SSH

Contenido

[SSH 1](#_Toc525035555)

[Criptografía 1](#_Toc525035556)

[Establecimiento de una conexión TCP 3](#_Toc525035557)

[Intercambio de claves 3](#_Toc525035558)

[Instalación y configuración 4](#_Toc525035559)

[Instalación 4](#_Toc525035560)

[Configuración 4](#_Toc525035561)

[Reiniciar el servicio 5](#_Toc525035562)

[Utilización del servicio SSH 5](#_Toc525035563)

[Administrar otra máquina desde Linux 5](#_Toc525035564)

[Control de la máquina remota 6](#_Toc525035565)

[Administrar otra máquina desde Windows 7](#_Toc525035566)

[Intercambiar archivos en linux 8](#_Toc525035567)

[Por scp 8](#_Toc525035568)

[Por sftp 9](#_Toc525035569)

# SSH

SSH (Secure Shell) es una herramienta que permite realizar conexiones seguras entre equipos unidos mediante una red insegura como puede ser, por ejemplo, Internet. Utiliza el puerto 22 y sigue el modelo cliente. servidor.

Funciones:

* Administración remota
* Intercambio de ficheros seguro y cifrado.
* Otras funciones como administración gráfica, túneles . . .

La aportación más importante de la herramienta SSH es que da soporte seguro a cualquier protocolo que funcione sobre TCP. Dicha seguridad se basa en la utilización de mecanismos de criptografía, de forma que toda transmisión de información es cifrada y el mecanismo de autenticación es transparente al usuario

## Criptografía

La criptografía (del [griego](http://es.wikipedia.org/wiki/Idioma_griego) κρύπτω krypto, «oculto», y γράφω graphos, «escribir», literalmente «escritura oculta») es el [arte](http://es.wikipedia.org/wiki/Arte) o [ciencia](http://es.wikipedia.org/wiki/Ciencia) de cifrar y descifrar [información](http://es.wikipedia.org/wiki/Información) mediante técnicas especiales y se emplea frecuentemente para permitir un intercambio de mensajes que sólo puedan ser leídos por personas a las que van dirigidos y que poseen los medios para descifrarlos.

En la jerga de la criptografía, la información original que debe protegerse se denomina texto en claro o [texto plano](http://es.wikipedia.org/wiki/Texto_plano). El cifrado es el proceso de convertir el texto plano en un galimatías ilegible, denominado texto cifrado o criptograma. Por lo general, la aplicación concreta del algoritmo de cifrado (también llamado cifra) se basa en la existencia de una clave: información secreta que adapta el algoritmo de cifrado para cada uso distinto. Cifra es una antigua palabra arábiga para designar el número cero; en la Antigüedad, cuando Europa empezaba a cambiar del sistema de numeración romano al arábigo, se desconocía el cero, por lo que este resultaba misterioso, de ahí probablemente que cifrado signifique misterioso.

Las dos técnicas más sencillas de cifrado, en la criptografía clásica, son la sustitución (que supone el cambio de significado de los elementos básicos del mensaje -las letras, los dígitos o los símbolos-) y la transposición (que supone una reordenación de los mismos); la gran mayoría de las cifras clásicas son combinaciones de estas dos operaciones básicas.

El descifrado es el proceso inverso que recupera el texto plano a partir del criptograma y la clave. El protocolo criptográfico especifica los detalles de cómo se utilizan los algoritmos y las claves (y otras operaciones primitivas) para conseguir el efecto deseado. El conjunto de protocolos, algoritmos de cifrado, procesos de gestión de claves y actuaciones de los usuarios, es lo que constituyen en conjunto un criptosistema, que es con lo que el usuario final trabaja e interactúa.

Existen dos grandes grupos de cifras: los algoritmos que usan una única clave tanto en el proceso de cifrado como en el de descifrado, y los que emplean una clave para cifrar mensajes y una clave distinta para descifrarlos. Los primeros se denominan cifras simétricas, de clave simétrica o de clave privada, y son la base de los algoritmos de cifrado clásico. Los segundos se denominan cifras asimétricas, de clave asimétrica o de clave pública y forman el núcleo de las técnicas de cifrado modernas.

La criptografía simétrica es un método [criptográfico](http://es.wikipedia.org/wiki/Criptografía) en el cual se usa una misma clave para cifrar y descifrar mensajes. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez ambas tienen acceso a esta clave, el remitente cifra un mensaje usándola, lo envía al destinatario, y éste lo descifra con la misma.

