Cómo configurar claves de SSH en Ubuntu 16.04

Introducción

SSH (*Secure SHell*), o **intérprete de comandos seguro**, es un protocolo cifrado que se usa para comunicarse con servidores y administrarlos en consecuencia. Al trabajar con un servidor de Ubuntu, es probable que pase la mayor parte de su tiempo en una sesión de terminal conectada a su servidor a través de SSH.

En esta guía, nos centraremos en configurar claves SSH para una instalación de Ubuntu 16.04 con vanilla (un kernel bien depurado y estable). Las claves de SSH proporcionan una alternativa sencilla y segura para iniciar sesión en su servidor y se recomienda para todos los usuarios.

Paso 1: Crear el par de claves RSA

RSA es un sistema criptográfico de clave pública creadas por los profesores **R**ivest, **S**hamir y **A** dleman. El primer paso es crear un par de claves de este tipo en la máquina cliente (por lo general, su computadora) por medio de la siguienta herramienta:

shell
$ ssh-keygen

De forma predeterminada, ssh-keygen creará un par de claves RSA de 2048 bits, que ofrece suficiente seguridad para la mayoría de los casos de uso (como opción, puede pasar en el indicador **-b 4096** para crear una clave más grande de 4096 bits).

Después de ingresar el comando, verá el siguiente resultado:

Salida por pantalla

Generating public/private rsa key pair.
Enter file in which to save the key (/your_home/.ssh/id_rsa):

Presione INTRO para guardar el par de claves en el subdirectorio .ssh/ de su directorio principal, o especificar una ruta alternativa.

Si generó previamente un par de claves de SSH, puede ver el siguiente mensaje:

Salida por pantalla

/home/your_home/.ssh/id\rsa already exists.
Overwrite (y/n)?

Si elige sobrescribir la clave en el disco, ya **no** podrá autenticar usando la clave anterior. Tenga mucho cuidado al convalidar la operación, ya que este es un proceso destructivo que no puede revertirse.

Debería ver el siguiente mensaje:

Salida por pantalla

Enter passphrase (empty for no passphrase):

Aquí, puede introducir una frase de contraseña segura, lo cual es muy recomendable. Una frase de contraseña agrega una capa de seguridad adicional para evitar el inicio de sesión de usuarios no autorizados. Para obtener más información sobre seguridad, consulte nuestro tutorial (en idioma inglés) Cómo configurar la autenticación basada en claves de SSH en un servidor de Linux.

Debería ver el siguiente resultado:

Salida por pantalla

Your identification has been saved in /your_home/.ssh/id_rsa.
Your public key has been saved in /your_home/.ssh/id_rsa.pub.
The key fingerprint is:
a9:49:2e:2a:5e:33:3e:a9:de:4e:77:11:58:b6:90:26 username@remote_host
The key's randomart image is:
+--[RSA 2048]----+
| ..o |
| E o= . |
| o. o |
| .. |
| ..S |
| o o. |
| =o.+. |
|. =++.. |
|o=++. |
+-----------------+

Ahora dispondrá de una clave pública y privada que puede usar para realizar la autenticación. El siguiente paso es ubicar la clave pública en su servidor a fin de poder usar la autenticación basada en claves de SSH para iniciar sesión.

Paso 2: Copiar la clave pública al servidor de Ubuntu

La alternativa más rápida para copiar su clave pública al anfitrión de Ubuntu es usar una utilidad llamada ssh-copy-id. Debido a su simplicidad, este método se recomienda mucho si está disponible. Si no tiene la utilidad ssh-copy-id disponible en su máquina cliente, puede usar uno de los dos métodos alternativos proporcionados en esta sección (copiar mediante SSH con contraseña o copiar manualmente la clave).

Copiar clave pública usando ssh-copy-id

La herramienta ssh-copy-id se incluye por defecto en muchos sistemas operativos. Por ello, es posible que tenga la posibilidad de disponer de ella en su sistema local. Para que este método funcione, ya debe disponer de acceso con SSH basado en contraseña en su servidor.

