TUTORIAL

How To Set Up a Firewall with UFW on Ubuntu 18.04

Ubuntu | Security | Networking | Firewall | Ubuntu 18.04

By Brian Boucheron

Posted July 5, 2018 627.4k

English

Not using Ubuntu 18.04?
Choose a different version or distribution.

Ubuntu 18.04

A previous version of this tutorial was written by Hazel Virdó

Introduction

UFW, or Uncomplicated Firewall, is an interface to `iptables` that is geared towards simplifying the process of configuring a firewall. While `iptables` is a solid and flexible tool, it can be difficult for beginners to learn how to use it to properly configure a firewall. If you're looking to get started securing your network, and you're not sure which tool to use, UFW may be the right choice for you.

This tutorial will show you how to set up a firewall with UFW on Ubuntu 18.04.

Prerequisites

To follow this tutorial, you will need:

- One Ubuntu 18.04 server with a sudo non-root user, which you can set up by following Steps 1–3 in the Initial Server Setup with Ubuntu 18.04 tutorial.

UFW is installed by default on Ubuntu. If it has been uninstalled for some reason, you can install it with `sudo apt install ufw`.

Step 1 — Using IPv6 with UFW (Optional)

This tutorial is written with IPv4 in mind, but will work for IPv6 as well as long as you enable it. If your Ubuntu server has IPv6 enabled, ensure that UFW is configured to support IPv6 so that it will manage firewall rules for IPv6 in addition to IPv4. To do this, open the UFW configuration with `nano` or your favorite editor.

```
$ sudo nano /etc/default/ufw
```

Then make sure the value of `IPV6` is `yes`. It should look like this:

/etc/default/ufw excerpt

```
IPV6=yes
```

Save and close the file. Now, when UFW is enabled, it will be configured to write both IPv4 and IPv6 firewall rules. However, before enabling UFW, we will want to ensure that your firewall is configured to allow you to connect via SSH. Let's start with setting the default policies.

Step 2 — Setting Up Default Policies

If you're just getting started with your firewall, the first rules to define are your default policies. These rules control how to handle traffic that does not explicitly match any other rules. By default, UFW is set to deny all incoming connections and allow all outgoing connections. This means anyone trying to reach your server would not be able to connect, while any application within the server would be able to reach the outside world.

Let's set your UFW rules back to the defaults so we can be sure that you'll be able to follow along with this tutorial. To set the defaults used by UFW, use these commands:

```
$ sudo ufw default deny incoming
$ sudo ufw default allow outgoing
```

These commands set the defaults to deny incoming and allow outgoing connections. These firewall defaults alone might suffice for a personal computer, but servers typically need to respond to incoming requests from outside users. We'll look into that next.

Step 3 — Allowing SSH Connections

If we enabled our UFW firewall now, it would deny all incoming connections. This means that we will need to create rules that explicitly allow legitimate incoming connections — SSH or HTTP connections, for example — if we want our server to respond to those types of requests. If you're using a cloud server, you will probably want to allow incoming SSH connections so you can connect to and manage your server.

To configure your server to allow incoming SSH connections, you can use this command:

```
$ sudo ufw allow ssh
```

This will create firewall rules that will allow all connections on port `22`, which is the port that the SSH daemon listens on by default. UFW knows what port `allow ssh` means because it's listed as a service in the `/etc/services` file.

However, we can actually write the equivalent rule by specifying the port instead of the service name. For example, this command works the same as the one above:

```
$ sudo ufw allow 22
```

If you configured your SSH daemon to use a different port, you will have to specify the appropriate port. For example, if your SSH server is listening on port `2222`, you can use this command to allow connections on that port:

```
$ sudo ufw allow 2222
```

Now that your firewall is configured to allow incoming SSH connections, we can enable it.

Step 4 — Enabling UFW

To enable UFW, use this command:

```
$ sudo ufw enable
```

You will receive a warning that says the command may disrupt existing SSH connections. We already set up a firewall rule that allows SSH connections, so it should be fine to continue. Respond to the prompt with `y` and hit `ENTER`.

The firewall is now active. Run the `sudo ufw status verbose` command to see the rules that are set. The rest of this tutorial covers how to use UFW in more detail, like allowing or denying different kinds of connections.

Step 5 — Allowing Other Connections

At this point, you should allow all of the other connections that your server needs to respond to. The connections that you should allow depends on your specific needs. Luckily, you already know how to write rules that allow connections based on a service name or port; we already did this for SSH on port `22`. You can also do this for:

- HTTP on port 80, which is what unencrypted web servers use, using `sudo ufw allow http` or `sudo ufw allow 80`
- HTTPS on port 443, which is what encrypted web servers use, using `sudo ufw allow https` or `sudo ufw allow 443`

There are several others ways to allow other connections, aside from specifying a port or known service.

