How to Configure Your Ubuntu Firewall with UFW





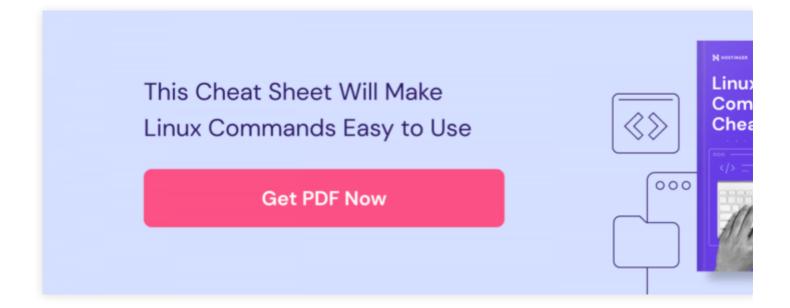






<u>Server security</u> is something that should not be taken lightly, in the age when cyber-crimes are on the headling is always advisable to take security measures to add an additional level of security to your server.

By protecting our server, we also protect our data. An efficient way to do this is to configure a Firewall in Ubur supervise the incoming and outgoing connections made to our server. In this tutorial, we will teach you how to



Using the Ubuntu Firewall to Protect Your Server

Configuring the Firewall with UFW on Ubuntu 18.04

Setting Firewall Rules on Ubuntu 18.04 with UFW

Open and Close Ports with UFW

Working with Services on Ubuntu Firewall

Deny or Allow IP Address Connections

Deleting a Specific Rule on Ubuntu Firewall

Using the Ubuntu Firewall to Protect Your Server

Once we have access to our server, we will enable UFW with the following command:

sudo ufw enable

If you receive the command not found error, install UFW with the following command.

sudo apt-get install ufw

Then, we have to check the UFW status.

sudo ufw status

As we can see, UFW is now enabled.

By default, UFW denies all incoming connections and allows all outgoing connections. For many users this co services or applications, we have to establish some rules.

Your virtual hosting server, your rules. Get all the resources you ne project.



Setting Firewall Rules on Ubuntu 18.04 with UFW

A Firewall rule is an instruction that shapes how a Firewall works. The rules define which connections are acc

Next, we will configure some Firewall rules using UFW:

Open and Close Ports with UFW

The ports are connection interfaces used by applications to establish a connection to a server.

With UFW it is quite easy to open or close them as we see fit. To open a port, we need to run this command:

sudo ufw allow [port/protocol]

To open ports the command would look like the following:

sudo ufw allow 300:310/tcp

Or, to deny them:

sudo ufw deny 300:310/tcp

Working with Services on Ubuntu Firewall

There are some network services that UFW can enforce. The way to manage them is to know the port they us

For example, HTTP requires that port 80 is available and for HTTPS port 443 is available.

So, we need to run this command for HTTP:

sudo ufw allow http

The command run is equivalent to enabling port 80 as previously explained.

So, we only need to know the ports used by the network services.

Deny or Allow IP Address Connections

It is also possible to deny access for a specific IP address.

To do this, we have to execute the following command:

sudo ufw deny from IPADRESS

For example:

sudo ufw deny from 192.168.1.2

Or on the contrary, if we want to allow access to that IP address.

sudo ufw allow from 192.168.1.3

Another thing we can do is specify if we want an IP address to be able to connect only to a specific port.

```
angelo@ubuntu:~$ sudo ufw delete 4
Deleting:
deny 56/tcp
Proceed with operation (y|n)? y
Rule deleted (v6)
angelo@ubuntu:~$ sudo ufw status numbered
Status: active
    Τo
                                Action
                                            From
                                DENY IN
[ 1] 56/tcp
                                            Anywhere
[ 2] Anywhere
                                DENY IN
                                            192.168.1.2
[ 3] Anywhere
                                ALLOW IN
                                             192.168.1.3
angelo@ubuntu:~$
```

After that, we delete the rule that we want. For example, We will delete rule number four.

```
sudo ufw delete 4
```

That's all the basic functions you should be aware of! You're ready to configure your server's security the way the UFW manual. You can access it with the following command:

```
sudo ufw -help
```

Conclusion

The process of configuring a Firewall in Ubuntu 18.04 is easy to do thanks to UFW. However, the application I of our server. Here you learned all the basics that shouldn't be skipped. We hope you found this tutorial usefu