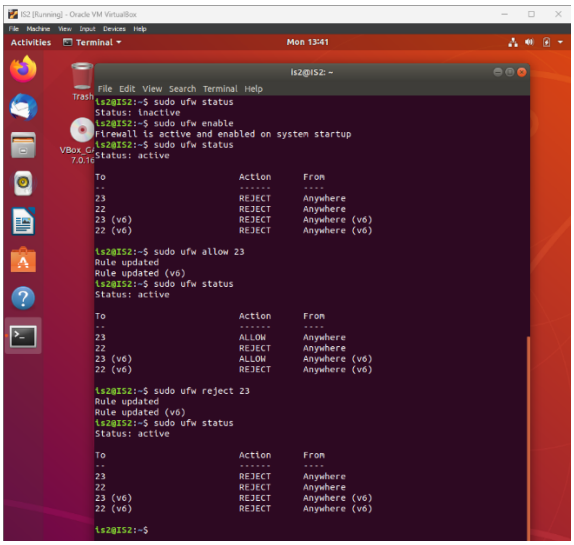
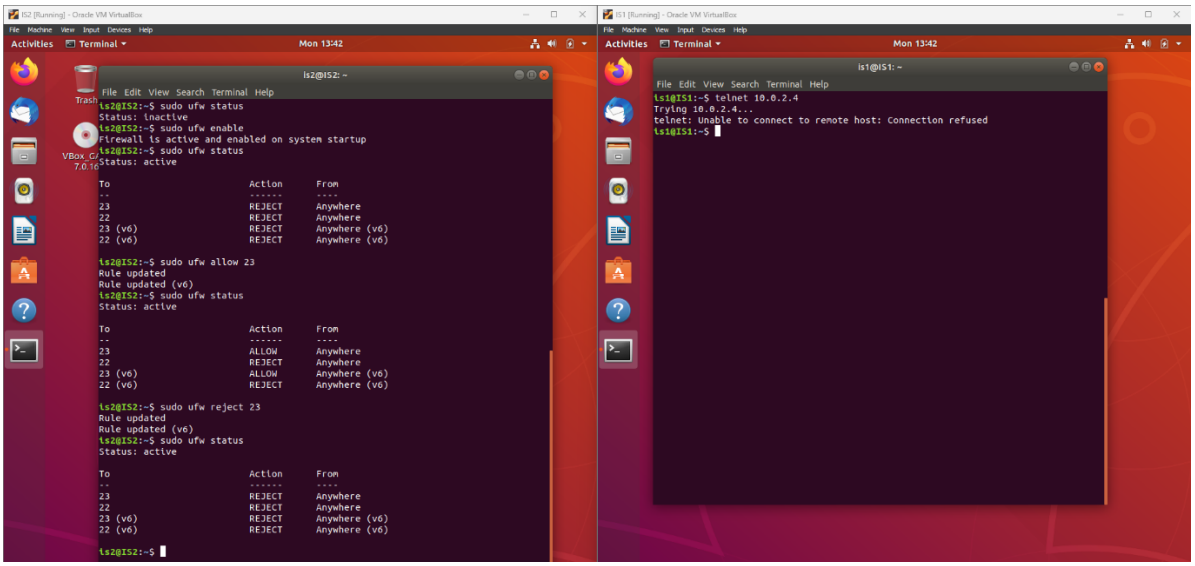
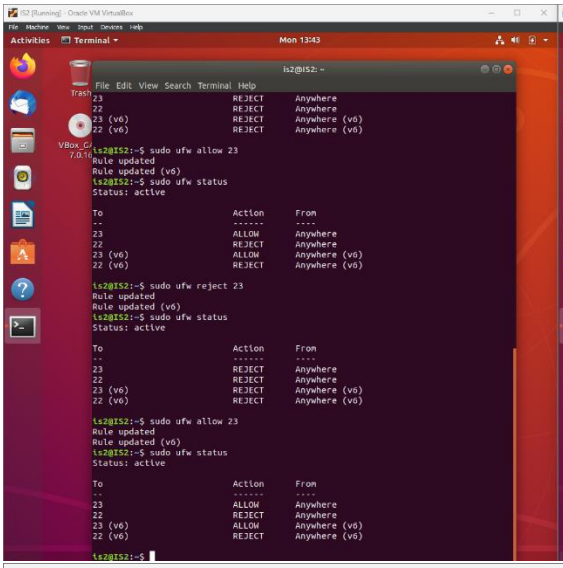
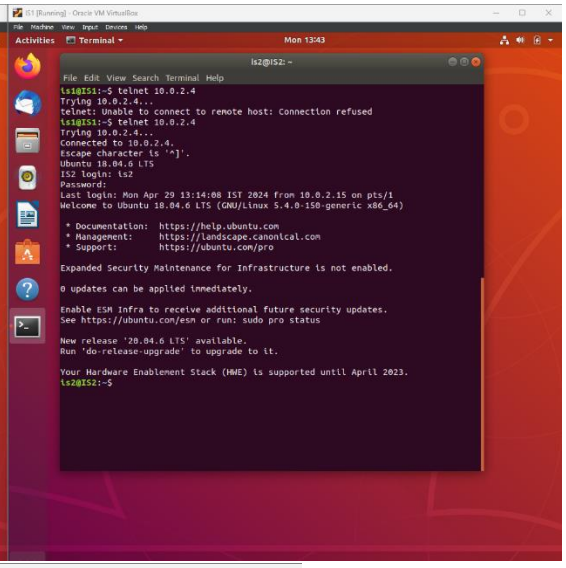
IP TABLES & FIREWALL ASSIGNMENT