Specific Port Ranges

You can specify port ranges with UFW. Some applications use multiple ports, instead of a single port.

For example, to allow X11 connections, which use ports `6000`-`6007`, use these commands:

```
$ sudo ufw allow 6000:6007/tcp
$ sudo ufw allow 6000:6007/udp
```

When specifying port ranges with UFW, you must specify the protocol (`tcp` or `udp`) that the rules should apply to. We haven't mentioned this before because not specifying the protocol automatically allows both protocols, which is OK in most cases.

Specific IP Addresses

When working with UFW, you can also specify IP addresses. For example, if you want to allow connections from a specific IP address, such as a work or home IP address of `203.0.113.4`, you need to specify `from`, then the IP address:

```
$ sudo ufw allow from 203.0.113.4
```

You can also specify a specific port that the IP address is allowed to connect to by adding `to any port` followed by the port number. For example, If you want to allow `203.0.113.4` to connect to port `22` (SSH), use this command:

```
$ sudo ufw allow from 203.0.113.4 to any port 22
```

Subnets

If you want to allow a subnet of IP addresses, you can do so using CIDR notation to specify a netmask. For example, if you want to allow all of the IP addresses ranging from `203.0.113.1` to `203.0.113.254` you could use this command:

```
$ sudo ufw allow from 203.0.113.0/24
```

Likewise, you may also specify the destination port that the subnet `203.0.113.0/24` is allowed to connect to. Again, we'll use port `22` (SSH) as an example:

```
$ sudo ufw allow from 203.0.113.0/24 to any port 22
```

Connections to a Specific Network Interface

If you want to create a firewall rule that only applies to a specific network interface, you can do so by specifying "allow in on" followed by the name of the network interface.

You may want to look up your network interfaces before continuing. To do so, use this command:

```
$ ip addr
```

```
Output Excerpt
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
. . .
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
. . .
```

The highlighted output indicates the network interface names. They are typically named something like `eth0` or `enp3s2`.

So, if your server has a public network interface called `eth0`, you could allow HTTP traffic (port `80`) to it with this command:

```
$ sudo ufw allow in on eth0 to any port 80
```

Doing so would allow your server to receive HTTP requests from the public internet.

Or, if you want your MySQL database server (port `3306`) to listen for connections on the private network interface `eth1`, for example, you could use this command:

```
$ sudo ufw allow in on eth1 to any port 3306
```

This would allow other servers on your private network to connect to your MySQL database.

Step 6 — Denying Connections

If you haven't changed the default policy for incoming connections, UFW is configured to deny all incoming connections. Generally, this simplifies the process of creating a secure firewall policy by requiring you to create rules that explicitly allow specific ports and IP addresses through.

However, sometimes you will want to deny specific connections based on the source IP address or subnet, perhaps because you know that your server is being attacked from there. Also, if you want to change your default incoming policy to **allow** (which is not recommended), you would need to create **deny** rules for any services or IP addresses that you don't want to allow connections for.

To write **deny** rules, you can use the commands described above, replacing **allow** with **deny**.

For example, to deny HTTP connections, you could use this command:

```
$ sudo ufw deny http
```

Or if you want to deny all connections from `203.0.113.4` you could use this command:

```
$ sudo ufw deny from 203.0.113.4
```

Now let's take a look at how to delete rules.

Step 7 — Deleting Rules

Knowing how to delete firewall rules is just as important as knowing how to create them. There are two different ways to specify which rules to delete: by rule number or by the actual rule (similar to how the rules were specified when they were created). We'll start with the **delete by rule number** method because it is easier.

By Rule Number

If you're using the rule number to delete firewall rules, the first thing you'll want to do is get a list of your firewall rules. The UFW status command has an option to display numbers next to each rule, as demonstrated here:

```
$ sudo ufw status numbered
```

```
Numbered Output:
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         ALLOW IN    15.15.15.0/24
[ 2] 80                         ALLOW IN    Anywhere
```

If we decide that we want to delete rule 2, the one that allows port 80 (HTTP) connections, we can specify it in a UFW delete command like this:

```
$ sudo ufw delete 2
```

This would show a confirmation prompt then delete rule 2, which allows HTTP connections. Note that if you have IPv6 enabled, you would want to delete the corresponding IPv6 rule as well.

