



Tip: SSH, SCP, y un as bajo la manga

Publicado el [12 Enero, 2007](#) en [Linux](#), [Tips](#).

Entre vuelta y vuelta, contando el servidor de la oficina, el y uno que tengo en mi casa como laboratorio de experimentos, yo diría que no hay minuto del día en que no tenga una conexión remota abierta en el terminal.

Pero no estoy hablando de FTP, ni SaMBa ni nada de eso, sino de algo muchísimo mejor. Algo que no sólo te permite copiar archivos de un lado a otro, sino también abrir programas, instalar paquetes e incluso . Algo que, simplemente, cambió mi vida (en broma& aunque ni tanto!).

¿La maravilla? SSH.

SSH

SSH significa **Secure SHell**, y es la vía por la cual te conectas remotamente a un tarro con Linux para acceder a un terminal de comandos (shell), uno igual al que usas en tu PC pero remotamente. La información viaja encriptada por lo que además de ser útil es seguro, y claro, mucho más rápido que FTP si quieres hacer algo como instalar y configurar un Wordpress +plugins.

Otro buen ejemplo de SSH es cuando te llama un amigo para pedirte que le ayudes con su recién instalado Ubuntu. ¿Qué le dices?

Compadre, échele un vistazo a [un post](#) que encontré por ahí& es bastante malo pero algo ayuda. Después de eso instala un paquete llamado **openssh-server** por Synaptic, y me creas una cuenta de usuario en la Administración.

Listoco. Te mando mi IP también.

Perfecto. Ahora para conectarte remotamente por SSH, la sintaxis del comando es ésta:

```
$ ssh usuario@NU.ME.RO.IP
```

O bien:

```
$ ssh usuario@servidor.com
```

Obviamente el usuario con que te conectas no es el del tarro que estás usando, sino de la cuenta que te creó tu amigo allá. Ahora, si el usuario es el mismo en ambos lados, puedes omitirlo:

```
$ ssh servidor.com
```

Una vez que estés conectado, puedes ejecutar comandos al igual que si abrieras un terminal en el mismo computador. Prueba con algo simple como **uname -a** o lo que quieras. Después, para cerrar la conexión basta con un simple **exit** y listo.

También puedes ejecutar un comando único vía SSH, sin necesariamente loggarte al sistema:

```
$ ssh servidor.com aplicacion
```

O sea:

```
$ ssh bootlog.cl uname -a
```

Y por último, puedes ejecutar una aplicación X en el sistema remoto pero utilizando el display local. Sólo tienes que agregar **-X** después de SSH:

```
$ ssh -X servidor.com aplicacion
```

¿Todo bien? Todo bien. Prosigamos.

SCP

SCP significa **Secure CoPy** es una pata más de SSH, que permite transferir archivos o carpetas entre computadores. La sintaxis es bien simple:

```
$ scp archivo usuario@servidor.com:ruta
```

Y para copiar a la inversa, desde el computador remoto al tuyo, simplemente tienes que invertir el orden de los elementos:

```
$ scp usuario@servidor.com:ruta/archivo ruta_local
```

O sea por ejemplo, si quisiéramos mandar algo al servidor:

```
$ scp hola.txt tomas@bootlog.cl:/www/sitio
```

Mandaría el archivo **hola.txt** y lo dejaría en la carpeta **/www/sitio** en el servidor **bootlog.cl**.

También puedes mandar carpetas completas (con **:**):

```
$ scp -r viajealsur/ tomas@bootlog.cl:/www/sitio/fotos
```