Para usar la utilidad, solo necesita especificar el host remoto al que desee conectarse y la cuenta de usuario a la que tenga acceso SSH con contraseña. Esta es la cuenta a la que se copiará su clave de SSH pública.

La sintaxis es la siguiente:

$ ssh-copy-id username@remote_host

Es posible que vea el siguiente mensaje:

Salida por pantalla

```
The authenticity of host '203.0.113.1 (203.0.113.1)' can't be established
. ECDSA key fingerprint is fd:fd:d4:f9:77:fe:73:84:e1:55:00:ad:d6:6d:22:f
e. Are you sure you want to continue connecting (yes/no)? yes
```

Esto significa que su computadora local no reconoce el host remoto. Esto sucederá la primera vez que establezca conexión con un nuevo host. Escriba “yes” (sí) y presione INTRO para continuar.

A continuación, la utilidad analizará su cuenta local en busca de la clave id_rsa.pub que creamos antes. Cuando la encuentre, le solicitará la contraseña de la cuenta del usuario remoto:

Salida por pantalla

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
username@203.0.113.1's password:

Escriba la contraseña (la cual, por motivos de seguridad, no se mostrará cuando usted escriba) y presione INTRO. La utilidad se conectará a la cuenta en el anfitrión remoto usando la contraseña que proporcionó. Luego, copie el contenido de su clave ~/.ssh/id_rsa.pub a un archivo en el directorio principal de la cuenta remota ~/.ssh llamado authorized_keys.

Debería ver el siguiente resultado:

Salida por pantalla

```
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'username@203.0.113.1'" and
check to make sure that only the key(s) you wanted were added.
```

En este punto, su clave id_rsa.pub se habrá cargado en la cuenta remota y puede continuar con el paso 3.

Copiar la clave pública usando SSH

Si no tiene ssh-copy-id disponible, pero tiene acceso de SSH basado en contraseña a una cuenta de su servidor, puede cargar sus claves usando un método de SSH convencional.

Podemos hacer esto usando el comando cat para leer el contenido de la clave de SSH pública en nuestra computadora local y canalizando esto a través de una conexión SSH al servidor remoto.

Por otra parte, podemos asegurarnos de que el directorio ~/.ssh exista y tenga los permisos correctos conforme a la cuenta que usamos.

Luego podemos transformar el contenido que canalizamos a un archivo llamado authorized_keys dentro de este directorio. Usaremos el símbolo de redireccionamiento >> para anexar el contenido en lugar de sobrescribirlo. Esto nos permitirá agregar claves sin eliminar claves previamente

agregadas.

El comando completo tiene este aspecto:

$ cat ~/.ssh/id_rsa.pub | ssh username@remote_host "mkdir -p ~/.ssh && touch ~/.ssh/authorized_keys && chmod -R go= ~/.ssh && cat >> ~/.ssh/authorized_keys"

Es posible que vea el siguiente mensaje:

Salida por pantalla

The authenticity of host '203.0.113.1 (203.0.113.1)' can't be established.
ECDSA key fingerprint is fd:fd:d4:f9:77:fe:73:84:e1:55:00:ad:d6:6d:22:fe.
Are you sure you want to continue connecting (yes/no)? yes

Esto significa que su computadora local no reconoce el host remoto. Esto sucederá la primera vez que establezca conexión con un nuevo host. Escriba “yes” (sí) y presione INTRO para continuar.

Posteriormente, deberá recibir la solicitud de introducir la contraseña de la cuenta de usuario remota:

Salida por pantalla

$ username@203.0.113.1's password:

Una vez que ingrese su contraseña, el contenido de su clave id_rsa.pub se copiará al final del archivo authorized_keys de la cuenta del usuario remoto. Continúe con el paso 3 si el procedimiento se completó de forma correcta.

