Qué es el protocolo SSH y cómo configurarlo para mejorar la seguridad de acceso a los servidores Linux

Cardenal Gardoki, 1
48008 BILBAO (Vizcaya)
Teléfono: 902 012 199
www.hostalia.com

HOSTALIA .com

Cuando uno contrata un servidor dedicado o un VPS con una empresa de alojamiento web, como puede ser Hostalia, debe utilizar algún tipo de mecanismo para conectarse con él remotamente y poder interactuar con él como si lo tuviera al lado. Debido a la importancia de los datos que se suelen mover en este tipo de máquinas, es fundamental contar con algún protocolo de conexión que nos **garantice total seguridad a la hora de comunicarnos con el servidor**. Esta seguridad nos la proporciona el **protocolo SSH (Secure Shell)**, probablemente el sistema más utilizado actualmente para comunicarse entre equipos con sistema operativo Linux.

Hasta la aparición de este protocolo, la forma que utilizaban los usuarios para conectarse con los servidores era por medio del uso de otros protocolos como Telnet o FTP, protocolos que tienen una gran desventaja respecto al SSH, ya que en este último **se encripta la sesión de conexión**, impidiendo que alguien pueda capturar la contraseña no encriptada.

Un poco de historia

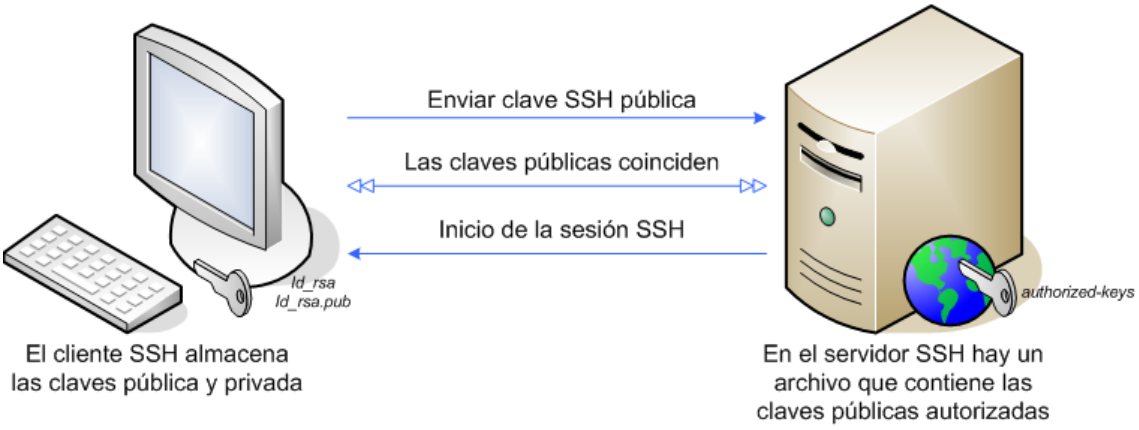
El protocolo SSH **fue desarrollado en el año 1995 por el finlandés Tatu Ylönen**, quien publicó su trabajo bajo una licencia de libre uso, pero ante el éxito del programa desarrollado pronto registró la marca SSH y fundó la empresa SSH Communications Security con fines comerciales, la cual permitía el uso del protocolo gratuitamente para uso doméstico y educativo.

Ante este cambio en la política de uso del protocolo, los desarrolladores del sistema operativo OpenBSD empezaron a desarrollar en el año 1999 una versión libre de este protocolo que recibió el nombre de OpenSSH.

Desde la aparición de este protocolo han sido dos las versiones que han estado activas. En la primera de ellas, se ofrecía una alternativa a las sesiones interactivas mediante el uso de herramientas como TELNET, RSH o RLOGIN entre otras, sin embargo pronto se descubrió que este protocolo tenía un punto débil que permitía a los hackers introducir datos en los flujos cifrados.

Ante este problema, en el año 1997 fue lanzada la versión 2, donde una serie de medidas solucionaban el problema descubierto en la primera versión. Además de corregir ese problema, esta **segunda versión incorporaba el protocolo SFTP (Secure File Transfer Protocol** – Protocolo seguro de transferencia de archivos) que proporciona la funcionalidad necesaria para la transferencia y manipulación de archivos de forma segura.

