Cómo utilizar las llaves SSH para conectarse sin una contraseña.

OpenSSH es una forma muy segura de conectarse de forma remota a una máquina Slackware. Pero la forma más fácil de usar SSH es, simplemente, usar sus llaves.

El concepto de claves públicas/privadas puede ser difícil de explicar, trataremos de hacerlo de la manera más simple posible.

Permítanme decir esto de nuevo, para todos ustedes, crypto nerds allí afuera: Sí, ya sé que esta es una versión muy simplificada de SSH. Esto se crea para todos los novatos SSH... mmmmkay?

Esencialmente, las llaves SSH se basan en la criptografía de clave pública. Esto significa que creas dos claves: una se llama llave pública y se usa para cifrar datos que solo usted puede descifrar. Puede dar su llave pública a cualquier persona, ya que su única función es cifrar datos - No hay mucho más que puedas hacer con él. La otra llave se llama llave PRIVADA, y es esta llave la que se utiliza para descifrar los datos cifrados con la llave pública.

Hasta ahora todo bien ... Ahora, ¿cómo se usa esto con SSH?

Cada vez que se comunique con una máquina Slackware (o cualquier máquina que ejecute OpenSSH, en realidad) a través del protocolo SSH, su SSH cliente (el programa, instalado en la computadora que tiene delante, que utiliza para conectarse) hablará con el SSH server instalado en la máquina distante. Determinarán en conjunto las capacidades que ambos pueden usar y la versión del protocolo que deben usar para comunicarse de forma segura.

Luego, intentarán determinar cómo usted (el usuario) iniciará sesión en la máquina remota. Si no se usan las claves, usualmente SSH (pero no siempre) predeterminará para pedirle una contraseña. Por otro lado, si se usan claves, las máquinas las usarán en el siguiente orden:

Then, they will try to determine how you (the user) will login on the remote machine. If keys are not used, SSH will usually (but not always) default to asking you a password. On the other hand, if keys are used, the machines are going to use them in the following order:

1. El servidor SSH cifrará un mensaje corto (técnicamente un valor hash) con su llave pública y lo enviará a su computadora.
2. Su cliente de SSH descifrará este mensaje con la clave privada (cuya única copia debe estar en su computadora) y lo enviará de vuelta al servidor de SSH.
3. El servidor de SSH estará satisfecho de que usted 'sea usted', por así decirlo, ya que teóricamente es la única persona capaz de descifrar el mensaje enviado y le otorgará acceso de inmediato.

Si todo esto parece un poco complicado, solo recuerda esto: tienes una clave pública y una privada. La clave pública debe estar en la computadora a la que desea acceder, o la computadora 'remota'. La clave privada debe estar en su computadora.

¡Vamos a pasar por este proceso paso a paso!

Crear el par de claves pública/privada

Para crear una clave pública y una privada, use la utilidad OpenSSH `ssh-keygen`. Esto generará automáticamente un par de claves, utilizando los valores predeterminados. Aquí hay un pequeño ejemplo:

```
noryungi@mypc:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/noryungi/.ssh/id_rsa): TEST.rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in TEST.rsa.
Your public key has been saved in TEST.rsa.pub.
The key fingerprint is:
1a:99:51:a6:12:69:53:aa:d8:f6:c2:56:66:6e:68:5a noryungi@udon
The key's randomart image is:
+--[ RSA 2048]----+
|    .o. o        |
|    +o +         |
|   .o.o          |
| o . . +         |
|. + + + S        |
| o B   o         |
|  E + .          |
| = o             |
|.                |
+-----------------+
noryungi@mypc:~$ ls TEST*
TEST.rsa  TEST.rsa.pub
```

OK, ¿qué está pasando aquí? Primero, `ssh-keygen` generará un par de claves. Hasta ahora, todo bien, asegúrese de leer el ssh-keygen man page entender todas las opciones, y hay muchas de ellas.

A continuación, `ssh-keygen` responderá que está creando el par de claves RSA (clave pública y privada). RSA es el nombre del algoritmo de cifrado utilizado. Hay tres tipos de cifrado posibles: DSA, RSA y ECDSA. Cuál es el mejor se deja como ejercicio para el lector ... 

Luego le preguntará dónde guardar la clave. Aquí se ingresa `TEST.rsa`, ya que hay otras claves en el sistema. Es importante dar un nombre propio a la clave, ya que hace que sea mucho más fácil recordar a qué se conecta cada clave.

Por ejemplo, si tuviera una cuenta en una máquina llamada: `stang.slackware.com`, un buen nombre para el par de claves sería `stang_slackware_com.rsa` o algo así.