Copiar la clave pública de forma manual

Si no tiene disponibilidad de acceso de SSH basado en contraseña a su servidor, deberá completar el proceso anterior de forma manual.

Habilitaremos el contenido de su archivo id_rsa.pub para el archivo ~/.ssh/authorized_keys en su máquina remota.

Para mostrar el contenido de su clave id_rsa.pub , escriba esto en su computadora local:

shell
$ cat ~/.ssh/id_rsa.pub

Verá el contenido de la clave, que debería tener un aspecto similar a este:

Salida por pantalla

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCqql6MzstZYh1TmWWv11q5O3pISj2ZFl9HgH
1JLknLLx44+tXfJ7mIrKNxOOwxIxvcBF8PXSYvobFYEZjGIVCEAjrUzLiIxbyCoxVyle7Q+bqgZ8See
M8wzytsY+dVGcBxF6N4JS+zVk5eMcV385gG3Y6ON3EG112n6d+SMXY0OEBIcO6x+PnUSGHrS
gpBgX7Ks1r7xqFa7heJLLt2wWwkARptX7udSq05paBhcpB0pHtA1Rfz3K2B+ZVIpSDfki9UVKzT8J
UmwW6NNzSgxUfQHGwnW7kj4jp4AT0VZk3ADw497M2G/12N0PPB5CnhHf7ovgy6nL1ikrygTKRF
mNZISvAcywB9GVqNAVE+ZHDSCuURNsAInVzgYo9xgJDW8wUw2o8U77+xiFxgI5QSZX3Iq7YL
MgeksaO4rBJEa54k8m5wEiEE1nUhLuJ0X/vh2xPff6SQ1BL/zkOhvJCACK6Vb15mDOeCSq54Cr7
kvS46itMosi/uS66+PujOO+xt/2FWYepz6ZlN70bRly57Q06J+ZJoc9FfBCbCyYH7U/ASsmY095ywP
sBo1XQ9PqhnN1/YOorJ068foQDNVpm146mUpILVxmq41Cj55YKHEazXGsdBIbXWhcrRf4G2fJLR
cGUr9q8/lERo9oxRm5JFX6TCmj6kmiFqv+Ow9gI0x8GvaQ== demo@test

Acceda a su anfitrión remoto usando el método que esté a su disposición.

Una vez que tenga acceso a su cuenta en el servidor remoto, debe asegurarse de que exista el directorio ~/.ssh. Con el siguiente comando se creará el directorio, si es necesario. Si este último ya existe, no se creará (el parámetro -p previene un mensaje de error si ya existe la carpeta):

shell
$ mkdir -p ~/.ssh

Ahora, podrá crear o modificar el archivo authorized_keys dentro de este directorio. Puede agregar el contenido de su archivo id_rsa.pub al final del archivo authorized_keys y, si es necesario, crear este último con el siguiente comando:

shell
$ echo public_key_string >> ~/.ssh/authorized_keys

En el comando anterior, reemplace public_key_string por el resultado del comando cat ~/.ssh/id_rsa.pub que ejecutó en su sistema local. Debería iniciar con ssh-rsa AAAA... .

Por último, verificaremos que el directorio ~/.ssh y el archivo authorized_keys tengan el conjunto

de permisos apropiados:

shell
$ chmod -R go= ~/.ssh

Con esto, se eliminan de forma recursiva todos los permisos “grupo” y “otros” del directorio ~/.ssh/.

Si está usando la cuenta root (usuario *raíz* o **administrador**) para configurar claves para una cuenta de usuario, también es importante que el directorio ~/.ssh/ pertenezca al usuario y no sea root:

shell
$ chown -R sammy:sammy ~/.ssh

En este tutorial, nuestro usuario recibe el nombre sammy pero debe sustituir el nombre de usuario que corresponda en el comando anterior.

Ahora podemos intentar la autenticación sin contraseña con nuestro servidor de Ubuntu.

