CONTENTS

RELATED

Cómo instalar Elasticsearch, Logstash y Kibana (Elastic Stack) en Ubuntu 18.04

View

Cómo instalar y configurar una entidad de certificación (CA) en Debian 10

View

// Tutorial //

Cómo configurar un firewall con UFW en Ubuntu 18.04

Published on December 5, 2019

Security Ubuntu Firewall Networking Ubuntu 18.04

By Erika Heidi and Brian Boucheron

Español

Hazel Virdó escribió una versión anterior de este tutorial.

Introducción

UFW o Uncomplicated Firewall es una interfaz para `iptables` orientada a simplificar el proceso de configuración de un firewall. Aunque `iptables` es una herramienta sólida y flexible, puede resultar difícil para los principiantes aprender a usarlo para configurar correctamente un firewall. Si desea comenzar a proteger su red y no está seguro de qué herramienta utilizar, UFW puede ser su mejor opción.

En este tutorial, verá la forma de configurar un firewall con UFW en Ubuntu 18.04.

Requisitos previos

Para completar este tutorial, necesitará lo siguiente:

- Un servidor de Ubuntu 18.04 con un usuario sudo no root, que puede configurarse siguiendo los pasos 1 a 3 del tutorial de configuración inicial para servidores de Ubuntu 18.04.

UFW se instala por defecto en Ubuntu. Si se desinstaló por alguna razón, puede instalarlo con `sudo apt install ufw2`.

Paso 1: Utilizar IPv6 con UFW (opcional)

Este tutorial se redactó teniendo en cuenta IPv4, pero funcionará para IPv6 siempre que lo habilite. Si su servidor de Ubuntu tiene IPv6 habilitado, compruebe que UFW esté configurado para que admitir IPv6 de modo que administre las reglas de firewall para IPv6 además de IPv4. Para hacerlo, abra la configuración de UFW con `nano` o su editor favorito.

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

Copy

A con[illegible]ión, asegúrese de que el valor de `IPV6` sea `yes`. Debería tener el siguiente aspecto:

/etc/default/ufw excerpt

```
IPV6=yes
```

Guarde y cierre el archivo. Cuando UFW esté habilitado, se configurará para escribir reglas de firewall de IPv4 y IPv6. Sin embargo, antes de habilitar UFW, nos convendrá comprobar que su firewall esté configurado para que pueda conectarse a través de SSH. Empezaremos con la configuración de las políticas predeterminadas.

Paso 2: Configurar políticas predeterminadas

Si recién está dando los primeros pasos con su firewall, las primeras reglas a definir son sus políticas predeterminadas. Estas reglas controlan la administración del tráfico que no coincida de forma explícita con otras reglas. Por defecto, UFW está configurado para denegar todas las conexiones entrantes y permitir todas las conexiones salientes. Esto significa que quien intente llegar a su servidor no podrá conectarse, mientras que cualquier aplicación dentro del servidor podría llegar al mundo exterior.

Restableceremos los valores predeterminados de sus reglas de UFW para garantizar que podamos seguir con este tutorial. Para fijar los valores predeterminados utilizados por UFW, emplee estos comandos:

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

Establecen los valores predeterminados para denegar las conexiones entrantes y permitir las salientes. Con solo estos valores predeterminados de firewall podría bastar para una computadora personal, pero normalmente los servidores deben responder a las solicitudes de usuarios externos. Lo veremos a continuación.

Paso 3: Habilitar conexiones SSH

Si habilitamos nuestro firewall de UFW ahora, denegaría todas las conexiones entrantes. Esto significa que deberemos crear reglas que permitan explícitamente las conexiones entrantes legítimas (SSH o HTTP, por ejemplo) si queremos que nuestro servidor responda a estos tipos de solicitudes. Si utiliza un servidor en nube, probablemente le convenga permitir las conexiones SSH entrantes para poder conectarse y administrar su servidor.

Para configurar su servidor de modo que permita las conexiones SSH entrantes, puede utilizar este comando:

```
$ sudo ufw allow ssh
```
Copy

Esto creará reglas de firewall que permitirán todas las conexiones en el puerto `22`, que es el que escucha el demonio SSH por defecto. UFW registra el significado del puerto `allow ssh` porque está enumerado como servicio en el archivo `/etc/services`.