A continuación, `ssh-keygen` le pide una frase de contraseña. ¡Siempre es una buena idea ingresar una frase de contraseña! Esto le permite proteger su clave privada, incluso si cae en las manos equivocadas. Si está absolutamente seguro al 100% de que su clave privada **no** va a caer en las manos equivocadas (¡qué optimista es!), Simplemente presione `Entrar` aquí.

El resto son solo mensajes informativos, y notará que el par de claves se ha guardado de la siguiente manera:

1. La clave privada se llama TEST.rsa.
2. La clave pública, la que desea copiar en la máquina remota, se llama TEST.rsa.pub

¡Felicidades! Estás a medio camino!

Configure su clave pública en la computadora remota

Muy bien, ahora, ¿cómo usar la clave pública/privada? Eso es bastante simple: copie la *clave pública* (llamada TEST.rsa.pub como hemos visto) en la computadora remota. La mejor manera de hacer esto es usar scp el programa de copia segura OpenSSH. Por ejemplo:

```
noryungi@mypc$ scp TEST.rsa.pub
nr@test.example.com:/home/nr/.ssh/authorized_keys
```

En el ejemplo anterior, copió la *clave pública* TEST.rsa.pub en la máquina remota llamada test.example.com, como usuario nr. El archivo cambia de nombre a authorized_keys, que es el nombre del archivo que contiene todas las claves públicas autorizadas para conectarse al servidor.

Una advertencia: ¡no ejecute el comando scp anterior si ya tiene un archivo de authorized_keys en la computadora remota! ¡¡Esto reemplazará el contenido del archivo con su clave pública!! Si ya tiene un archivo authorized_keys, ejecute un cat TEST.rsa.pub » authorized_keys en la máquina remota para agregar su clave pública al final de las claves autorizadas.

Dado que todas las claves SSH que usa deben colocarse en el directorio .ssh, ahí es donde va en la máquina remota.

Entonces, ¿hemos terminado? Realmente no, solo hay una pequeña cosa por hacer, pero es realmente importante, ya que es la fuente de muchos problemas...

Check the public key permissions on the remote machine

Since the private and public keys are very sensitive, they should be protected against prying eyes. To do this, on both the remote and the local machine, enter the following command:

```
nr@test.example.com$ chmod -R -v g-rwx,o-rwx ./.ssh/
mode of `./.ssh/' changed from 0755 (rwxr-xr-x) to 0700 (rwx------)
mode of `./.ssh/authorized_keys' changed from 0644 (rw-r--r--) to 0600 (rw-
------)
```

This command allows you to make sure nobody (except you and the SSH server) can read the public key.

Please note: if the permissions on the `authorized_keys` file OR the `.ssh` directory are not correct OpenSSH won't use the keys! If you have any problem with a public/private key, check the permissions and/or run the above command to make sure they are correct!

Connect using your newly created SSH key

Let's try to connect, back on the local machine, to the remote server named `test.example.com`:

```
noryungi@mypc$ ssh -i TEST.rsa nr@test.example.com
```

Please note that I have entered the option `-i` right after the `ssh` command: this option selects the private key to be used to connect, as user `nr`, to the remote server named `test.example.com`.

If you have chosen to protect the private key with a passphrase, `ssh` will ask you this passphrase before connecting. If not... Well, if the permissions are correct (see above), you should see the equivalent of the following:

```
nr@test.example.com$
```

That's it! You are connected to a remote machine, without ever entering a password, and with a much better security than with a password - which can be guessed, while a key is way too long to be ever guessed.

What could go wrong at this point?

Well, not much, really, except the possibility that your system administrator does not want you to connect with a public/private key pair...

In which case, let's face it, he is not a very good system administrator. This can be checked, however, with the following command on the remote machine:

```
nr@test.example.com$ grep -i pubkeyauth /etc/ssh/sshd_config
#PubkeyAuthentication yes
```

Please note the `#PubkeyAuthentication yes` line: this is the default value for public key authentication, and, as you can see, it is set to `yes`. You are good to go and use your key! On the other hand, if you see something like this:

```
nr@test.example.com$ grep -i pubkeyauth /etc/ssh/sshd_config
PubkeyAuthentication no
```

Then you are not allowed to connect to the remote machine (`test.example.com` above) with a public key. Time to contact your system administrator or your security administrator and request politely to be able to use them.

(Yes, there are other, more sneaky, ways to connect without entering a password, but none of them are as secure as a public/private key pair - maybe in another documentation on this wiki?)

You have reached the end of this short documentation - go forth, and use OpenSSH keys!

See Also

- The OpenSSH manual pages (online)

Sources

- Originalmente redactado por Noryungi
- Traducción al Español antares_alf

howtos, security, ssh, sshkeys, author noryungi