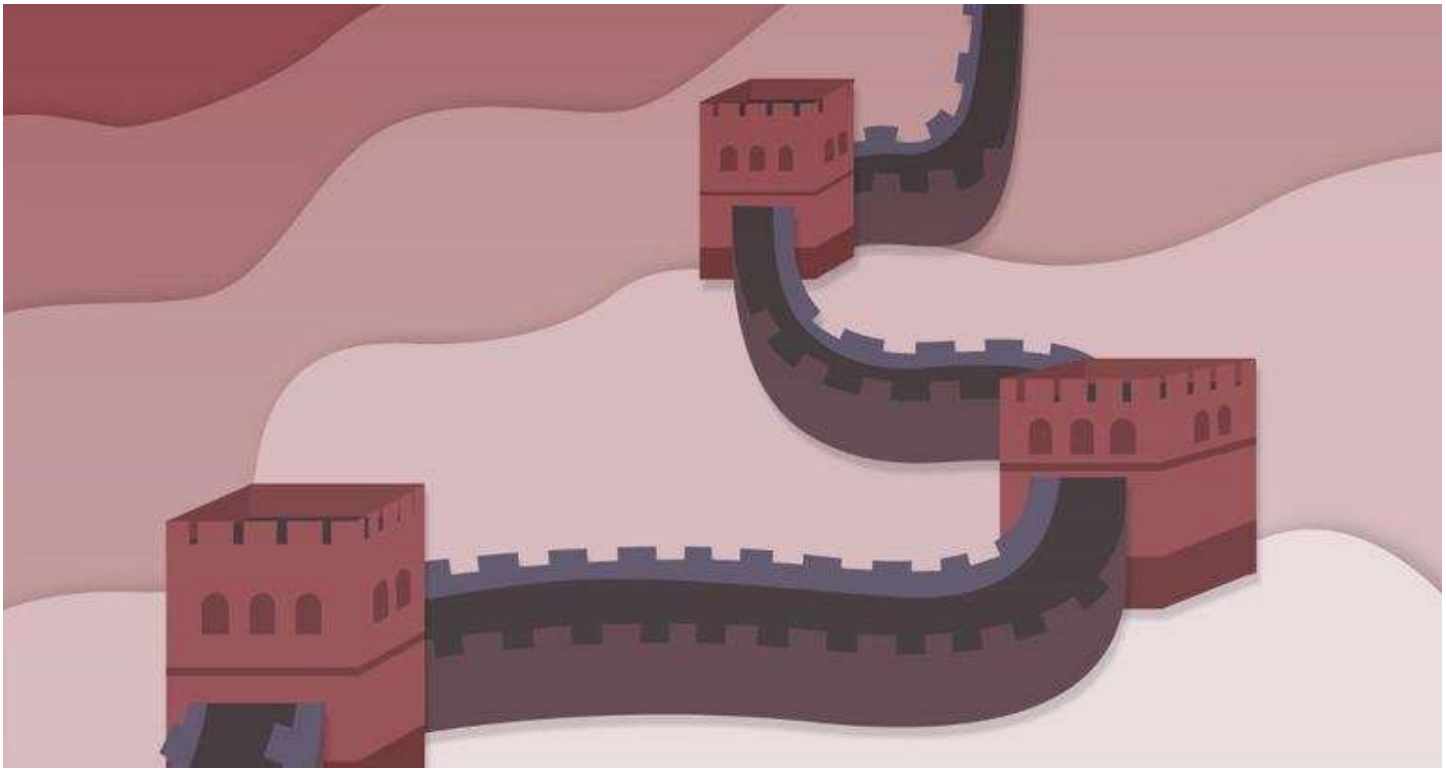
# How to Set Up a Firewall with UFW on Ubuntu 18.04

Updated Feb 15, 2019 • 8 min read



A properly configured firewall is one of the most important aspects of overall system security. By default, Ubuntu comes with a firewall configuration tool called UFW (Uncomplicated Firewall). UFW is a user-friendly front-end for managing iptables firewall rules and its main goal is to make managing iptables easier or as the name says uncomplicated.