El principal problema con los sistemas de cifrado simétrico no está ligado a su seguridad, sino al intercambio de claves. Una vez que el remitente y el destinatario hayan intercambiado las claves pueden usarlas para comunicarse con seguridad, pero ¿qué canal de comunicación que sea seguro han usado para transmitirse las claves? Sería mucho más fácil para un atacante intentar interceptar una clave que probar las posibles combinaciones del [espacio de claves](http://es.wikipedia.org/w/index.php?title=Espacio_de_claves&action=edit&redlink=1).

Otro problema es el número de claves que se necesitan. Si tenemos un número n de personas que necesitan comunicarse entre sí, se necesitan n/2 claves para cada pareja de personas que tengan que comunicarse de modo privado. Esto puede funcionar con un grupo reducido de personas, pero sería imposible llevarlo a cabo con grupos más grandes

La criptografía asimétrica es el método [criptográfico](http://es.wikipedia.org/wiki/Criptografía) que usa un par de claves para el envío de mensajes. Las dos claves pertenecen a la misma persona a la que se ha enviado el mensaje. Una clave es pública y se puede entregar a cualquier persona, la otra clave es privada y el propietario debe guardarla de modo que nadie tenga acceso a ella. Además, los métodos criptográficos garantizan que esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves.

Si el remitente usa la clave pública del destinatario para cifrar el mensaje, una vez cifrado, sólo la clave privada del destinatario podrá descifrar este mensaje, ya que es el único que la conoce. Por tanto se logra la confidencialidad del envío del mensaje, nadie salvo el destinatario puede descifrarlo.

Si el propietario del par de claves usa su clave privada para cifrar el mensaje, cualquiera puede descifrarlo utilizando su clave pública. En este caso se consigue por tanto la identificación y autentificación del remitente, ya que se sabe que sólo pudo haber sido él quien empleó su clave privada (salvo que alguien se la hubiese podido robar). Esta idea es el fundamento de la [firma electrónica](http://es.wikipedia.org/wiki/Firma_electrónica).

Los sistemas de cifrado de clave pública o sistemas de cifrado asimétricos se inventaron con el fin de evitar por completo el problema del intercambio de claves de los [sistemas de cifrado simétricos](http://es.wikipedia.org/wiki/Criptografía_simétrica). Con las claves públicas no es necesario que el remitente y el destinatario se pongan de acuerdo en la clave a emplear. Todo lo que se requiere es que, antes de iniciar la comunicación secreta, el remitente consiga una copia de la clave pública del destinatario. Es más, esa misma clave pública puede ser usada por cualquiera que desee comunicarse con su propietario. Por tanto, se necesitarán sólo n pares de claves por cada n personas que deseen comunicarse entre sí.

SSH utiliza los algoritmos RSA y DSA de cifrado con clave pública/privada. DSA es más lento que RSA pero más seguro.

## Establecimiento de una conexión TCP

SSH para trabajar necesita establecer una conexión TCP de la siguiente manera:

1. El cliente envía un SYN al servidor.
2. El servidor le responde con un SYN ACK.
3. El cliente envía un ACK.

Existe un número de secuencia generado e intercambiado por cada lado, ayudando de este modo a que no se puedan establecer conexiones falseadas

## Intercambio de claves

Una vez establecida la conexión TCP y antes de que se abra la sesión, ocurrirá lo siguiente:

1. Los dos se envían el banner (empezando por el servidor) y se comprueba que las versiones son compatibles.
2. Si es así se intercambian las claves, el cliente primero. El método usado para el intercambio es el [Diffie-Hellman](http://es.wikipedia.org/wiki/Diffie-Hellman) .
3. Se inicia la comunicación cifrada.
4. Para desconectarse el cliente envía un ACK.

# Instalación y configuración

## Instalación

**#apt-get install ssh**

## 

## Configuración

El fichero de configuración del servidor SSH se encuentra en la mayoría de sistemas en /etc/ssh/sshd\_config. Antes de modificarlo conviene hacer una copia de seguridad del mismo. Para editarlo se escribirá en la consola:

**#nano /etc/ssh/sshd\_config**

En este fichero aparecerán diferentes directivas que se podrán habilitar o no mediante el carácter # y que seguirán el siguiente formato:

**directiva valor\_en\_minúsculas**

### Parametro port

Suele ser habitual que los script-kidies se dediquen a buscar puertos 22 abiertos e intentar ataques de diccionario o fuerza bruta. Para evitar deberemos cambiar el servidor de puerto así:

**port puerto**

Por ejemplo lo movemos al 1022

**port 1022**

### Parámetro Protocol

Se debe poner siempre e indica la versión SSH que se está utilizando.