By Actual Rule

The alternative to rule numbers is to specify the actual rule to delete. For example, if you want to remove the `allow http` rule, you could write it like this:

```
$ sudo ufw delete allow http
```

You could also specify the rule by `allow 80`, instead of by service name:

```
$ sudo ufw delete allow 80
```

This method will delete both IPv4 and IPv6 rules, if they exist.

Step 8 — Checking UFW Status and Rules

At any time, you can check the status of UFW with this command:

```
$ sudo ufw status verbose
```

If UFW is disabled, which it is by default, you'll see something like this:

```
Output
Status: inactive
```

If UFW is active, which it should be if you followed Step 3, the output will say that it's active and it will list any rules that are set. For example, if the firewall is set to allow SSH (port `22`) connections from anywhere, the output might look something like this:

```
Output
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
```

Use the `status` command if you want to check how UFW has configured the firewall.

Step 9 — Disabling or Resetting UFW (optional)

If you decide you don't want to use UFW, you can disable it with this command:

```
$ sudo ufw disable
```

Any rules that you created with UFW will no longer be active. You can always run `sudo ufw enable` if you need to activate it later.

If you already have UFW rules configured but you decide that you want to start over, you can use the reset command:

```
$ sudo ufw reset
```

This will disable UFW and delete any rules that were previously defined. Keep in mind that the default policies won't change to their original settings, if you modified them at any point. This should give you a fresh start with UFW.

Conclusion

Your firewall is now configured to allow (at least) SSH connections. Be sure to allow any other incoming connections that your server requires, while limiting any unnecessary connections, so your server will be functional and secure.

To learn about more common UFW configurations, check out the UFW Essentials: Common Firewall Rules and Commands tutorial.

Report an issue

About the authors

Brian Boucheron

Senior Technical Writer at DigitalOcean

Still looking for an answer?

Ask a question

Search for more help

Comments

3 Comments

Leave a comment...

Sign In to Comment

sworks October 27, 2018

1

Thank you so much. Very helpful guide.

Reply Report

pierretempel February 20, 2019

4

Before enabling the firewall and potentially disrupting services, double check your configuration using `ufw show added`. This command exisits, because `ufw status` only works on an active firewall.

Reply Report

ekaharo October 29, 2019

0

Very thorough....thanks

Reply Report

BECOME A CONTRIBUTOR

You get paid; we donate to tech nonprofits.

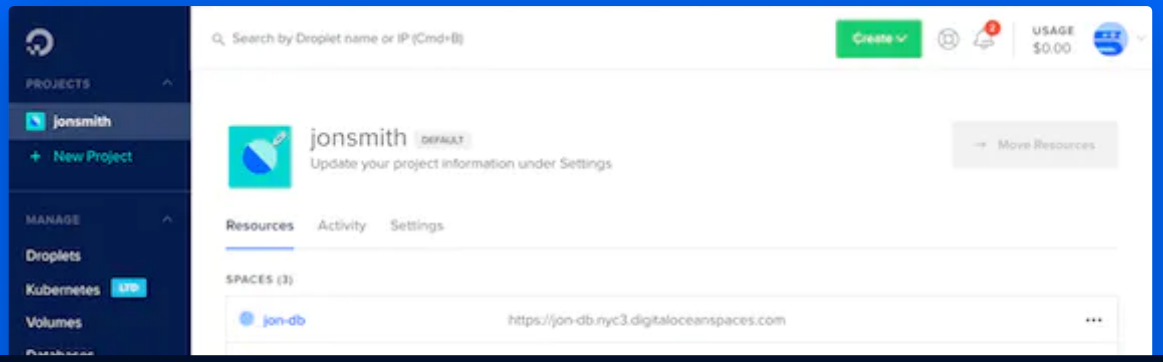
GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.

HUB FOR GOOD

Working on improving health and education, reducing inequality, and spurring economic growth? We'd like to help.

Learn More



Company

- About
- Leadership
- Blog
- Careers
- Partners
- Referral Program
- Press
- Legal
- Security & Trust Center

Products

- Pricing
- Products Overview
- Droplets
- Kubernetes
- Managed Databases
- Spaces
- Marketplace
- Load Balancers
- Block Storage
- API Documentation
- Documentation
- Release Notes

Community

- Tutorials
- Q&A
- Tools and Integrations
- Tags
- Product Ideas
- Write for DigitalOcean
- Presentation Grants
- Hatch Startup Program
- Shop Swag
- Research Program
- Open Source
- Code of Conduct

Contact

Get Support
Trouble Signing In?
Sales
Report Abuse
System Status