Sin embargo, realmente podemos escribir la regla equivalente especificando el puerto en vez del nombre de servicio. Por ejemplo, este comando funciona como el anterior:

```
$ sudo ufw allow 22
```
Copy

Si configuró su demonio SSH para utilizar un puerto diferente, deberá especificar el puerto apropiado. Por ejemplo, si su servidor SSH escucha en el puerto `2222` puede utilizar este comando para permitir las conexiones en ese puerto:

```
$ sudo ufw allow 2222
```
Copy

Ahora que su firewall está configurado para permitir las conexiones SSH entrantes, podemos habilitarlo.

Paso 4: Habilitar UFW

Para habilitar UFW, utilice este comando:

```
$ sudo ufw enable
```
Copy

Recibirá una advertencia que indicará que el comando puede interrumpir las conexiones SSH existentes. Ya configuramos una regla de firewall que permite conexiones SSH. Debería ser posible continuar sin inconvenientes. Responda a la solicitud con `y` y presione `ENTER`.

Con esto, el firewall quedará activo. Ejecute el comando `sudo ufw status verbose` para ver las reglas que se configuran. En el resto de este tutorial se abarca en mayor profundidad la forma de utilizar UFW. Se analizarán opciones como las de permitir o denegar diferentes tipos de conexiones.

Paso 5: Habilitar otras conexiones

En este momento, debería permitir el resto de las conexiones a las que su servidor debe responder. Las conexiones que debería permitir dependen de sus necesidades específicas. Afortunadamente, ya sabe cómo escribir reglas que permiten las conexiones basadas en un nombre de servicio o un puerto; ya lo hicimos para SSH en el puerto `22`. También puede hacerlo para:

- HTTP en el puerto 80, que es lo que utilizan los servidores web no cifrados, con `sudo ufw allow http` o `sudo ufw allow 80`
- HTTPS en el puerto 443, que es lo que utilizan los servidores web cifrados, con `sudo ufw allow https` o `sudo ufw allow 443`

Existen varias maneras de permitir otras conexiones, aparte de especificar un puerto o un servicio conocido.

Intervalos de puerto específicos

Puede especificar intervalos de puerto con UFW. Algunas aplicaciones utilizan varios puertos en vez de uno solo.

Por ejemplo, para permitir las conexiones de X11 que utilizan los puertos `6000`-`6007`, emplee estos comandos:

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

Cuando se especifiquen intervalos de puerto con UFW, debe especificar el protocolo (`tcp` o `udp`) a los que deberían aplicarse las reglas. No lo mencionamos antes porque cuando no se especifica el protocolo se permiten ambos de forma automática, lo cual está bien en la mayoría de los casos.

Direcciones IP específicas

Al trabajar con UFW, también puede especificar direcciones IP. Por ejemplo, si desea permitir las conexiones desde una dirección IP específica, como una dirección IP de trabajo o doméstica `203.0.113.4`, debe especificar `from` y luego la dirección IP:

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

También puede especificar un puerto concreto al que la dirección IP pueda conectarse agregando `to any port` seguido del número de este. Por ejemplo, si desea permitir que `203.0.113.4` se conecte al puerto `22` (SSH), utilice este comando:

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

Subredes

Si desea permitir una subred de direcciones IP, puede hacerlo utilizando la notación CIDR para especificar una máscara de red. Por ejemplo, si desea permitir todas las direcciones IP de la `203.0.113.1` a la `203.0.113.254` podría utilizar este comando:

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

Del mismo modo, también puede especificar el puerto de destino al que puede conectarse la subred `203.013.0/24`. Una vez más, utilizaremos el puerto `22` (SSH) como ejemplo:

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

Conexiones a una interfaz de red específica

Si desea crear una regla de firewall que solo se aplique a una interfaz de red específica, puede hacerlo especificando “allow in on” seguido del nombre de la interfaz de red.

Tal vez le convenga revisar sus interfaces de red antes de continuar. Para hacerlo, utilice este comando:

```
$ ip addr
```
Copy

```
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
. . .
```

El resultado resaltado indica los nombres de la interfaz de red. Normalmente tienen nombres similares a `eth0` o `enp3s2`.