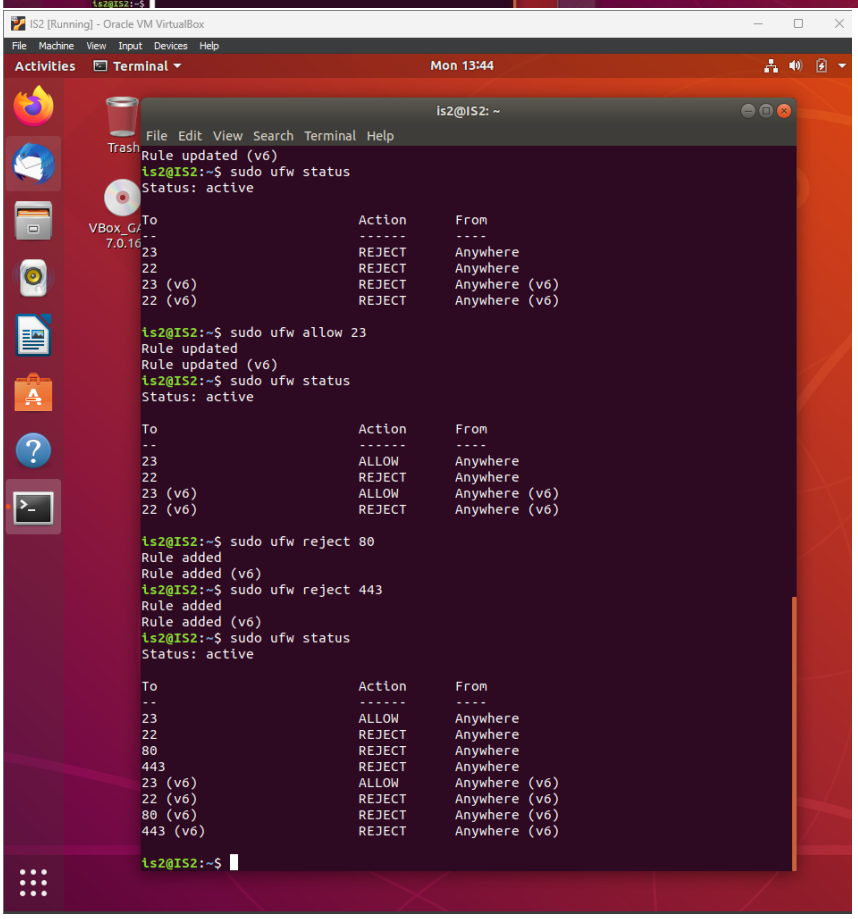
1. On your machine, you need to block incoming connections to the ports 22 (ssh), 23 (telnet), 80 (HTTP), and 443 (HTTPS). What will you do? Write rules for the same and demonstrate them. Also explain the purpose of each rule. (You may use iptables – an inbuilt firewall in Linux Systems)
    - We can demonstrate the by using ufw (uncomplicated firewall) as shown below. We can see that incoming from port 23 is blocked since when we try to access from other VM the connection refuses to establish.











