Aim: Setting up personal Firewall using iptables.

Theory: -

All packets inspected by iptables pass through a sequence of built-in tables (queues) for processing. Each of these queues is dedicated to a particular type of packet activity and is controlled by an associated packet transformation/filtering chain.

1. Filter Table

Filter is default table for iptables.

Iptables's filter table has the following built-in chains.

- INPUT chain – Incoming to firewall. For packets coming to the local server.
- OUTPUT chain – Outgoing from firewall. For packets generated locally and going out of the local server.
- FORWARD chain – Packet for another NIC on the local server. For packets routed through the local server.

2. NAT Table

This table is consulted when a packet that creates a new connection is encountered.
Iptable's NAT table has the following built-in chains.

- PREROUTING chain – Alters packets before routing. i.e Packet translation happens immediately after the packet comes to the system (and before routing). This helps to translate the destination ip address of the packets to something that matches the routing on the local server. This is used for DNAT (destination NAT).
- POSTROUTING chain – Alters packets after routing. i.e Packet translation happens when the packets are leaving the system. This helps to translate the source ip address of the packets to something that might match the routing on the destination server. This is used for SNAT (source NAT).
- OUTPUT chain – NAT for locally generated packets on the firewall.

3. Mangle Table

Iptables's Mangle table is for specialized packet alteration. This alters QOS bits in the TCP header. Mangle table has the following built-in chains.

- PREROUTING chain
- OUTPUT chain
- FORWARD chain
- INPUT chain
- POSTROUTING chain

4. Raw Table

Iptable's Raw table is for configuration exemptions. Raw table has the following built-in chains.

- PREROUTING chain
- OUTPUT chain

5. Security Table

This table is used for Mandatory Access Control (MAC) networking rules, such as those enabled by the SECMARK and CONNSECMARK targets. Mandatory Access Control is implemented by Linux Security Modules such as SELinux. The security table is called after the filter table, allowing any Discretionary Access Control (DAC) rules in the filter table to take effect before MAC rules. This table provides the following built-in chains: INPUT (for packets coming into the box itself), OUTPUT (for altering locally-generated packets before routing), and FORWARD (for altering packets being routed through the box).

Chains
Tables consist of *chains*; Rules are combined into different chains. The kernel uses chains to manage packets it receives and sends out. A chain is simply a checklist of rules which are lists of rules which are followed in order. The rules operate with an if-then -else structure.

Input – This chain is used to control the behaviour for incoming connections. For example, if a user attempts to SSH into your PC/server, iptables will attempt to match the IP address and port to a rule in the input chain.

Forward – This chain is used for incoming connections that aren't actually being delivered locally. Think of a router – data is always being sent to it but rarely actually destined for the router itself; the data is just forwarded to its target.

Output – This chain is used for outgoing connections. For example, if you try to ping howtogeek.com, iptables will check its output chain to see what the rules are regarding ping and howtogeek.com before making a decision to allow or deny the connection attempt.

Targets:
ACCEPT: Allow packet to pass through the firewall.
DROP: Deny access by the packet.
REJECT: Deny access and notify the server.
QUEUE: Send packets to user space.
RETURN: jump to the end of the chain and let the default target process it

iptables command Switch	Description
-L	Listing of rules present in the chain
-n	Numeric output of addresses and ports
-v	Displays the rules in verbose mode
-t <-table->	If you don't specify a table, then the filter table is assumed. As discussed before, the possible built-in tables include: filter, nat, mangle
-j <target>	Jump to the specified target chain when the packet matches the current rule.
-A	Append rule to end of a chain
-F	Flush. Deletes all the rules in the selected table
-p <protocol-type>	Match protocol. Types include, icmp, tcp, udp, and all
-s <ip-address>	Match source IP address
-d <ip-address>	Match destination IP address
-i <interface-name>	Match "input" interface on which the packet enters.
-o <interface-name>	Match "output" interface on which the packet exits

