

Enable UFW

To turn UFW on with the default set of rules:

```
sudo ufw enable
```

To check the status of UFW:

```
sudo ufw status verbose
```

The output should be like this:

```
youruser@yourcomputer:~$ sudo ufw status verbose
[sudo] password for youruser:
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing)
New profiles: skip
youruser@yourcomputer:~$
```

Note that by default, deny is being applied to incoming. There are exceptions, which can be found in the output of this command:

```
sudo ufw show raw
```

You can also read the rules files in `/etc/ufw` (the files whose names end with `.rules`).

Disable UFW

To disable ufw use:

```
sudo ufw disable
```

Allow and Deny (specific rules)

Allow

```
sudo ufw allow <port>/<optional: protocol>
```

example: To allow incoming tcp and udp packet on port 53

```
•sudo ufw allow 53
```

example: To allow incoming tcp packets on port 53

```
•sudo ufw allow 53/tcp
```

example: To allow incoming udp packets on port 53

```
•sudo ufw allow 53/udp
```

Deny

```
sudo ufw deny <port>/<optional: protocol>
```

example: To deny tcp and udp packets on port 53

```
•sudo ufw deny 53
```

example: To deny incoming tcp packets on port 53

```
•sudo ufw deny 53/tcp
```

example: To deny incoming udp packets on port 53

```
•sudo ufw deny 53/udp
```

Delete Existing Rule

To delete a rule, simply prefix the original rule with delete. For example, if the original rule was:

```
ufw deny 80/tcp
```

Use this to delete it:

```
sudo ufw delete deny 80/tcp
```

Services

You can also allow or deny by service name since ufw reads from /etc/services To see get a list of services:

```
less /etc/services
```

Allow by Service Name

```
sudo ufw allow <service name>
```

example: to allow ssh by name

```
•sudo ufw allow ssh
```

Deny by Service Name

```
sudo ufw deny <service name>
```

example: to deny ssh by name

```
•sudo ufw deny ssh
```

Status



Checking the status of ufw will tell you if ufw is enabled or disabled and also list the current ufw rules that are applied to your iptables.

To check the status of ufw:

```
sudo ufw status
```

```
Firewall loaded
```

```
To
```

```
Action From
```

```
--
22:tcp          DENY    192.168.0.1
22:udp          DENY    192.168.0.1
22:tcp          DENY    192.168.0.7
22:udp          DENY    192.168.0.7
22:tcp          ALLOW   192.168.0.0/24
22:udp          ALLOW   192.168.0.0/24
```

if ufw was not enabled the output would be:

```
sudo ufw status
Status: inactive
```

Logging

To enable logging use:

```
sudo ufw logging on
```

To disable logging use:

```
sudo ufw logging off
```

Advanced Syntax

You can also use a fuller syntax, specifying the source and destination addresses, ports and protocols.

Allow Access

This section shows how to allow specific access.

Allow by Specific IP

```
sudo ufw allow from <ip address>
```

example: To allow packets from 207.46.232.182:

```
•sudo ufw allow from 207.46.232.182
```

Allow by Subnet

You may use a net mask :

```
sudo ufw allow from 192.168.1.0/24
```

Allow by specific port and IP address

```
sudo ufw allow from <target> to <destination> port <port number>
```

example: allow IP address 192.168.0.4 access to port 22 for all protocols

```
•sudo ufw allow from 192.168.0.4 to any port 22
```

Allow by specific port, IP address and protocol

```
sudo ufw allow from <target> to <destination> port <port number> proto <protocol name>
```

example: allow IP address 192.168.0.4 access to port 22 using TCP

```
•sudo ufw allow from 192.168.0.4 to any port 22 proto tcp
```

Enable PING

Note: Security by obscurity may be of very little actual benefit with modern cracker scripts.**By default, UFW allows ping requests.** You may find you wish to leave (icmp) ping requests enabled to diagnose networking problems.

In order to disable ping (icmp) requests, you need to edit **/etc/ufw/before.rules** and remove the following lines:

```
# ok icmp codes
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type source-quench -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT
```

or change the "ACCEPT" to "DROP"

```
# ok icmp codes
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j DROP
-A ufw-before-input -p icmp --icmp-type source-quench -j DROP
-A ufw-before-input -p icmp --icmp-type time-exceeded -j DROP
-A ufw-before-input -p icmp --icmp-type parameter-problem -j DROP
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP
```

Deny Access

Deny by specific IP

```
sudo ufw deny from <ip address>
```

example: To block packets from 207.46.232.182:

```
•sudo ufw deny from 207.46.232.182
```

Deny by specific port and IP address

```
sudo ufw deny from <ip address> to <protocol> port <port number>
```

example: deny ip address 192.168.0.1 access to port 22 for all protocols

```
•sudo ufw deny from 192.168.0.1 to any port 22
```

Working with numbered rules

Listing rules with a reference number

You may use status numbered to show the order and id number of rules:

```
sudo ufw status numbered
```

Editing numbered rules

Delete numbered rule

You may then delete rules using the number. This will delete the first rule and rules will shift up to fill in the list.

```
sudo ufw delete 1
```

Insert numbered rule

```
sudo ufw insert 1 allow from <ip address>
```

Advanced Example

Scenario: You want to block access to port 22 from 192.168.0.1 and 192.168.0.7 but allow all other 192.168.0.x IPs to have access to port 22 using tcp

```
sudo ufw deny from 192.168.0.1 to any port 22
sudo ufw deny from 192.168.0.7 to any port 22
sudo ufw allow from 192.168.0.0/24 to any port 22 proto tcp
```



This puts the specific rules first and the generic second. Once a rule is matched the others will not be evaluated (see manual below) so you must put the specific rules first. **As rules change you may need to delete old rules to ensure that new rules are put in the proper order.**

To check your rules orders you can check the status; for the scenario the output below is the desired output for the rules to work properly

```
sudo ufw status
Firewall loaded

To                                Action  From
--                                -
22:tcp                           DENY    192.168.0.1
22:udp                           DENY    192.168.0.1
22:tcp                           DENY    192.168.0.7
22:udp                           DENY    192.168.0.7
22:tcp                           ALLOW   192.168.0.0/24
```

Scenario change: You want to block access to port 22 to 192.168.0.3 as well as 192.168.0.1 and 192.168.0.7.

```
sudo ufw delete allow from 192.168.0.0/24 to any port 22
sudo ufw status
Firewall loaded
```

```
To                                Action  From
--                                -
```

```
22:tcp          DENY    192.168.0.1
22:udp          DENY    192.168.0.1
22:tcp          DENY    192.168.0.7
22:udp          DENY    192.168.0.7
```

```
sudo ufw deny 192.168.0.3 to any port 22
sudo ufw allow 192.168.0.0/24 to any port 22 proto tcp
sudo ufw status
```

Firewall loaded

To	Action	From
--	-----	----
22:tcp	DENY	192.168.0.1
22:udp	DENY	192.168.0.1
22:tcp	DENY	192.168.0.7
22:udp	DENY	192.168.0.7
22:tcp	DENY	192.168.0.3
22:udp	DENY	192.168.0.3
22:tcp	ALLOW	192.168.0.0/24



If you simply add the deny rule the allow would have been above it and been applied instead of the deny