- Now by using the commands of the iptables we will do the same.
  Blocking Incoming Connections to Specific Ports : 443, 23, 22, 80
  Block SSH (port 22): iptables -A INPUT -p tcp --dport 22 -j DROP
  Block Telnet (port 23): iptables -A INPUT -p tcp --dport 23 -j DROP
  Block HTTP (port 80): iptables -A INPUT -p tcp --dport 80 -j DROP
  Block HTTPS (port 443): iptables -A INPUT -p tcp --dport 443 -j DROP

The purpose of these rules is to prevent incoming connections on these specific ports, enhancing security by restricting access to services like SSH, Telnet, HTTP, and HTTPS.

Allowing Outgoing and Specific Incoming TCP Traffic

Allowing Outgoing and Specific Incoming TCP Traffic:

```
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A OUTPUT -p tcp -j ACCEPT
is2@is2:~$ iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
is2@is2:~$ iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
is2@is2:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:ssh
DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
DROP       tcp  --  anywhere             anywhere             tcp dpt:https
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             owner UID match bob
ACCEPT     tcp  --  anywhere             anywhere
is2@is2:~$ sudo iptables -A OUTPUT -m owner --uid-owner Bob -j DROP
iptables v1.6.1: owner: Bad value for "--uid-owner" option: "Bob"
Try `iptables -h' or 'iptables --help' for more information.
is2@is2:~$ sudo iptables -A INPUT -p icmp --icmp-type 8 -j DROP
is2@is2:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:ssh
DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
DROP       tcp  --  anywhere             anywhere             tcp dpt:https
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       icmp --  anywhere             anywhere             icmp echo-request

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             owner UID match bob
ACCEPT     tcp  --  anywhere             anywhere
is2@is2:~$ sudo iptables -A OUTPUT -m owner --uid-owner Bob -j DROP
```

2. Allow all outgoing TCP traffic but incoming traffic only on TCP ports 22 and 80.

To allow all outgoing TCP traffic and incoming traffic only on ports 22 and 80, we can use the following rules:

Allow outgoing TCP traffic: iptables -A OUTPUT -p tcp -j ACCEPT

Allow incoming TCP traffic on ports 22 and 80: iptables -A INPUT -p tcp --dport 22 -j ACCEPT and iptables -A INPUT -p tcp --dport 80 -j ACCEPT

These rules enable unrestricted outgoing TCP connections while permitting incoming traffic only on ports 22 (SSH) and 80 (HTTP).

```
See "man sudo_root" for details.

is2@is2:~$ sudo iptables =l
[sudo] password for is2:
Bad argument `=l'
Try `iptables -h' or 'iptables --help' for more information.
is2@is2:~$ sudo iptables -l
iptables v1.6.1: unknown option "-l"
Try `iptables -h' or 'iptables --help' for more information.
is2@is2:~$ sudo iptables -L
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
is2@is2:~$ clear
is2@is2:~$ iptables -t filter -L -n --line-numbers
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -t filter -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
is2@is2:~$ sudo iptables -L
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
is2@is2:~$ whoami
is2
is2@is2:~$ sudo adduser bob
Adding user `bob' ...
Adding new group `bob' (1001) ...
Adding new user `bob' (1001) with group `bob' ...
Creating home directory `/home/bob' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

3. Create a machine with the name ‘Bob and write a rule to drop the packets generated by any program owned by user Bob on your machine. Other users should not be affected.

Blocking Packets from a Specific User

Blocking Packets from User 'Bob':

To drop packets generated by any program owned by user Bob, you can create a rule to block traffic specifically from that user:

iptables -A OUTPUT -m owner --uid-owner Bob -j DROP

This rule ensures that packets generated by any program owned by user Bob are dropped, while other users remain unaffected.

