

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Materia: | **ASEGURAMIENTO DE INFRAESTRUCTURAS TECNOLÓGICAS** | | **Taller 2 Firewall Linux** | |
| Docente: | **Donny Alexander Rodríguez Cáceres** | Fecha: | |  |

|  |  |
| --- | --- |
| **Integrantes** | **ID** |
| David Guillermo Gonzalez Breton |  |
| Daniel Esteban Valdivieso Diaz |  |
| Juan Pablo Adames |  |
|  |  |

|  |
| --- |
| **OBJETIVOS** |
| 1. Realizar la configuración y pruebas en el firewall de Linux |
| **COMPETENCIAS A DESARROLLAR** |
| * Capacidad de análisis de resolución de necesidades técnicas. * Capacidad de abrir y cerrar puertos en firewall según las necesidades |

|  |
| --- |
| **ENTREGABLES** |
| 1. **(1) archivo** con las pruebas de configuración y ping de firewall Linux |

|  |
| --- |
| **Realizar** |

**TALLER FIREWALL UFW - LINUX**

**Introducción**

UFW, o Uncomplicated Firewall, es una interfaz de gestión de firewall simplificada que oculta la complejidad de las tecnologías de filtrado de paquetes de nivel inferior, como iptables y nftables. Si desea comenzar a proteger su red y no está seguro respecto de la herramienta que debe utilizar, UFW puede ser la mejor opción.

En este tutorial, verá la manera de configurar un firewall con UFW en Ubuntu 20.04.

Requisitos previos

Para este tutorial, necesitará lo siguiente:

1. **Un servidor de Ubuntu 20.04 con un non-root user sudo**

Una vez que haya iniciado sesión como raíz, podrá agregar la nueva cuenta de usuario. En el futuro, iniciaremos sesión con esta nueva cuenta en lugar de root.

Este ejemplo crea un nuevo usuario llamado sammy, pero debe reemplazarlo con un nombre de usuario que le guste:

**adduser sammy**

Se le harán algunas preguntas, comenzando con la contraseña de la cuenta.

Ingrese una contraseña segura y, opcionalmente, complete cualquier información adicional si lo desea. Esto no es obligatorio y puede presionar ENTER cualquier campo que desee omitir.

Ahora tenemos una nueva cuenta de usuario con privilegios de cuenta regulares. Sin embargo, en ocasiones es posible que necesitemos realizar tareas administrativas.

Para evitar tener que salir de nuestro usuario normal y volver a entrar como cuenta root , podemos configurar lo que se conoce como privilegios de superusuario o root para nuestra cuenta normal. Esto permitirá que nuestro usuario normal ejecute comandos con privilegios administrativos poniendo la palabra sudo antes del comando.

Para agregar estos privilegios a nuestro nuevo usuario, debemos agregar el usuario al grupo sudo . De forma predeterminada, en Ubuntu 20.04, los usuarios que son miembros del grupo sudo pueden usar el sudo comando.

Como root , ejecute este comando para agregar su nuevo usuario al grupo sudo (sustituya el nombre de usuario resaltado con su nuevo usuario):

**usermod -aG sudo Sammy**

**su Sammy**

**sudo apt-get update – ejemplo**

**sudo ls -la /root**

Texto

Descripción generada automáticamente

1. **UFW viene instalado por defecto en Ubuntu. Si se desinstaló por alguna razón, puede instalarlo con:**

**sudo apt install ufw2.**

**Configurar políticas predeterminadas**

Si recién está dando los primeros pasos con su firewall, las primeras reglas que deben definirse son sus políticas predeterminadas. Estas reglas controlan el manejo 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 establecer conexión con su servidor no podrá hacerlo, mientras que cualquier aplicación dentro del servidor podrá 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**

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.

Texto

Descripción generada automáticamente

1. **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**

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

**sudo ufw allow 22**

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**

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

Texto

Descripción generada automáticamente con confianza media

1. **Habilitar UFW**

Para habilitar UFW, utilice este comando:

**sudo ufw enable**

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.

**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 a 6007, use estos comandos:

**sudo ufw allow 6000:6007/tcp**

**sudo ufw allow 6000:6007/udp**

Al especificar intervalos de puerto con UFW, debe definir el protocolo (tcp o udp) al 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 definir from y luego la dirección IP:

**sudo ufw allow from 203.0.113.4**

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**

Texto

Descripción generada automáticamente

**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**

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**

**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 optando por “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**

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**

Texto

Descripción generada automáticamente

Al 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.

Texto

Descripción generada automáticamente

1. **Denegar conexiones**

Si no cambió la política predeterminada para las conexiones entrantes, UFW está configurado para denegarlas a 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 el valor su política entrante predeterminada a 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**.**

Texto

Descripción generada automáticamente

1. **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 número de regla porque es más sencillo.

Por número de regla

Si utiliza el 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 para mostrar números junto a cada regla, como se muestra aquí:

**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**

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**

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.

Texto

Descripción generada automáticamente

**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**

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

**sudo ufw delete allow 80**

Con este método se eliminarán las reglas IPv4 y IPv6, si existen.

Texto

Descripción generada automáticamente

1. **Comprobar el estado y las reglas de UFW**

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

**sudo ufw status verbose**

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, en el resultado se indicará que está activo y se 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.

Texto

Descripción generada automáticamente

1. **Deshabilitar o reiniciar UFW (opcional)**

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

**sudo ufw disable**

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

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

**sudo ufw reset**

Con esto se desactivará UFW y se 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.

Texto

Descripción generada automáticamente

**CONCLUSIONES**

UFW es una herramienta que facilita al usuario las configuraciones de firewall internas a nivel de sistema, pues con esta, no hay necesidad de implementar directamente una herramienta como IPTABLES que debido a su complejidad puede llevar más tiempo de configuración.

La implementación de un firewall es esencial en un sistema operativo, pues mantener el control del trafico es una capa de seguridad que no debe saltarse.