Steps: -

1. Get root access: $ sudo su root

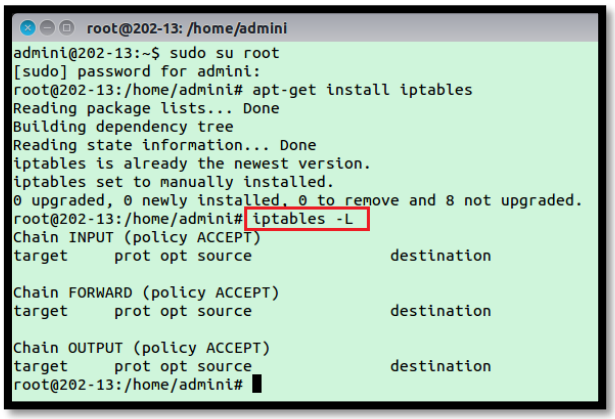
2. # apt-get install iptables

Commands: -

1. To see the list of iptables rules

iptables -L

Initially it is empty

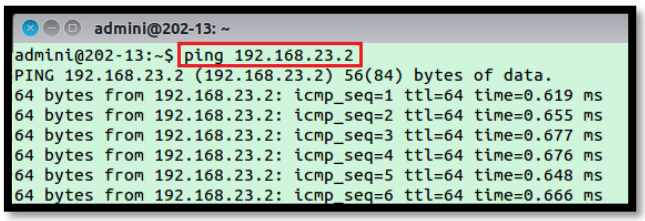


2. To block outgoing traffic to a particular destination for a specific protocol from a machine

Syntax: iptables -I OUTPUT -s <your ip> -d <neighbour ip> -p <protocol> -j <action>

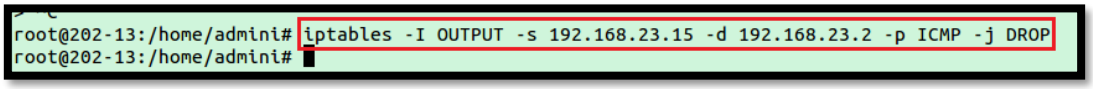
Open one terminal and Ping the neighbour. Let the ping run.

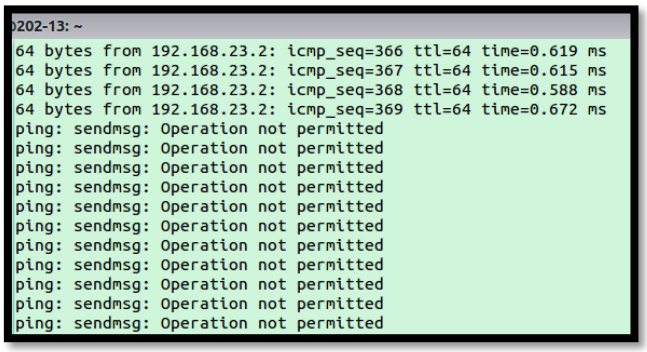
#ping 192.168.208.6



Open another terminal and run the iptables command

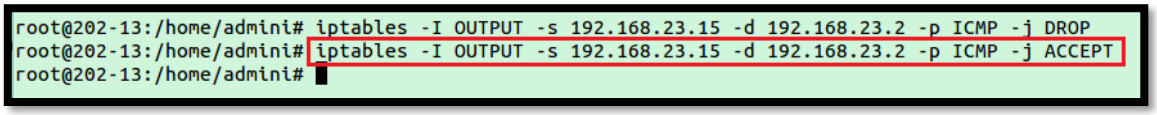
iptables -I OUTPUT -s 192.168.208.18 -d 192.168.208.6 -p icmp -j DROP

