```
[root@server ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
CLASS-RULES  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain CLASS-RULES (1 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     udp  --  anywhere             anywhere            udp dpt:ipp
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ipp
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTAB
LISHED
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
LOG        all  --  anywhere             anywhere            LOG level warning
REJECT     all  --  anywhere             anywhere            reject-with icmp-ho
st-prohibited
[root@server ~]# _
```

11. On the client, attempt to ssh to server. This should work.

```
[student@client ~]$ ssh server
Password: redhad
[student@server ~]$
```

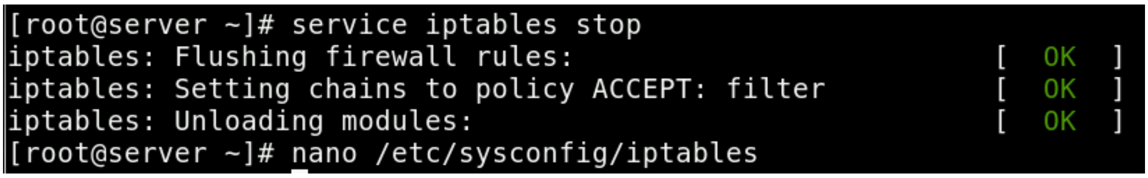
12. Modify your rules to restrict ssh connections to only be accepted from 10.0.0.200 (your client's ip address).

- Change the /etc/sysconfig/iptables line that is filtering – dport 22 to look like the following:

```
-A CLASS-RULES  - p tcp -dport 22 -s 10.0.0.200 - j ACCEPT
```

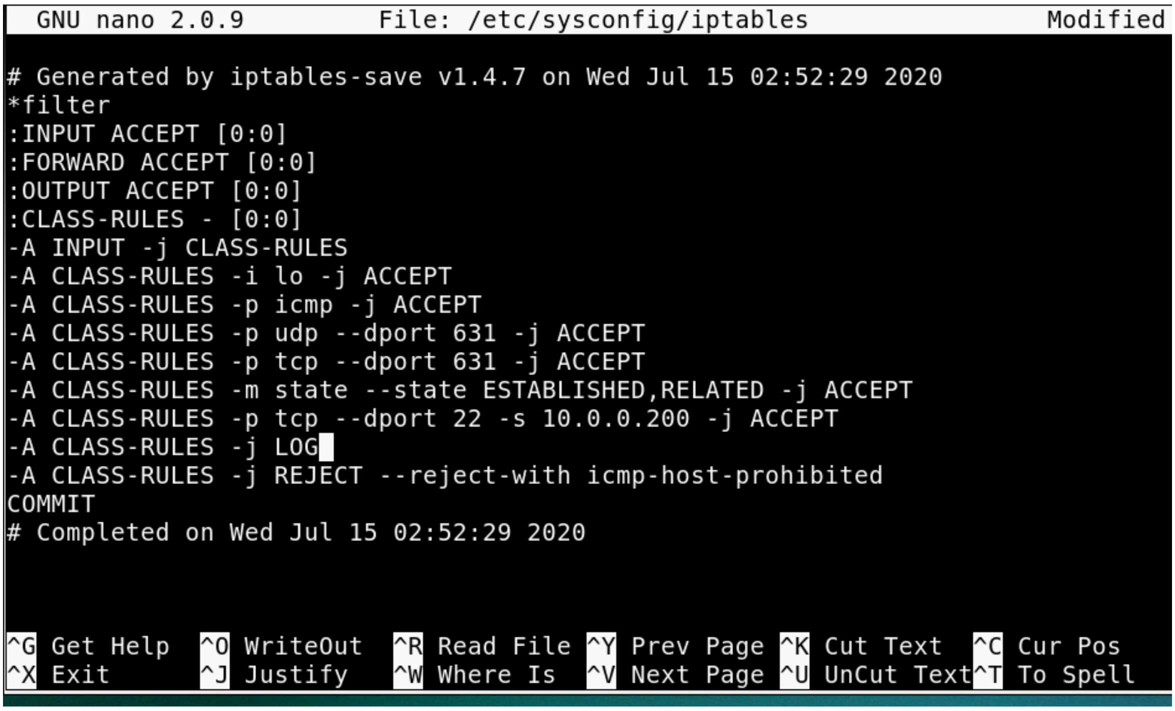
```
[root@server ~]# service iptables restart
```