**Protocol 2**

### 

### Parámetro PermitRootLogin

Quizás el más importante en cuanto a seguridad, permite o impide que el root se conecte. Así en caso de algún fallo el intruso solo conseguiría acceso como usuario y tendría que explotar otra vulnerabilidad para hacerse root.

**PermitRootLogin yes,no**

### Parámetro AllowUsers

También muy importante. Permite restringir el acceso por usuario. Por ejemplo:

**AllowUsers usuario1 usuario2@IP**

De esta manera solo usuario1 y usuario2 podrán loguearse en el servidor. Ademas usuario2 solo podrá conectarse desde IP

### 

### Parametro LoginGraceTime

Establece el tiempo en segundos que tiene el cliente para loguearse

**LoginGraceTime 30**

### Parametro MaxAuthTries

Establece el número máximo de intentos de login

**MaxAuthTries 2**

### Parametro MaxStartups

Establece el número máximo de conexiones simultáneas

**MaxStartups 5**

Establece el número de intentos erróneos

### Texto de Bienvenida

Se escribirá en el archivo /var/run/motd

## 

## Reiniciar el servicio

Después de hacer cualquier modificación deberemos reiniciar el servicio con el comando:

**/etc/init.d/ssh restart**

**o**

**service ssh restart**

Si este comando no funciona significa que hay algún error en el fichero de configuración.

También tenemos las opciones de parar (stop), iniciar (start) o estado (status).

Para que el servicio se inicie en el arranque del sistema, habrá que asegurarse que aparece en el directorio /etc/rc2.d. Si no, podremos ejecutar:

**ln /etc/rc2.d/ /etc/init.d/ssh**

# 

# Utilización del servicio SSH

## Administrar otra máquina desde Linux

Una vez puesto en marcha el servidor, nos conectaremos a él para poder trabajar como si estuviéramos logueados en local. Para ello, el comando que utilizaremos será:

**ssh –p puerto usuario@maquina**

Donde *–p puerto* indica el puerto que hemos configurado en el servidor (22 por defecto), *usuario* el usuario local del servidor con el que nos queremos loguear y *maquina* la IP o el nombre de dominio del servidor al que nos queremos conectar.

En el siguiente ejemplo, nos conectaríamos al servidor ssh con IP 192.168.2.100 por el puerto 22.

**ssh 192.168.2.100**

En el siguiente ejemplo, nos conectaríamos al servidor ssh con nombre de dominio blas.barriosesamo.com, con el usuario epi y por el puerto 22. Para ello, deberemos tener correctamente configurado el DNS.

**ssh epi@blas.barriosesamo.com**

En el siguiente ejemplo, nos conectaríamos al servidor ssh con IP 192.168.2.100, con el usuario sm2 y por el puerto 1984.:

**ssh –p 1984 sm2@192.168.2.100**

**ms2@192.168.2.100´s password:**

**MENSAJE DE BIENVENIDA**

**ms2@icjardin:~$**

El prompt habrá cambiado y nos mostrará el usuario (sm2) con el que nos hemos logueado y el equipo (icjardin) al que hemos accedido. Si el usuario con el que nos hemos logueado es un usuario normal, únicamente podremos administrar nuestra carpeta personal y utilizar las aplicaciones comunes a todos los usuarios. Sin embargo, si el usuario tiene permisos de administración podremos controlar completamente la máquina como se explica en el apartado siguiente.

## 

## Control de la máquina remota

Una vez establecido el acceso al equipo remoto, si el usuarios tiene permisos de administración, podremos realizar cualquier tarea de administración ejecutando el comando correspondiente con *sudo* por delante. Por ejemplo, podremos reiniciar el equipo del siguiente modo:

**ms2@192.168.2.50:~$ sudo reboot**

**Broadcast message from ms2@192.168.2.50**

**(/dev/pts/1) at 12:32 …**

**The system is going down for reboot NOW!**

O ejecutar el comando de apagado:

**ms2@192.168.2.50:~$ sudo halt**

**Broadcast message from ms2@192.168.2.50**

**(/dev/pts/1) at 12:37 …**

**The system is going down for halt NOW!**

O crear y administrar usuario y carpetas:

**ms2@192.168.2.50:~$ sudo addusr user1**

**ms2@192.168.2.50:~$ sudo chmod 777 /srv/ftp/publico**

De igual modo, podremos editar y cambiar cualquier archivo de configuración y luego reiniciar el servicio. Por tanto, cualquier servicio (DNS, DHCP, HTTP, FTP…) se pueden instalar y administrar a través de SSH. Esto permite subcontratar estas tareas a empresas externas ubicadas en cualquier parte el mundo.