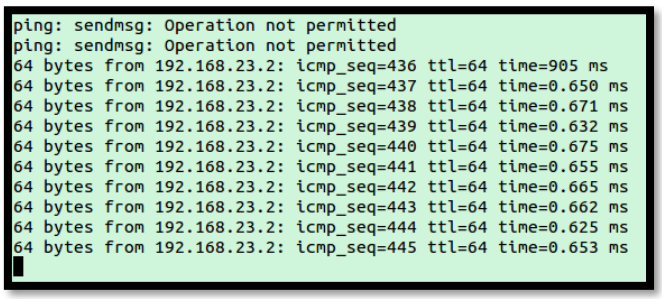




2. To allow outgoing traffic to a particular destination for a specific protocol from a machine

iptables -I OUTPUT -s 192.168.208.18 -d 192.168.208.6 -p icmp -j ACCEPT





3. To block outgoing traffic to a particular destination for a specific protocol from a machine for sometime

iptables -I OUTPUT -s 192.168.208.18 -d 192.168.208.6 -p icmp -j REJECT

```
root@202-13:/home/admini# iptables -I OUTPUT -s 192.168.23.15 -d 192.168.23.2 -p ICMP -j DROP
root@202-13:/home/admini# iptables -I OUTPUT -s 192.168.23.15 -d 192.168.23.2 -p ICMP -j ACCEPT
root@202-13:/home/admini# iptables -I OUTPUT -s 192.168.23.15 -d 192.168.23.2 -p ICMP -j REJECT
root@202-13:/home/admini#
```

```
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
From 192.168.23.15 icmp_seq=1 Destination Port Unreachable
```

Allow the traffic again by using ACCEPT instead of REJECT

4. To block incoming traffic from particular destination for a specific protocol to machine

Syntax: iptables -I INPUT -s <neighbour ip> -d <firewall ip> -p <protocol> -j <action>

Open one terminal and Ping the neighbour. Let the ping run.

#ping 192.168.208.6

Open another terminal and run the iptables command

iptables -I INPUT -s 192.168.208.6 -d 192.168.208.18 -p icmp -j DROP

```
root@202-13:/home/admini# iptables -I OUTPUT -s 192.168.23.15 -d 192.168.23.2 -p ICMP -j ACCEPT
root@202-13:/home/admini# iptables -I INPUT -s 192.168.23.2 -d 192.168.23.15 -p ICMP -j DROP
root@202-13:/home/admini#
```

```
64 bytes from 192.168.23.15: icmp_seq=123 ttl=64 time=0.669 ms
64 bytes from 192.168.23.15: icmp_seq=124 ttl=64 time=0.649 ms
64 bytes from 192.168.23.15: icmp_seq=125 ttl=64 time=0.661 ms
64 bytes from 192.168.23.15: icmp_seq=126 ttl=64 time=0.709 ms
64 bytes from 192.168.23.15: icmp_seq=127 ttl=64 time=0.702 ms
64 bytes from 192.168.23.15: icmp_seq=128 ttl=64 time=0.661 ms
64 bytes from 192.168.23.15: icmp_seq=129 ttl=64 time=0.632 ms
64 bytes from 192.168.23.15: icmp_seq=130 ttl=64 time=0.681 ms
64 bytes from 192.168.23.15: icmp_seq=131 ttl=64 time=0.689 ms
64 bytes from 192.168.23.15: icmp_seq=132 ttl=64 time=0.655 ms
64 bytes from 192.168.23.15: icmp_seq=133 ttl=64 time=0.647 ms
64 bytes from 192.168.23.15: icmp_seq=134 ttl=64 time=0.673 ms
64 bytes from 192.168.23.15: icmp_seq=135 ttl=64 time=0.687 ms
64 bytes from 192.168.23.15: icmp_seq=136 ttl=64 time=0.676 ms
64 bytes from 192.168.23.15: icmp_seq=137 ttl=64 time=0.621 ms
64 bytes from 192.168.23.15: icmp_seq=138 ttl=64 time=0.666 ms
64 bytes from 192.168.23.15: icmp_seq=139 ttl=64 time=0.699 ms
64 bytes from 192.168.23.15: icmp_seq=140 ttl=64 time=0.613 ms
64 bytes from 192.168.23.15: icmp_seq=141 ttl=64 time=0.521 ms
64 bytes from 192.168.23.15: icmp_seq=142 ttl=64 time=0.664 ms
64 bytes from 192.168.23.15: icmp_seq=143 ttl=64 time=0.678 ms
64 bytes from 192.168.23.15: icmp_seq=144 ttl=64 time=0.644 ms
64 bytes from 192.168.23.15: icmp_seq=145 ttl=64 time=0.676 ms
```