```
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
is2@is2:~$ whoami
is2
is2@is2:~$ sudo adduser bob
Adding user `bob' ...
Adding new group `bob' (1001) ...
Adding new user `bob' (1001) with group `bob' ...
Creating home directory `/home/bob' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bob
Enter the new value, or press ENTER for the default
        Full Name []: Bob the Builder
        Room Number []: 69
        Work Phone []: 69
        Home Phone []: 69
        Other []: 69
Is the information correct? [Y/n] y
is2@is2:~$ su bob
Password:
bob@is2:/home/is2$ cd
bob@is2:~$ ls
bob@is2:~$ iptables -A OUTPUT -m owner --uid-owner Bob -j DROP
owner: Could not determine whether revision 1 is supported, assuming it is.
iptables v1.6.1: owner: Bad value for "--uid-owner" option: "Bob"
Try `iptables -h' or 'iptables --help' for more information.
bob@is2:~$ iptables -A OUTPUT -m is2 --uid-is2 Bob -j DROP
iptables v1.6.1: Couldn't load match `is2':No such file or directory

Try `iptables -h' or 'iptables --help' for more information.
bob@is2:~$ iptables -A OUTPUT -m owner --uid-owner bob -j DROP
owner: Could not determine whether revision 1 is supported, assuming it is.
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
bob@is2:~$ sudo iptables -A OUTPUT -m owner --uid-owner bob -j DROP
[sudo] password for bob:
bob is not in the sudoers file.  This incident will be reported.
bob@is2:~$ su is2
Password:
is2@is2:/home/bob$ cd
is2@is2:~$ sudo iptables -A OUTPUT -m owner --uid-owner bob -j DROP
is2@is2:~$ su bob
Password:
bob@is2:/home/is2$ cd
bob@is2:~$ sudo iptables -L
[sudo] password for bob:
```

```
bob@is2:~$ su is2
Password:
is2@is2:/home/bob$ cd
is2@is2:~$ sudo iptables -A OUTPUT -m owner --uid-owner bob -j DROP
is2@is2:~$ su bob
Password:
bob@is2:/home/is2$ cd
bob@is2:~$ sudo iptables -L
[sudo] password for bob:
bob is not in the sudoers file.  This incident will be reported.
bob@is2:~$ iptables -L
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
bob@is2:~$ su is2
Password:
is2@is2:/home/bob$ cd
is2@is2:~$ sudo iptables -L
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
DROP       all  --  anywhere             anywhere             owner UID match bob
is2@is2:~$ iptables -A INPUT -p tcp --dport 22 -j DROP
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 22 -j DROP
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 23 -j DROP
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 80 -j DROP
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 443 -j DROP
is2@is2:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:ssh
DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
DROP       tcp  --  anywhere             anywhere             tcp dpt:https

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             owner UID match bob
is2@is2:~$ iptables -A OUTPUT -p tcp -j ACCEPT
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A OUTPUT -p tcp -j ACCEPT
```

75%

Screenshot from 2024-04-18 15-11-11.png

Activities Terminal Thu 15:11

is2@is2: ~

File Edit View Search Terminal Help