Cómo funciona el protocolo SSH



El funcionamiento de este protocolo se puede resumir en los siguientes pasos que os dejamos a continuación:

1. **El cliente inicia una conexión TCP sobre el puerto 22 del servicio**. Este puerto es el que utiliza por defecto el protocolo, aunque como veremos en siguientes puntos, se puede modificar.

2. El **cliente y el servidor se ponen de acuerdo en la versión del protocolo a utilizar**, así como el algoritmo de cifrado utilizado para el intercambio de la información.

3. El **servidor**, que tiene en su poder dos claves (una privada y una pública), **manda su clave pública al cliente**.

4. Cuando el cliente recibe la clave enviada por el servidor, la compara con la que tiene almacenada para verificar su autenticidad. El **protocolo SSH exige que el cliente la confirme la primera vez**.

5. Con la clave pública del servidor en su poder, **el cliente genera una clave de sesión aleatoria**, creando un mensaje que contiene esa clave y el algoritmo seleccionado para la encriptación de la información. Toda esa información es enviada al servidor haciendo uso de la clave pública que envió en un paso anterior de forma cifrada.

6. Si todo es correcto, el cliente queda autenticado, iniciando la sesión para comunicarse con el servidor.

Características del servicio SSH

El uso del protocolo **Secure Shell** por parte de los usuarios, ofrece una serie de interesantes características, que lo ha llevado a convertirse en el método más utilizado por todos los usuarios que gestionan algún tipo de servidor Linux ya sea en la nube o dedicado. Algunas de estas características que podemos destacar son:

- El uso de **SSH encripta la sesión** de registro impidiendo que cualquier persona pueda conseguir contraseñas no encriptadas.
- Las claves de encriptación utilizadas sólo son conocidas por quien emite la información y por quien la recibe.
- **Una sola alteración de la clave modifica el mensaje original**, lo que permite que si por alguna razón un tercero descubre la clave no acceda al mensaje completo.
- El usuario tiene la posibilidad de verificar que sigue conectado al mismo servidor que se conectó inicialmente.
- Cuando un usuario se autentica, entre él y el servidor se crea un **canal seguro cifrado por donde intercambiar la información** con total garantía.
- Los datos enviados y recibidos mediante el uso de SSH se realiza por medio de **algoritmos de encriptación de 128 bits**, lo que hace que sea muy complicado de descifrar y de leer.
- El cliente tiene la posibilidad de utilizar de forma segura aplicaciones desde el intérprete de comandos del servidor, que permite administrar la máquina como si estuviera delante de ella.
- El uso de **SSH permite convertirse en un conducto para hacer seguros aquellos protocolos que no los son** mediante el uso de una técnica denominada reenvío por puerto.

Cómo configurar el protocolo SSH para hacerlo más seguro

Aunque hemos estado hablando de que el uso del protocolo SSH es totalmente seguro, esto no quiere decir que esté ajeno a sufrir algún tipo de ataque que ponga en riesgo nuestra información. Por este motivo, los usuarios tienen la opción de modificar la configuración por defecto que trae este protocolo para hacerlo aún más seguro, como puede ser el cambio del puerto por defecto o el número máximo de reintentos para conectarse al servidor. Veamos cómo mejorar la seguridad de nuestro SSH.

Lo primero que debemos hacer es localizar el fichero de configuración y que lleva por nombre "**sshd_config**". Este fichero suele estar en la ruta "**/etc/ssh**". Si no estuviera en esa dirección, podemos hacer uso del comando "**find**" de Linux para localizar su ubicación.

Una vez localizado lo editamos con nuestro programa de edición de ficheros favorito, como puede ser **Vi**, **Nano** o cualquier otro que tengamos instalado. Una vez que lo abrimos, nos debemos encontrar un contenido parecido al que os dejamos a continuación (los valores pueden variar).