## Prerequisites

Before you start with this tutorial, make sure you are logged into your server with a user account with sudo privileges or with the root user. The best practice is to run administrative commands as a sudo user instead of root. If you don't have a sudo user on your Ubuntu system you can create one by following these instructions .

## Install UFW

Uncomplicated Firewall should be installed by default in Ubuntu 18.04, but if it is not installed on your system, you can install the package by typing:

```
$ sudo apt install ufw
```

## Check UFW Status

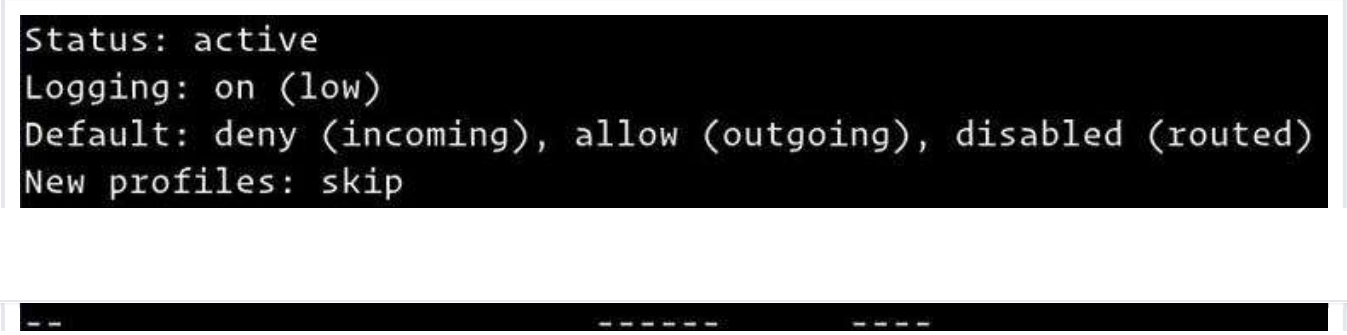
Once the installation is completed you can check the status of UFW with the following command:

```
$ sudo ufw status verbose
```

UFW is disabled by default. If you never activated UFW before, the output will look like this:

```
Output
Status: inactive
```

If UFW is activated, the output will look similar to the following:



```
22/tcp                     ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
```

## UFW Default Policies

By default, UFW will block all of the incoming connections and allow all outbound connections. This means that anyone trying to access your server will not be able to connect unless you specifically open the port, while all applications and services running on your server will be able to access the outside world.

The default polices are defined in the `/etc/default/ufw` file and can be changed using the `sudo ufw default <policy> <chain>` command.

Firewall policies are the foundation for building more detailed and user-defined rules. In most cases, the initial UFW Default Policies are a good starting point.

## Application Profiles

When installing a package with the `apt` command it will add an application profile to `/etc/ufw/applications.d` directory. The profile describes the service and contains UFW settings.

You can list all application profiles available on your server by typing:

```
$ sudo ufw app list
```

Depending on the packages installed on your system the output will look similar to the following:

```
Output
Available applications:
  Dovecot IMAP
  Dovecot POP3
  Dovecot Secure IMAP
  Dovecot Secure POP3
```

```
  Nginx HTTPS
  OpenSSH
  Postfix
  Postfix SMTPS
  Postfix Submission
```

To find more information about a specific profile and included rules, use the following command:

```
$ sudo ufw app info 'Nginx Full'
```

```
Output
Profile: Nginx Full
Title: Web Server (Nginx, HTTP + HTTPS)
Description: Small, but very powerful and efficient web server

Ports:
  80,443/tcp
```

As you can see from the output above the 'Nginx Full' profile opens port `80` and `443`.

