DigitalOcean

Subscribe

How To Setup a Firewall with UFW on an Ubuntu and Debian Cloud Server

253

Posted June 25, 2013 1m FIREWALL SECURITY IPV6 UBUNTU DEBIAN

By: Shaun Lewis

Introduction

One of the first lines of defense in securing your cloud server is a functioning firewall. In the past, this was often done through complicated and arcane utilities. There is a lot of functionality built into these utilities, iptables being the most popular nowadays, but they require a decent effort on behalf of the user to learn and understand them. Firewall rules are not something you want yourself second-guessing.

To this end, UFW is a considerably easier-to-use alternative.

What is UFW?

UFW, or Uncomplicated Firewall, is a front-end to iptables. Its main goal is to make managing your firewall drop-dead simple and to provide an easy-to-use interface. It's well-supported and popular in the Linux community—even installed by default in a lot of distros. As such, it's a great way to get started securing your sever.

Before We Get Started

First, obviously, you want to make sure UFW is installed. It should be installed by default in Ubuntu, but if for some reason it's not, you can install the package using aptitude or apt-get using the following commands:

```
sudo aptitude install ufw
```

or

```
sudo apt-get install ufw
```

Check the Status

You can check the status of UFW by typing:

```
sudo ufw status
```

Right now, it will probably tell you it is inactive. Whenever ufw is active, you'll get a listing of the current rules that looks similar to this:

```
Status: active

To                 Action      From
--                 ------      ----
22                 ALLOW       Anywhere
```

Using IPv6 with UFW

If your VPS is configured for IPv6, ensure that UFW is configured to support IPv6 so that will configure both your IPv4 and IPv6 firewall rules. To do this, open the UFW configuration with this command:

```
sudo vi /etc/default/ufw
```

Then make sure "IPV6" is set to "yes", like so:

```
IPV6=yes
```

Save and quit. Then restart your firewall with the following commands:

```
sudo ufw disable
sudo ufw enable
```

Now UFW will configure the firewall for both IPv4 and IPv6, when appropriate.

Set Up Defaults

One of the things that will make setting up any firewall easier is to define some default rules for allowing and denying connections. UFW's defaults are to deny all incoming connections and allow all outgoing connections. This means anyone trying to reach your cloud server would not be able to connect, while any application within the server would be able to reach the outside world. To set the defaults used by UFW, you would use the following commands:

```
sudo ufw default deny incoming
```

and

```
sudo ufw default allow outgoing
```

Note: if you want to be a little bit more restrictive, you can also deny all outgoing requests as well. The necessity of this is debatable, but if you have a public-facing cloud server, it could help prevent against any kind of remote shell connections. It does make your firewall more cumbersome to manage because you'll have to set up rules for all outgoing connections as well. You can set this as the default with the following:

```
sudo ufw default deny outgoing
```

Allow Connections

The syntax is pretty simple. You change the firewall rules by issuing commands in the terminal. If we turned on our firewall now, it would deny all incoming connections. If you're connected over SSH to your cloud server, that would be a problem because you would be locked out of your server. Let's enable SSH connections to our server to prevent that from happening:

```
sudo ufw allow ssh
```

As you can see, the syntax for adding services is pretty simple. UFW comes with some defaults for common uses. Our SSH command above is one example. It's basically just shorthand for:

```
sudo ufw allow 22/tcp
```

This command allows a connection on port 22 using the TCP protocol. If our SSH server is running on port 2222, we could enable connections with the following command:

```
sudo ufw allow 2222/tcp
```

Other Connections We Might Need

Now is a good time to allow some other connections we might need. If we're securing a web server with FTP access, we might need these commands:

`sudo ufw allow www` or `sudo ufw allow 80/tcp` `sudo ufw allow ftp` or `sudo ufw allow 21/tcp`

You mileage will vary on what ports and services you need to open. There will probably be a bit of testing necessary. In addition, you want to make sure you leave your SSH connection allowed.

Port Ranges

You can also specify port ranges with UFW. To allow ports 1000 through 2000, use the command:

```
sudo ufw allow 1000:2000/tcp
```

If you want UDP:

```
sudo ufw allow 1000:2000/udp
```

IP Addresses

You can also specify IP addresses. For example, if I wanted to allow connections from a specific IP address (say my work or home address), I'd use this command:

```
sudo ufw allow from 192.168.255.255
```

Denying Connections

Our default set up is to deny all incoming connections. This makes the firewall rules easier to administer since we are only selectively allowing certain ports and IP addresses through. However, if you want to flip it and open up all your server's ports (not recommended), you could allow all connections and then restrictively deny ports you didn't want to give access to by replacing "allow" with "deny" in the commands above. For example:

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

would allow access to port 80 while:

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

would deny access to port 80.

Deleting Rules

There are two options to delete rules. The most straightforward one is to use the following syntax:

```
sudo ufw delete allow ssh
```

As you can see, we use the command “delete” and input the rules you want to eliminate after that. Other examples include:

```
sudo ufw delete allow 80/tcp
```

or

```
sudo ufw delete allow 1000:2000/tcp
```

This can get tricky when you have rules that are long and complex.

A simpler, two-step alternative is to type:

```
sudo ufw status numbered
```

which will have UFW list out all the current rules in a numbered list. Then, we issue the command:

```
sudo ufw delete [number]
```

where “[number]” is the line number from the previous command.

