Listing rules

- `-L` is for table style display which is much harder to read
- `-S` is command style display which is easier to understand
- `-t` is tables (filter raw, nat, mangle, security)
- `--line-numbers` shows line numbers per chain

```
iptables -L --line-numbers
iptables -S

iptables -L -t nat
iptables -S -t nat

iptables -L -t raw
iptables -S -t raw

iptables -S POSTROUTING -t nat
iptables -S INPUT

iptables -L -v
```

Creating default rules

```
iptables -P INPUT  ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

Flush rules by using `iptables -F` this will flush everything except the default rules

Extending rules

- `-A` appends rule to the end of the table
- `-I <rule no>` inserts a new rule at the line number
- `-m` module
- `-j` target jump what to do if the condition is met for the packet (ACCEPT, DROP, RETURN)

Allows all packets which have already established connection through even after adding new rules `iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT`

Accept all connections for port 22 and 80

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

- `-p` protocol (UDP, TCP, ICMP, ALL)
- `--dport` destination port
- `-i` in interface
- `-o` out interface

Accept all loopback connection and put it on line 1 INPUT chain rule `iptables -I INPUT 1 -i lo -j ACCEPT`

Add the drop rule for INPUT chain at the last line `iptables -A INPUT -j DROP`

Now you need to delete and recreate the last drop rule each time or find the second to the last line number and apply it ther

```
iptables -D INPUT -j DROP
iptables -A INPUT new_rule_here
iptables -A INPUT -j DROP
#or
iptables -I INPUT 4 new_rule_here
```

Save new rules to persist during reboots

```
sudo apt-get update
sudo apt-get install iptables-persistent
sudo invoke-rc.d iptables-persistent save
```

New Chains

```
  iptables -N DOCKER
  iptables -N TCP
```

Deleting

Delete by Specification:

From `iptables -S` if you have `-D INPUT -m conntrack --ctstate INVALID -j DROP`

```
  sudo iptables -D INPUT -m conntrack --ctstate INVALID -j DROP
```

Delete by chain and line number:

From `iptables -L --line-number`

```
  sudo iptables -D INPUT 3
```

Delete a chain:

```
  sudo iptables -F INPUT
```

Delete all chains:

```
  sudo iptables -F
```

Delete all chains and rules and accept all connections

```
  sudo iptables -P INPUT ACCEPT
  sudo iptables -P FORWARD ACCEPT
  sudo iptables -P OUTPUT ACCEPT

  sudo iptables -t nat -F
  sudo iptables -t mangle -F
  sudo iptables -F
  sudo iptables -X
```

Example Iptables automatically configured by docker

```
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-N DOCKER
-N DOCKER-INGRESS
-N DOCKER-ISOLATION
-A FORWARD -j DOCKER-INGRESS
-A FORWARD -j DOCKER-ISOLATION
-A FORWARD -o docker0 -j DOCKER
-A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i docker0 ! -o docker0 -j ACCEPT
-A FORWARD -i docker0 -o docker0 -j ACCEPT
-A FORWARD -o docker_gwbridge -j DOCKER
-A FORWARD -o docker_gwbridge -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i docker_gwbridge ! -o docker_gwbridge -j ACCEPT
-A FORWARD -i docker_gwbridge -o docker_gwbridge -j DROP
-A DOCKER-INGRESS -p tcp -m tcp --dport 17017 -j ACCEPT
-A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 17017
-j ACCEPT
-A DOCKER-INGRESS -p tcp -m tcp --dport 17147 -j ACCEPT
-A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 17147
-j ACCEPT
-A DOCKER-INGRESS -p tcp -m tcp --dport 80 -j ACCEPT
-A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 80 -j
ACCEPT
-A DOCKER-INGRESS -p tcp -m tcp --dport 5151 -j ACCEPT
-A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 5151
-j ACCEPT
-A DOCKER-INGRESS -j RETURN
-A DOCKER-ISOLATION -i docker_gwbridge -o docker0 -j DROP
-A DOCKER-ISOLATION -i docker0 -o docker_gwbridge -j DROP
-A DOCKER-ISOLATION -j RETURN
```