Paso 3: Autenticación en el servidor de Ubuntu con claves de SSH

Si completó con éxito uno de los procedimientos anteriores, debería poder iniciar sesión en el host remoto *sin* la contraseña de la cuenta remota.

El proceso básico es el mismo:

shell
$ ssh username@remote_host

Si es la primera vez que establece conexión con este host (si empleó el último método anterior), es posible que vea algo como esto:

Salida por pantalla

The authenticity of host '203.0.113.1 (203.0.113.1)' can't be established.
ECDSA key fingerprint is fd:fd:d4:f9:77:fe:73:84:e1:55:00:ad:d6:6d:22:fe.
Are you sure you want to continue connecting (yes/no)? yes

Esto significa que su computadora local no reconoce el host remoto. Escriba “yes” y presione INTRO para continuar.

Si no proporcionó una frase de contraseña para su clave privada, se iniciará sesión de inmediato. Si proporcionó una frase de contraseña para la clave privada al crearla, se solicitará que la introduzca ahora (tenga en cuenta que, por motivos de seguridad, las pulsaciones de teclas no se mostrarán en la sesión de terminal). Después de la autenticación, se debería abrir una nueva sesión del intérprete de comandos con la cuenta configurada en el servidor de Ubuntu.

Si la autenticación basada en claves se realizó con éxito, puede aprender a proteger más su sistema inhabilitando la autenticación con contraseña.

Paso 4: Inhabilitar la autenticación con contraseña en su servidor

Si pudo iniciar sesión en su cuenta usando SSH sin una contraseña, habrá configurado con éxito la autenticación basada en claves de SSH para su cuenta. Sin embargo, su mecanismo de autenticación basado en contraseña sigue activo. Esto significa que su servidor sigue expuesto a ataques de fuerza bruta.

Antes de completar los pasos de esta sección, asegúrese de tener configurada la autenticación basada en claves de SSH para la cuenta root en este servidor o, preferentemente, la autenticación basada en clave de SSH para una cuenta no root en este servidor con privilegios sudo. Con este paso, se bloquearán los registros basados en contraseñas. Por lo tanto, es fundamental que se asegure de seguir teniendo acceso administrativo.

Una vez que haya confirmado que su cuenta remota tiene privilegios administrativos, inicie sesión en su servidor remoto con claves de SSH, ya sea como root o con una cuenta con privilegios sudo. Luego, abra el archivo de configuración del demonio de SSH:

```
$ sudo nano /etc/ssh/sshd_config
```

Dentro del archivo, busque una directiva llamada PasswordAuthentication. Puede insertar comentarios sobre esto. Elimine los comentarios de la línea y fije el valor en “no”. Esto inhabilitará su capacidad de iniciar sesión a través de SSH usando contraseñas de cuenta:

(Fichero) /etc/ssh/sshd_config

...

PasswordAuthentication

no
...

Guarde y cierre el archivo cuando haya terminado presionando CTRL + X, luego Y para confirmar la operación de guardado y, por último, INTRO para cerrar nano. Para implementar realmente estos cambios, debemos reiniciar el servicio sshd:

shell
$ sudo systemctl restart ssh

Como medida de precaución, abra una nueva ventana de terminal y compruebe que el servicio SSH funcione correctamente antes de cerrar esta sesión:

shell
$ ssh username@remote_host

Una vez que haya verificado su servicio SSH, podrá cerrar de forma segura todas las sesiones de servidores actuales.

El demonio de SSH de su servidor de Ubuntu ahora solo responderá a claves de SSH. La autenticación basada en contraseña se habrá desactivado con éxito.

Conclusion

De esta manera, la autenticación basada en claves de SSH debería quedar configurada en su servidor. Esto le permitirá iniciar sesión sin proporcionar una contraseña de cuenta.

Si desea obtener más información sobre cómo trabajar con SSH, consulte nuestra Guía de aspectos básicos de SSH.

Escrito por Hanif Jetha.

Editor: Lisa Tagliaferry.