**ms2@192.168.2.50:~$ sudo apt-get install dhcp3**

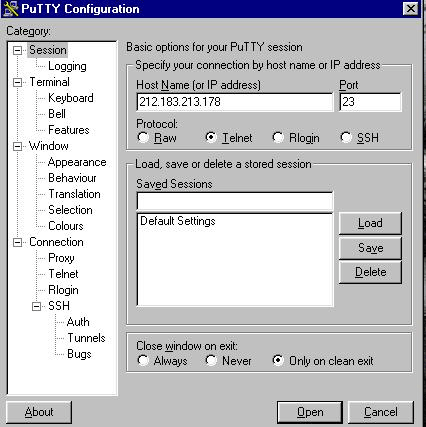
**ms2@192.168.2.50:~$ sudo gedit /etc/dhcp3/dhcpd.conf**

**ms2@192.168.2.50:~$ sudo /etc/init.d/dhcp3 restart**

## 

## Administrar otra máquina desde Windows

Para conectarnos vía ssh desde un equipo Windows se utilizan aplicaciones gráficas como putty donde configuraremos puerto de conexión, usuario y dirección IP o nombre DNS como se ve en la imagen siguiente:



## 

## Intercambiar archivos en linux

### Por scp

Una de las funcionalidades más utilizada por los administradores de redes es la transferencia de ficheros. Esta se puede realizar vía FTP como se vio en el tema anterior. Sin embargo, si queremos hacerlo de manera segura existe recurriremos al protocolo SCP (Secure Copy) que se apoya en el servidor SSH.

Para utilizarlo, se ejecutará el comando scp del siguiente modo:

**scp -r –P puerto u1@origen:/ruta/archivo/origen u2@destino:/ruta/archivo/destino**

Donde:

* *-r* indica recursividad, se copiará la carpeta y todo lo que haya dentro
* *–P* puerto indica el puerto SSH al que nos conectamos
* *u1*: especifica el usuario que se conectará a la máquina origen
* *origen*: indica la IP o el nombre DNS del equipo desde el que queremos copiar un archivo
* */ruta//archivo/origen* es la ruta del archivo que queremos copiar dentro del equipo origen
* *u2*: especifica el usuario que se conectará a la máquina destino
* *destino*: indica la IP o el nombre DNS del equipo al que queremos copiar el archivo
* */ruta/archivo/destino* es la ruta donde queremos copiar el archivo dentro del equipo destino

En el ejemplo siguiente copiamos el fichero kaixo.txt que está en la carpeta /home/ms2 del equipo local al equipo con IP 192.168.2.55 con el nombre agur.txt y en la carpeta /home/user1. Nos conectamos a través del puerto SSH 1984 con el usuario user1.

**scp –P 1984 /home/ms2/kaixo.txt user1@192.168.2.55:/home/user1/agur.txt**

Al conectarnos al equipo 192.168.2.55, nos pedirá la contraseña del usuario user1 y realizará la operación:

**user1@192.168.2.55’s password:**

**agur.txt 100% 27KB 27KB/s 00:01**

En el ejemplo siguiente copiamos el fichero kaixo.txt que está en el equipo con IP 192.168.2.55, en la carpeta personal de user1, a la carpeta /home/ms2 del equipo local con el nombre agur.txt. Nos conectamos a través del puerto SSH 22 con el usuario user1.

**scp user1@192.168.2.55:kaixo.txt /home/sm2/agur.txt**

### Por sftp

Otra manera de transferir archivos de modo seguro es utilizar el protocolo SFTP (Secure File Transfer Protocol). A pesar de su nombre no utiliza FTP, sino que se apoya en el servidor SSH. Como *scp*, es un comando independiente que se ejecuta sin habernos conectado antes vía ssh.

Para utilizarlo, se ejecutará el comando sftp del siguiente modo:

**sftp -o port=1984 usuario@maquina**

Una vez conectados, podremos utilizar los comandos GET y PUT para transferir archivos desde la maquina local a la remota o viceversa.