Clave pública en SSH HOWTO

Jaime Pérez Crespo (japecre at pantuflo.escet.urjc.es)

Última actualización: 11 de Julio de 2003

Motivación

Aquellos que trabajamos habitualmente con sesiones remotas usando ssh sabemos lo pesado que llega a ser tener que introducir la contraseña cada vez que nos identificamos en una máquina remota, más aún si trabajamos a la vez en varias máquinas, abriendo y cerrando sesiones. Por eso, sería útil el poder abrir sesiones ssh sin necesidad de estar tecleando continuamente la contraseña, máxime si nuestra contraseña es larga o complicada, pero, obviamente, sin perder la seguridad que nos garantiza utilizar ssh.

Esto es precisamente lo que aprenderemos a hacer en este mini-howto, utilizar un esquema de claves pública y privada que sustituya a la clásica contraseña, para conectar desde una máquina en la que podemos confiar, a otra máquina que ofrece el servicio de ssh.

Fundamentos técnicos

El método explicado aquí es muy sencillo, y se basa en el modelo criptográfico de cifrado asimétrico. Consiste, básicamente, en un par de claves que se corresponden unívocamente entre sí, generadas con cierto grado de aleatoriedad, y si se quiere, con una frase secreta que sólo conoce el dueño de las mismas, que permite cifrar las claves con el algoritmo 3DES. La clave pública es, como su nombre indica, de libre utilización por cualquier persona. Cuando alguien quiere encriptar un mensaje que sólo nosotros podamos leer, utilizará nuestra clave pública para hacerlo. Posteriormente, nosotros usaremos nuestra clave privada (que solamente su dueño conoce) para desencriptar ese mensaje. Cualquier cosa encriptada con una clave pública, sólo se podrá desencriptar con su par privada. El proceso inverso, es decir, encriptar usando la clave privada y desencriptar usando la clave pública, se utiliza para las firmas digitales.

Este modelo es muy conocido y utilizado para criptografía en general, y aunque no lo es tanto en el uso que nos ocupa, si es muy útil y eficiente. Para más información sobre el modelo de cifrado asimétrico consultar las páginas referenciadas en enlaces interesantes.

Generación de las claves

Lo primero que tenemos que hacer es generar un par de claves, que serán las que utilicemos para identificarnos. Generaremos un par de claves RSA, y otro DSA. El programa que las genera nos preguntará primero dónde queremos guardarlas. Todo nuestro trabajo será en el directorio .ssh que cuelga de nuestro home, por lo tanto, creamos dicho directorio (en caso de que no exista), y procedemos a utilizar el generador de claves. Los nombres para las claves suelen ser id_rsa e id_rsa.pub para las claves RSA privada y pública, respectivamente, e id_dsa e id_dsa.pub para las claves DSA.

```
[icarus@mordor icarus]$ ls -al | grep ".ssh"
[icarus@mordor icarus]$ mkdir ~/.ssh
[icarus@mordor icarus]$ chmod 700 ~/.ssh
[icarus@mordor icarus]$ ssh-keygen -b 1024 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/icarus/.ssh/id_rsa): ENTER
Enter passphrase (empty for no passphrase): ENTER
[icarus@mordor icarus]$ ssh-keygen -b 1024 -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/icarus/.ssh/id_dsa): ENTER
Enter passphrase (empty for no passphrase): ENTER
```

Esto habrá creado las claves públicas y privadas RSA y DSA y las habrá almacenado en nuestro directorio ~/.ssh. El motivo de no haber indicado ninguna passphrase es simple. En nuestro caso, queremos que el login se realice de forma automática, por lo que si indicásemos una frase, ésta se nos pediría cada vez que iniciásemos una sesión de ssh, con lo cual no habríamos hecho nada. Pero el no indicar la passphrase tiene

un problema, y es que si alguien consigue de alguna manera nuestras claves privadas, podrá entrar en el sistema haciéndose pasar por nosotros. Ni que decir tiene, por tanto, que debemos mantener bien seguras nuestras claves privadas, de modo que no es necesaria explicación alguna de porqué damos permisos rwx------ a nuestro directorio .ssh.

Este sistema, por tanto, no es aconsejable utilizarlo en ordenadores en los que no tengamos una confianza plena en la seguridad de nuestras claves privadas. No obstante, si aún así lo usamos y tenemos la sospecha de que alguien ha podido robar nuestras claves privadas, deberemos eliminar inmediatamente dichas claves, y regenerarlas de nuevo utilizando el procedimiento explicado.

Preparación del entorno

La configuración del servidor y del ordenador seguro desde el que queremos acceder es sencilla. En el servidor, han de encontrarse las claves públicas que hemos generado, todas ellas juntas en un archivo llamado authorized_keys, almacenado bajo nuestro directorio ~/.ssh. Por contra, en el ordenador desde el que conectaremos debemos tener almacenadas las claves privadas, esto es, los ficheros id_rsa e id_dsa, bajo nuestro directorio ~/.ssh. Es por esto que es tan importante tener una buena seguridad en esta máquina, puesto que en ella almacenaremos nuestras credenciales.

Preparamos entonces el archivo authorized_keys que necesitaremos colocar en el servidor, y lo subimos mediante ftp a través de un túnel ssh, algo más sencillo, como scp.

```
[icarus@mordor icarus]$ cat ~/.ssh/id_rsa.pub ~/.ssh/id_dsa.pub > ~/.ssh/authorized_keys
[icarus@mordor icarus]$ scp ~/.ssh/authorized_keys japecre@pantuflo.escet.urjc.es/tmp
[...]
japecre@pantuflo.escet.urjc.es's password: password del servidor
```

Esto crea un archivo authorized_keys en el directorio .ssh de nuestro home, y lo copia de forma segura al servidor, en este caso llamado pantuflo.escet.urjc.es, en el que tenemos una cuenta llamada japecre. Tras pedirnos la contraseña y verificarla, el archivo se copia en /tmp/authorized_keys del servidor remoto. Una vez hecho esto, entramos al servidor via ssh con nuestra cuenta, y movemos el archivo en cuestión a su ubiación definitiva. Si no existe el directorio .ssh, lo creamos, y le damos los permisos necesarios.

```
[icarus@mordor icarus]$ ssh japecre@pantuflo.escet.urjc.es
japecre@pantuflo.escet.urjc.es's password: password del servidor
japecre@pantuflo:japecre$ mkdir ~/.ssh
japecre@pantuflo:japecre$ chmod 700 ~/.ssh
japecre@pantuflo:japecre$ mv /tmp/authorized_keys ~/.ssh/
```

Ya tenemos todo configurado y listo para probar nuestro nuevo sistema de autenticación. No tenemos más que salir del servidor y volver a conectarnos por ssh. Si todo hay ido bien, esta vez no nos preguntará nuestra contraseña y entraremos directamente.

Enlaces Interesantes

Sistemas de cifrado asimétrico: http://www.gnupg.org/gph/es/manual/x212.html
Características de OpenSSH: http://openbsd.bug.it/openssh/es/features.html
Seguridad en OpenSSH: http://openbsd.unixtech.be/openssh/es/security.html

Licencia

Este documento público puede ser distribuido y/o traducido libremente, siempre y cuando en el documento resultante se incluyan los datos del autor indicados en la cabecera, esta sección de licencia, y la url del documento original:
http://pantuflo.escet.urjc.es/~japecre/trabajos/independientes/docs/ssh.php?idioma=es