Turn It On

After we've gotten UFW to where we want it, we can turn it on using this command (remember: if you're connecting via SSH, make sure you've set your SSH port, commonly port 22, to be allowed to receive connections):

```
sudo ufw enable
```

You should see the command prompt again if it all went well. You can check the status of your rules now by typing:

```
sudo ufw status
```

or

```
sudo ufw status verbose
```

for the most thorough display.

To turn UFW off, use the following command:

```
sudo ufw disable
```

Reset Everything

If, for whatever reason, you need to reset your cloud server's rules to their default settings, you can do this by typing this command:

```
sudo ufw reset
```

Conclusion

You should now have a cloud server that is configured properly to restrict access to a subset of ports or IP addresses.

By: Shaun Lewis

Related Tutorials

How To Test your Firewall Configuration with Nmap and Tcpdump

How To Set Up an OpenVPN Server on Debian 9

How To Create a Self-Signed SSL Certificate for Nginx on Debian 9

How To Create a Self-Signed SSL Certificate for Apache in Debian 9

How To Set Up vsftpd for a User's Directory on Debian 9

82 Comments

Leave a comment...

Log In to Comment

james174592 *July 4, 2013*

0 Great article, glad I found this because looking at the iptables firewall option was way over my head as a noob. Quick question, aside from having this firewall setup, should I still use the iptables firewall option as well or is there another form of server security I should be using as well. Thanks for the info here and the help.

---

kamaln7 MOD *July 4, 2013*

1 @james: UFW is an iptables wrapper, you're indirectly using iptables while using ufw. ;]

---

piscue *July 17, 2013*

0 useful article, I set up everything I needed in 5 minutes...

---

caesarsgrunt *July 22, 2013*

0 This works in Debian too; you might want to change the title so as not to imply that it's just for Ubuntu.

---

kamaln7 MOD *July 22, 2013*

0 @caesarsgrunt: Thanks! Updated.

---

kevin_thulin *August 4, 2013*

0 So to allow ssh should I use:

sudo ufw allow ssh

AND

sudo ufw allow 22/tcp ?

Thanks

---

kamaln7 MOD *August 4, 2013*

0 No, you should use only one of them, not both.

---

craig198396 *August 16, 2013*

1 What's the easiest way to add a whole range of specific IP addresses, such as https://www.cloudflare.com/ips?

---

geekgonecrazy *August 28, 2013*

0 Thanks for this. Made it pretty quick for me to get started. Used iptable setups before, but they can be annoying to setup, and have remember the rules etc.

---

manuel.bua *August 31, 2013*

1 On my debian/wheezy64 i keep having problems with ufw: trying to enable it causes the error "ERROR: problem running ufw-init" and it will not autostart at boot, does anyone know how to solve this? On my local wheezy installation everything works fine, but my box at digitalocean won't do it :(

---

alexeydemin *July 19, 2016*

0 Set IPV6=no in /etc/default/ufw

---

kamaln7 MOD *August 31, 2013*

0 @manuel.bua: Try editing

```
/etc/default/ufw
```

and setting IPv6 to no.

---

manuel.bua *August 31, 2013*

0 @KamalNasser: that did it, thanks! So i suspect this has to do with the fact it's a VPS since my parallel wheezy installation in VirtualBox is working fine.

---

kamaln7 MOD *September 1, 2013*

0 @manuel.bua: It's because our platform does not support IPv6 so you have to disable ufw's IPv6 support in order for ufw to work :]

---

manuel.bua *September 1, 2013*

0

Ok, thanks for clarifying: keep up the good work guys, this platform is really awesome and i'm looking forward to making big use out of it!

Rosalyn *September 15, 2013*

0 Nice.
How do I forward port 8080(tomcat) to 80 via ufw?
Vic

---

kamaln7 MOD *September 16, 2013*

0 @cekvenich.vic: Edit **/etc/ufw/before.rules** and add

```
*nat
:PREROUTING - [0:0]

iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080

# don't delete the 'COMMIT' line or these nat table rules won't be processed
COMMIT
```

at the very top of the file.

Save and restart ufw:

```
sudo service ufw restart
```

---

lanbase *January 11, 2015*

0 This doesn´t work for me , I get an error restarting the firewall;

root@LanVPS:~# vi /etc/ufw/before.rules
root@LanVPS:~# sudo service ufw restart
[....] Reloading firewall: ufw...iptables-restore v1.4.14: Line 15 seems to have a -t table option.

Error occurred at line: 15
Try `iptables-restore -h' or 'iptables-restore --help' for more information.

[FAILem running '/etc/ufw/before.rules'...failed.
root@LanVPS:~#

Any advice ?

Thanks in advance

asterixzzz *September 23, 2013*

0 What is the difference between
sudo ufw insert 1 allow 80
and
sudo ufw allow 80/tcp
If I only want to allow web browser traffic (http/https) on this port, what shall enter?

kamaln7 MOD *September 23, 2013*

0 Both should work fine. The first command explicitly tells ufw to insert the rule at the top while the second command will insert it at the bottom.

ny.roy.berry *December 22, 2013*

0 Loved the article - was one of the first available in my google search on UFW -- been a while!

jb5531 *January 24, 2014*

0 I followed these instructions and when UFW is enabled I cannot access my site via a browser going to my IP address (when I could before). The only rules I have on my UFW are:

To Action From
-- ------ ----
2222/tcp ALLOW IN Anywhere
2222/tcp ALLOW IN Anywhere (v6)

(Where 2222 is my SSH port, but in reality 2222 is not it, just using this for privacy concerns).

CTala *April 9, 2015*

0 But your website should be on port 80 no ?

Load More Comments