Así mandarí la carpeta **/viajealsur** completa a **/www/sitio/fotos**. En acción:

```
Terminal
tomas@semos: /home/tomas
tomas@dev: /www/aardvark.cl

lwxlwxlwx 1 tomas tomas 11 2006-11-01 20:00 docs -> /media/docs/
dwx----- 3 tomas tomas 4.0K 2006-07-21 16:53 mozilla/
dwxr-xr-x 2 tomas tomas 4.0K 2007-01-08 19:54 varios/
dwxlwx-- 2 tomas tomas 4.0K 2007-01-12 12:38 viajealsur/

tomas@supernigga :: [-]$ date
Fri Jan 12 12:47:21 CLST 2007
tomas@supernigga :: [-]$ scp -r viajealsur/ bootlog.cl:/www/sitio/fotos
Password:
00095.jpg 100% 1077KB 1.1MB/s 00:00
00094.jpg 100% 1052KB 1.0MB/s 00:01
00093.jpg 100% 1116KB 1.1MB/s 00:00
00092.jpg 100% 948KB 947.5KB/s 00:00
00091.jpg 100% 889KB 888.7KB/s 00:00
00090.mov 100% 1875KB 1.8MB/s 00:00
00089.mov 100% 730KB 730.4KB/s 00:00
00088.jpg 100% 998KB 997.6KB/s 00:00
00087.jpg 100% 1088KB 1.1MB/s 00:00
00086.jpg 100% 944KB 943.5KB/s 00:00
00085.jpg 100% 944KB 943.5KB/s 00:00
00084.jpg 100% 1111KB 1.1MB/s 00:00
tomas@supernigga :: [-]$
```

Ahora, el mismo proceso a la inversa sería:

```
$ scp -r tomas@bootlog.cl:/www/sitio/fotos/viajealsur ~
```

Esto copiaría la carpeta **viajealsur/** del servidor a mi carpeta **/home**.

Ah, y a propósito de carpetas home, como en SCP (y en SSH) la ruta por defecto es tu carpeta de usuario, si quieres copiar algo hacia o desde ahí, puedes omitir la ruta:

```
$ scp cancion.mp3 bootlog.cl:
```

En este caso la sintaxis está bien, ya que mi usuario se llama **tomas** en ambos lados, así que no tengo que escribirlo de nuevo, y como estoy copiando el archivo **cancion.mp3** a mi carpeta de usuario, también puedo omitir la ruta.

Ahora llegó la hora de que tú hagas la prueba. Recuerda que tienes que instalar el paquete **openssh-server** en el PC remoto para poder conectarte.

El tip: Cómo conectarte por SSH o SCP sin contraseña

Si eres como yo, probablemente siempre te estás conectando a los mismos tarros, y cuando se corta la conexión tienes que escribir la clave una y otra vez. O quizás quieres configurar un **cron job** que ejecute un comando remotamente o mande un archivo cada noche a cierta hora (algo que la petición de contraseña te impide).

Con este truco puedes decirle al PC remoto que **tal usuario** está autorizado para conectarse sin contraseña **desde tal computador**, algo que hacemos a través de las llaves RSA. ¿Qué significa eso? Da lo mismo, por ahora lo importante es que lo hagamos funcionar!

Así dice el cuento.

```
$ ssh-keygen -t rsa
```

Este comando genera dos archivos, **id_rsa.pub** e **id_rsd**. Necesitamos copiar el contenido del primero al otro computador, específicamente al archivo **authorized_keys** dentro de **~/.ssh/** Primero

copiémoslo por SCP.

```
$ scp id_rsa.pub usuario@servidor:
```

Ahora conéctate por SSH y ve si existe el archivo **authorized_keys** dentro de **~/.ssh/**. Si ese no es el caso, escribe:

```
$ mv id_rsa.pub .ssh/authorized_keys
```

Si el archivo existía (que significa que para ese usuario en ése PC, hay otros tarros de confianza). Vamos a copiar los contenidos de **id_rsa.pub** a **authorized_keys**:

```
$ cat id_rsa.pub >> .ssh/authorized_keys
```

Eso es todo! Ahora veamos si funciona, ejecutando un comando remotamente a través de SSH (en este caso, **uname -a**). Obviamente no nos debería pedir la clave.

```
$ ssh servidor.com uname -a
```

¿Todo bien? ¡Excelente! En este preciso instante acabas de hacer tu vida mucho, mucho más fácil. Si no me crees, ya lo verás por tí mismo. :)