```
bob@is2:~$ su is2
Password:
is2@is2:/home/bob$ cd
is2@is2:~$ sudo iptables -A OUTPUT -m owner --uid-owner bob -j DROP
is2@is2:~$ su bob
Password:
bob@is2:/home/is2$ cd
bob@is2:~$ sudo iptables -L
[sudo] password for bob:
bob is not in the sudoers file.  This incident will be reported.
bob@is2:~$ iptables -L
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
bob@is2:~$ su is2
Password:
is2@is2:/home/bob$ cd
is2@is2:~$ sudo iptables -L
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
DROP       all  --  anywhere             anywhere             owner UID match bob
is2@is2:~$ iptables -A INPUT -p tcp --dport 22 -j DROP
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 22 -j DROP
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 23 -j DROP
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 80 -j DROP
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 443 -j DROP
is2@is2:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:ssh
DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
DROP       tcp  --  anywhere             anywhere             tcp dpt:https

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             owner UID match bob
is2@is2:~$ iptables -A OUTPUT -p tcp -j ACCEPT
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A OUTPUT -p tcp -j ACCEPT
```

75%

Screenshot from 2024-04-18 15-10-49.png

Activities Terminal Thu 15:10

is2@is2: ~

File Edit View Search Terminal Help

```
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A OUTPUT -p tcp -j ACCEPT
is2@is2:~$ iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
is2@is2:~$ iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables v1.6.1: can't initialize iptables table `filter': Permission denied (you must be root)
Perhaps iptables or your kernel needs to be upgraded.
is2@is2:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
is2@is2:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:ssh
DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
DROP       tcp  --  anywhere             anywhere             tcp dpt:https
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             owner UID match bob
ACCEPT     tcp  --  anywhere             anywhere
is2@is2:~$ sudo iptables -A OUTPUT -m owner --uid-owner Bob -j DROP
iptables v1.6.1: owner: Bad value for "--uid-owner" option: "Bob"
Try `iptables -h' or 'iptables --help' for more information.
is2@is2:~$ sudo iptables -A INPUT -p icmp --icmp-type 8 -j DROP
is2@is2:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:ssh
DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
DROP       tcp  --  anywhere             anywhere             tcp dpt:https
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       icmp --  anywhere             anywhere             icmp echo-request

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             owner UID match bob
ACCEPT     tcp  --  anywhere             anywhere
is2@is2:~$ sudo iptables -A OUTPUT -m owner --uid-owner Bob -j DROP
```

4. What are the different types of ICMP (Internet Control Message Protocol)? How can you Block Ping ICMP Requests to Linux Systems

The Internet Control Message Protocol (ICMP) encompasses various message types used for diagnostic and control purposes in IP networks. Some of the common ICMP message types include:

1. Echo Request/Echo Reply (Type 8/Type 0): Used for ping tests to check network connectivity and round-trip time.
2. Destination Unreachable (Type 3): Indicates that a packet cannot reach its destination, often due to network congestion, firewall rules, or other issues.
3. Source Quench (Type 4): Sent by routers to inform the sender to reduce the rate of packets being sent, usually due to network congestion.
4. Redirect (Type 5): Informs the sender that there is a better route for the specified destination.
5. Time Exceeded (Type 11): Indicates that the TTL (Time to Live) field of a packet has reached zero or that a reassembly timeout has occurred.
6. Parameter Problem (Type 12): Indicates that a problem with the IP header or options field has been detected.
7. Timestamp Request/Timestamp Reply (Type 13/Type 14): Used for timestamping purposes.

- -A INPUT appends the rule to the INPUT chain of the iptables firewall.
- -p icmp specifies the protocol (ICMP).
- --icmp-type echo-request targets ICMP echo requests, which are used in ping tests.
- -j DROP instructs iptables to drop any packets matching the specified criteria

Blocking ICMP Requests

Blocking Ping ICMP Requests:

Different types of ICMP include Echo Request, Echo Reply, Destination Unreachable, Time Exceeded, etc.

To block Ping ICMP requests to Linux systems, you can use the following rule:

iptables -A INPUT -p icmp --icmp-type 8 -j DROP

This rule blocks ICMP Echo Request packets (Ping) directed at the Linux system, enhancing security by preventing external hosts from pinging the system.

By implementing these iptables rules, you can effectively manage network traffic, enhance security, and control access to specific services on your Linux system.