## Allow SSH Connections

Before enabling the UFW firewall we need to add a rule which will allow incoming SSH connections. If you're connecting to your server from a remote location, which is almost always the case and you enable the UFW firewall before explicitly allow incoming SSH connections you will no longer be able to connect to your Ubuntu server.

To configure your UFW firewall to allow incoming SSH connections, type the following command:

```
$ sudo ufw allow ssh
```

```
Output
Rules updated
```

If you changed the SSH port to a custom port instead of the port 22, you will need to open that port.

For example, if your ssh daemon listens on port `4422`, then you can use the following command to allow connections on that port:

```
$ sudo ufw allow 4422/tcp
```

## Enable UFW

Now that your UFW firewall is configured to allow incoming SSH connections, we can enable it by typing:

```
$ sudo ufw enable
```

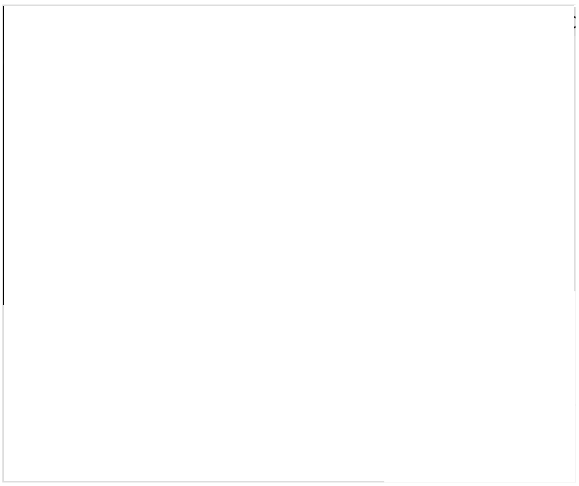
```
Output
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

You will be warned that enabling the firewall may disrupt existing ssh connections, just type `y` and hit `Enter`.

## Allow connections on other ports

Depending on the applications that run on your server and your specific needs you'll also need to allow incoming access to some other ports.

Below we will show you a few examples on how to allow incoming connections to some of the most common services:



## Open port 80 - HTTP

HTTP connections can be allowed with the following command:

```
$ sudo ufw allow http
```

instead of http you can use the port number, 80:

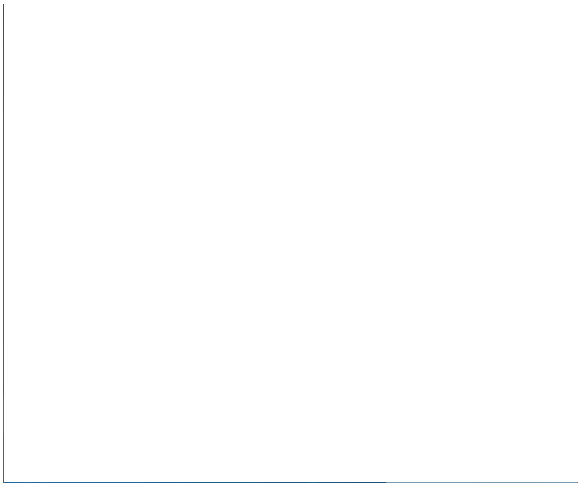
```
$ sudo ufw allow 80/tcp
```

or you can use the application profile, in this case, 'Nginx HTTP':

```
$ sudo ufw allow 'Nginx HTTP'
```

## Open port 443 - HTTPS

HTTP connections can be allowed with the following command:



```
$ sudo ufw allow https
```

To achieve the same instead of `https` profile you can use the port number, `443` :

```
$ sudo ufw allow 443/tcp
```

or you can use the application profile, 'Nginx HTTPS':

```
$ sudo ufw allow 'Nginx HTTPS'
```

### Open port 8080

If you run Tomcat or any other application that listens on port `8080` to allow incoming connections type:

```
$ sudo ufw allow 8080/tcp
```

## Allow Port Ranges

Instead of allowing access to single ports UFW allows us to allow access to port ranges. When allowing port ranges with UFW, you must specify the protocol, either `tcp` or `udp` . For example, if you want to allow ports from `7100` to `7200` on both `tcp` and `udp` then run the following command:

```
$ sudo ufw allow 7100:7200/tcp
$ sudo ufw allow 7100:7200/udp
```

## Allow Specific IP Addresses

To allow access on all ports from your home machine with IP address of 64.63.62.61, specify `from` followed by the IP address you want to whitelist:

```
$ sudo ufw allow from 64.63.62.61
```

## Allow Specific IP Addresses on Specific port

To allow access on a specific port let's say port 22 from your work machine with IP address of 64.63.62.61, use `to any port` followed by the port number:

```
$ sudo ufw allow from 64.63.62.61 to any port 22
```

## Allow Subnets

The command for allowing connection to a subnet of IP addresses is the same as when using a single IP address, the only difference is that you need to specify the netmask. For example, if you want to allow access for IP addresses ranging from 192.168.1.1 to 192.168.1.254 to port 3360 (MySQL ) you can use this command:

```
$ sudo ufw allow from 192.168.1.0/24 to any port 3306
```

## Allow Connections to a Specific Network Interface

To allow access on a specific port let's say port 3360 only to specific network interface `eth2` , then you need to specify `allow in on` and the name of the network interface:

```
$ sudo ufw allow in on eth2 to any port 3306
```

The default policy for all incoming connections is set to `deny` and if you haven't changed it, UFW will block all incoming connection unless you specifically open the connection.

Let's say you opened the ports `80` and `443` and your server is under attack from the `23.24.25.0/24` network. To deny all connections from `23.24.25.0/24` you can use the following command:

```
$ sudo ufw deny from 23.24.25.0/24
```

If you only want to deny access to ports `80` and `443` from `23.24.25.0/24` you can use the following command:

```
$ sudo ufw deny from 23.24.25.0/24 to any port 80
$ sudo ufw deny from 23.24.25.0/24 to any port 443
```

Writing deny rules is the same as writing allow rules, you only need to replace `allow` with `deny`.

## Delete UFW Rules

There are two different ways to delete UFW rules, by rule number and by specifying the actual rule.

Deleting UFW rules by rule number is easier especially if you are new to UFW. To delete a rule by a rule number first you need to find the number of the rule you want to delete, you can do that with the following command:

```
$ sudo ufw status numbered
```

```
Output
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 8080/tcp                   ALLOW IN    Anywhere
```

To delete rule number 3, the rule that allows connections to port 8080, use the following command:

The second method is to delete a rule by specifying the actual rule, for example if you added a rule to open port `8069` you can delete it with:

```
$ sudo ufw delete allow 8069
```

## Disable UFW

If for any reason you want to stop UFW and deactivate all the rules you can use:

```
$ sudo ufw disable
```

Later if you want to re-enable UTF and activate all rules just type:

```
$ sudo ufw enable
```

## Reset UFW

Resetting UFW will disable UFW, and delete all active rules. This is helpful if you want to revert all of your changes and start fresh.

To reset UFW simply type in the following command:

```
$ sudo ufw reset
```

## Conclusion

You have learned how to install and configure UFW firewall on your Ubuntu 18.04 server. Be sure to allow all incoming connections that are necessary for proper functioning of your system, while limiting all unnecessary connections.

If you have questions, feel free to leave a comment below.

ufw firewall iptables ubuntu security

If you like our content, please consider buying us a coffee.
Thank you for your support!

BUY ME A COFFEE

Sign up to our newsletter and get our latest tutorials and news straight to your mailbox.

Your email...

We'll never share your email address or spam you.

## Related Articles

MAY 8, 2020

### How to Set Up a Firewall with UFW on Ubuntu 20.04


OCT 18, 2018

### How to Set Up a Firewall with UFW on Debian 9


APR 11, 2020

## How to Set Up a Firewall with UFW on Debian 10


Show comments (3)

© 2022 Linuxize.com