13. Stop the iptable



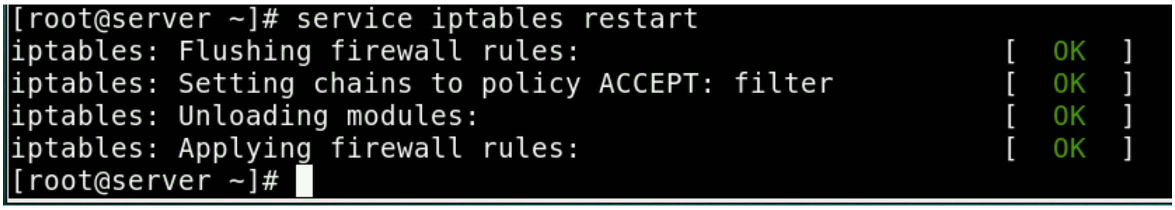
```
[root@server ~]# service iptables stop
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
[root@server ~]# nano /etc/sysconfig/iptables
```

14. Use nano to edit the iptables file



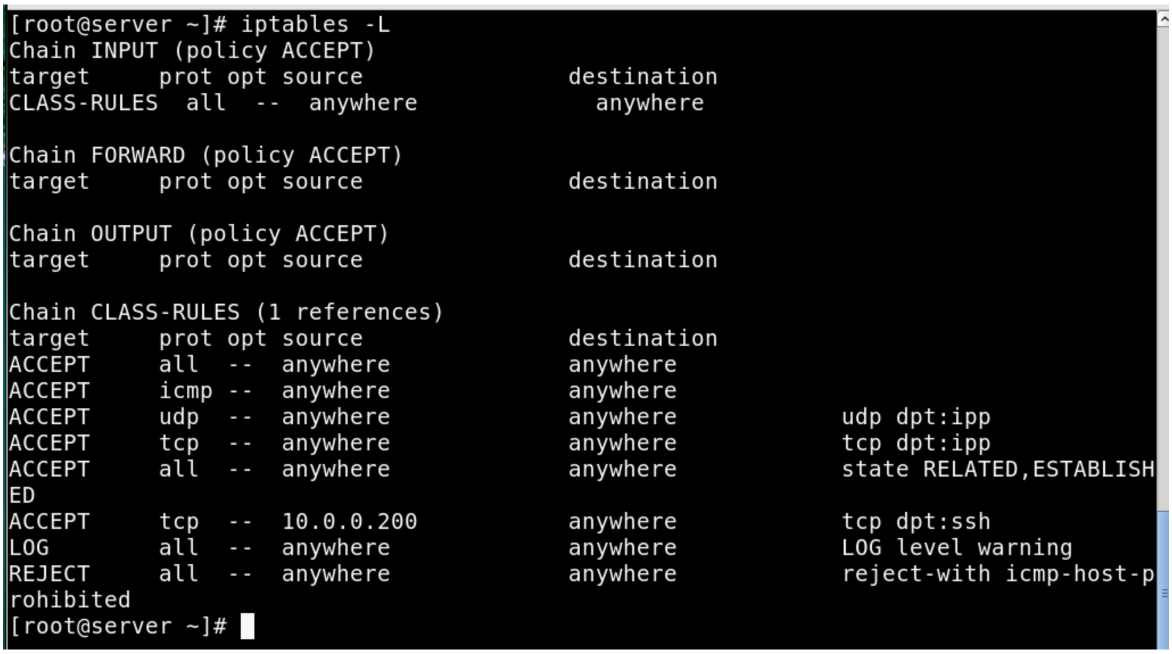
```
  GNU nano 2.0.9           File: /etc/sysconfig/iptables                Modified

# Generated by iptables-save v1.4.7 on Wed Jul 15 02:52:29 2020
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:CLASS-RULES - [0:0]
-A INPUT -j CLASS-RULES
-A CLASS-RULES -i lo -j ACCEPT
-A CLASS-RULES -p icmp -j ACCEPT
-A CLASS-RULES -p udp --dport 631 -j ACCEPT
-A CLASS-RULES -p tcp --dport 631 -j ACCEPT
-A CLASS-RULES -m state --state ESTABLISHED,RELATED -j ACCEPT
-A CLASS-RULES -p tcp --dport 22 -s 10.0.0.200 -j ACCEPT
-A CLASS-RULES -j LOG
-A CLASS-RULES -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Wed Jul 15 02:52:29 2020
```

15. Restart iptable service



```
[root@server ~]# service iptables restart
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
iptables: Applying firewall rules:                         [  OK  ]
[root@server ~]#
```

16. List the iptable to verify.



```
[root@server ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
CLASS-RULES  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain CLASS-RULES (1 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     udp  --  anywhere             anywhere            udp dpt:ipp
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ipp
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISH
ED
ACCEPT     tcp  --  10.0.0.200           anywhere            tcp dpt:ssh
LOG        all  --  anywhere             anywhere            LOG level warning
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-p
rohibited
[root@server ~]# 
```

17. On client, attempt to ssh to server. This should still work.

18. Change the ip of client to 10.0.0.100 and attempt to ssh to server.

19. What happened?

20. When done, close your ssh sessions and reboot the server and client.

[The End]