5. To allow incoming traffic from particular destination for a specific protocol to machine

Syntax: iptables -I INPUT -s <neighbour ip> -d <firewall ip> -p <protocol> -j <action>

Open another terminal and run the iptables command

iptables -I INPUT -s 192.168.208.6 -d 192.168.208.18 -p icmp -j ACCEPT

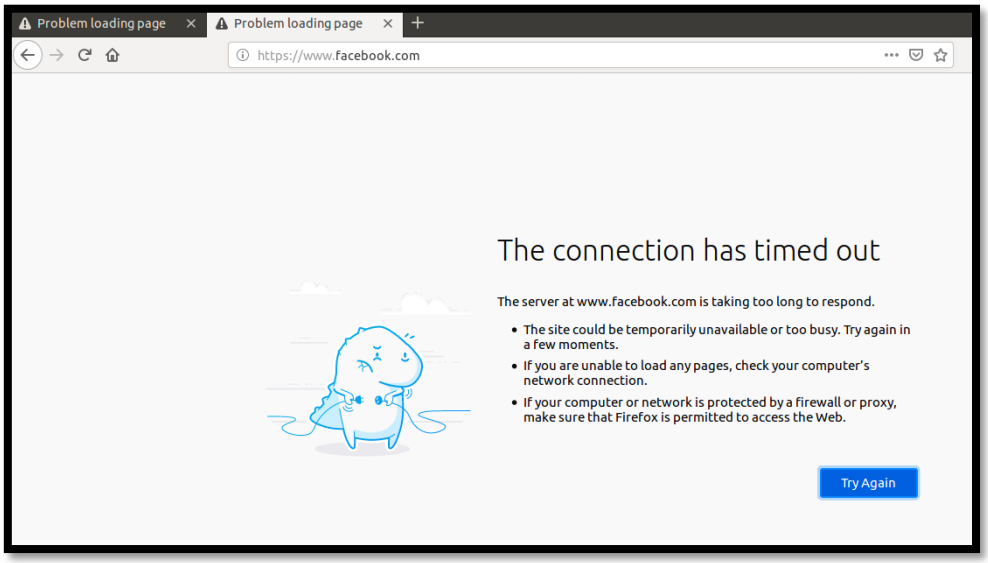
Check the ping status on the other terminal

6. To clear the rules in iptables

iptables -F

7. To block specific URL from machine
 # iptables -t filter -I OUTPUT -m string --string facebook.com -j REJECT --algo kmp

```
root@202-13:/home/admini# iptables -t filter -I INPUT -m string --string facebook.com -j REJECT --algo kmp
root@202-13:/home/admini#
```



It will block facebook.com by performing string matching. The algorithm used for string matching is KMP.
If we change target *from REJECT to ACCEPT*, the site can be visited again.

```
root@202-13:/home/admini# iptables -t filter -I INPUT -m string --string facebook.com -j REJECT --algo kmp
root@202-13:/home/admini# iptables -t filter -I OUTPUT -m string --string facebook.com -j REJECT --algo bm
root@202-13:/home/admini# iptables -t filter -I OUTPUT -m string --string facebook.com -j ACCEPT --algo bm
root@202-13:/home/admini#
```

Observations:

1. In case of OUTPUT chain, for DROP and REJECT chain, at source machine we get two different messages.
   For DROP – 'Operation Not Permitted'. Here No acknowledgement is provided.
   For REJECT – 'Destination Port Unreachable'. Here acknowledgement is given.
2. In case of INPUT chain for DROP and REJECT chain at source machine we get two different responses as follows:
   For DROP – No message. Here No acknowledgement is provided.
   For REJECT – 'Destination Port Unreachable'. Here acknowledgement is given.