Así, si su servidor tiene una interfaz de red pública llamada `eth0`, podría permitir el tráfico HTTP (puerto `80`) hacia él con este comando:

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

Hacerlo permitiría que su servidor recibiera solicitudes HTTP desde la Internet pública.

O bien, si desea que su servidor de base de datos de MySQL (puerto `3306`) escuche las conexiones en la interfaz de red privada `eth1`, por ejemplo, podría utilizar este comando:

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

Esto permitiría que otros servidores de su red privada se conectaran a su base de datos de MySQL.

Paso 6: Denegar conexiones

Si no ha cambiado la política predeterminada para las conexiones entrantes, UFW está configurado para denegarlas todas. Generalmente, esto simplifica el proceso de creación de una política de firewall segura al exigirle crear reglas que permitan de forma explícita el acceso de puertos específicos y direcciones IP.

Sin embargo, a veces le convendrá denegar conexiones específicas basadas en la dirección IP o subred de origen, quizás por saber que su servidor recibe ataques desde ellas. Además, si desea cambiar su política entrante predeterminada para **allow** (no lo recomendamos), debería crear reglas **deny** para cualquier servicio o dirección IP cuyas conexiones no desee permitir.

Para escribir reglas de **deny**, puede utilizar los comandos descritos anteriormente y sustituir **allow** por **deny**.

Por ejemplo, para denegar conexiones HTTP, podría utilizar este comando:

```
$ sudo ufw deny http
```

A su vez, si desea denegar todas las conexiones de `203.0.113.4` podría utilizar este comando:

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

Ahora veremos la forma de eliminar reglas.

Paso 7: Eliminar reglas

Saber eliminar reglas de firewall es tan importante como saber crearlas. Existen dos maneras diferentes de especificar las reglas que se eliminarán: por número de regla o por regla real (se asemejan a la forma en que las reglas se especifican al crearse). Comenzaremos con el **método de eliminación por el número de regla** porque es más sencillo.

Por número de regla

Si utiliz[illegible] número de regla para eliminar reglas de firewall, lo primero que le convendrá hacer es obtener una lista de reglas de firewall. El comando "UFW status" tiene una opción pa

```
Status: active

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

Si decidimos eliminar la regla 2, que permite las conexiones del puerto 80 (HTTP), podemos especificarlo en un comando “UFW delete”como este:

```
$ sudo ufw delete 2
```
Copy

Esto mostraría un mensaje de confirmación y eliminaría la regla 2, que permite conexiones HTTP. Tenga en cuenta que si tiene IPv6 habilitado, le convendría eliminar también la regla IPv6 correspondiente.

Por regla real

La alternativa a números de regla es especificar la regla real que se eliminará. Por ejemplo, si desea eliminar la regla `allow http`, podría escribir lo siguiente:

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

También podría especificar la regla mediante `allow 80` en vez de hacerlo por nombre de servicio:

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

Este método eliminará las reglas IPv4 y IPv6, si existen.

Paso 8: Comprobar el estado y las reglas de UFW

En cualquier momento, puede verificar el estado de UFW con este comando:

```
$ sudo ufw status verbose
```
Copy

Si UFW está desactivado, lo cual se aplica por defecto, verá algo como esto:

```
Output
Status: inactive
```

Si UFW está activo, lo cual debería suceder si siguió el paso 3, el resultado dirá que está activo y enumerará cualquier regla configurada. Por ejemplo, si el firewall está configurado para permitir conexiones SSH (puerto `22`) desde cualquier parte, el resultado podría ser parecido a este:

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

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

Utilice el comando `status` si desea verificar la configuración que UFW aplicó al firewall.

Paso 9: Deshabilitar o reiniciar UFW (opcional)

Si decide que no desea utilizar UFW, puede desactivarlo con este comando:

```
$ sudo ufw disable
```
Copy

Cualquier regla que haya creado con UFW dejará de estar activa. Siempre puede ejecutar `sudo ufw enable` si necesita activarla más adelante.

Si ya tiene reglas de UFW configuradas y decide que desea empezar de nuevo, puede utilizar el comando “reset”:

```
$ sudo ufw reset
```
Copy

Esto desactivará UFW y eliminará cualquier regla definida anteriormente. Tenga en cuenta que los ajustes originales de las políticas predeterminadas no se restablecerán si las modificó en algún momento. Esto debería permitirle empezar de nuevo con UFW.

Conclusión

De esta manera, su firewall quedará configurado para permitir conexiones SSH (al menos). Asegúrese de permitir cualquier otra conexión entrante de su servidor y, al mismo tiempo, limitar cualquier conexión innecesaria, de modo que su servidor funcione y sea seguro.

Para obtener información sobre más configuraciones comunes de UFW, consulte el tutorial Aspectos básicos de UFW: reglas y comandos comunes de firewall.

About the authors

[Erika Heidi](#) Author

Developer Advocate

Dev/Ops passionate about open source, PHP, and Linux.

[Brian Boucheron](#) Author

Developer and author at DigitalOcean.

Still looking for an answer?

Ask a question | Search for more help

Was this helpful? Yes No

Comments

1 Comments

Leave a comment...

This textbox defaults to using **Markdown** to format your answer.

You can type **!ref** in this text area to quickly search our full set of tutorials, documentation & marketplace offerings and insert the link!

Sign In or Sign Up to Comment

DrB4cter10 • November 18, 2020

Muchas gracias por la guía ha sido de bastante utilidad para tener un sitio por donde empezar, sólo un pequeño fallo que he visto, para instalar UFW no se hace con “sudo apt install ufw2”, se instala con “sudo apt install ufw”. Estaría bien una guía para añadir reglas especiales, por ejemplo como añadir este tipo de reglas a UFW: “iptables -t filter -A INPUT -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s --limit-burst 2 -j DROP”

Show replies Reply

Popular Topics

Ubuntu

Linux Basics

JavaScript

React

Python

Security

MySQL

Docker

Kubernetes

Browse all topic tags

Free Managed Hosting →

All tutorials →

Questions

Q&A Forum

Ask a question

DigitalOcean Support

Congratulations on unlocking the whale ambience easter egg! Click the whale button in the bottom left of your screen to toggle some ambient whale noises while you read.

Thank you to the Glacier Bay National Park & Preserve and Merrick079 for the sounds behind this easter egg.

Interested in whales, protecting them, and their connection to helping prevent climate change? We recommend checking out the Whale and Dolphin Conservation.

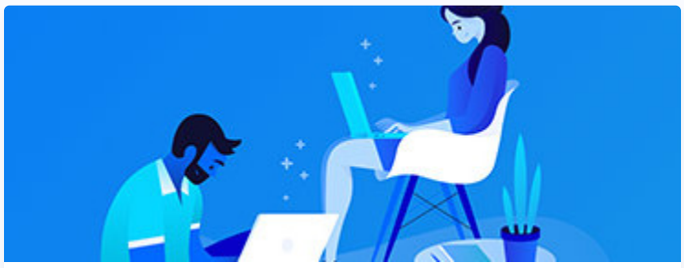
Reset easter egg to be discovered again / Permanently dismiss and hide easter egg

GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.

HOLLIE'S HUB FOR GOOD

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



BECOME A CONTRIBUTOR

You get paid; we donate to tech nonprofits.

Company

- About
- Leadership
- Blog
- Careers
- Customers
- Partners
- Channel Partners
- Referral Program
- Affiliate Program
- Press
- Legal
- Security
- Investor Relations
- DO Impact

Products

- Products Overview
- Droplets
- Kubernetes
- App Platform
- Functions
- Cloudways
- Managed Databases
- Spaces
- Marketplace
- Load Balancers
- Block Storage
- Tools & Integrations
- API
- Pricing
- Documentation
- Release Notes
- Uptime

Community

- Tutorials
- Q&A
- CSS-Tricks
- Write for DOnations
- Currents Research
- Hatch Startup Program
- deploy by DigitalOcean
- Shop Swag
- Research Program
- Open Source
- Code of Conduct
- Newsletter Signup
- Meetups

Solutions

- Website Hosting
- VPS Hosting
- Web & Mobile Apps
- Game Development
- Streaming
- VPN
- SaaS Platforms
- Cloud Hosting for Blockchain
- Startup Resources

Contact

- Support
- Sales
- Report Abuse
- System Status
- Share your ideas