Port 22

Protocol 2

LoginGraceTime 30

PermitRootLogin no

MaxAuthTries 2

MaxStartups 3

AllowUsers Hostalia

Entre las cosas que podemos hacer están:

a) Cambiar el puerto por defecto

Por defecto, SSH utiliza el puerto 22, por lo que cuando un hacker lanza un ataque lo suele hacer sobre este puerto. Si le cambiamos el número del puerto, el servicio no responderá al puerto por defecto y habremos puesto una nueva traba a quien intente conseguir nuestra información.

Para hacer este cambio, lo único que hay que hacer es en el fichero de configuración cambiar el valor del campo "**port**" por el valor que queramos, por ejemplo el puerto 10589.

Port 10589

b) Deshabilitar el acceso root

Todo servidor tiene asignado un usuario root que tiene privilegios para hacer cualquier tipo de acción sobre la máquina. Una buena práctica para mejorar la seguridad es impedir el acceso al servidor por medio de este usuario root y obligar al acceso por medio de alguno de los usuarios que hayamos creado y que no tienen privilegio root. Una vez logueados con nuestro usuario, podremos convertirnos en usuario root por medio del comando "**sudo**".

Para impedir el acceso del usuario root, debemos poner a "no" la variable "**PermitRootLogin**".

PermitRootLogin no

c) Deshabilitar el uso del protocolo 1 de SSH

Como hemos comentado a lo largo de este White Paper, hay dos versiones del protocolo SSH. La versión 1 que apareció primero y a la que se descubrió algunas vulnerabilidades, y la versión 2 que corregía esos problemas.

De forma general, se puede configurar para que se pueda utilizar cualquiera de las dos versiones, pero es recomendable no utilizar la primera de las versiones. Para indicar que vamos a utilizar la versión 2 hay que modificar la variable "**Protocol**" del siguiente modo:

Protocol 2

d) Limitar el número de reintentos

Por medio de la variable "**MaxAuthTries**", podemos indicar el número de veces que nos podemos equivocar al introducir el nombre de usuario o la contraseña. Una vez superado el número que hayamos indicado, la conexión se perderá y habrá que empezar de nuevo el proceso de conexión. Con esto evitaremos ataques de persistencia de la conexión.

Si queremos poner un máximo de 3 intentos, habría que indicarlo de la siguiente manera:

MaxAuthTries 3

e) Limitar el número de pantallas de login

Otra de las acciones que podemos llevar a cabo para mejorar la seguridad es limitar el número de ventanas de logueo simultáneas que podemos tener activas desde una misma IP, para de esta forma evitar ataques divididos. Una vez logueado el usuario, no será posible tener abiertos un número superior de terminales SSH al indicado en esta variable.

Si queremos limitar a una única pantalla de logueo por IP, habría que hacerlo de la siguiente manera:

MaxStartups 1

f) Limitar el tiempo que estará disponible la pantalla de login

Por medio de la instrucción "**LoginGraceTime**" indicamos el tiempo en segundos que estará disponible la pantalla de login para introducir nuestras credenciales. Pasado ese tiempo, la pantalla desaparecerá y habrá que volver a iniciar el proceso. Con esto evitamos que se pueda utilizar un script para que intente acceder al sistema.

Si queremos poner una duración de 20 segundos, lo haríamos de la siguiente manera:

LoginGraceTiem 20

g) Indicar los usuarios que pueden acceder vía SSH

Por medio de la directiva "**AllowUser**" podemos indicar los usuarios que podrán acceder al servidor vía SSH, así como desde que dirección IP lo podrán hacer. Veamos a continuación algunos ejemplos de cómo indicarlo.

1. **Indicar sólo el nombre de los usuarios que tendrán acceso**

   AllowUser pedro isabel

   En este caso estamos indicando que sólo los usuarios "pedro" e "isabel" tendrán acceso al sistema vía SSH, independientemente del equipo y la dirección IP desde donde se conecten.

2. **Acceso de un usuario desde una determinada dirección IP**

   AllowUser pedro@217.158.45.23

   De esta forma lo que indicamos es que el usuario "pedro" podrá acceder vía SSH a la máquina pero sólo desde la dirección IP que indiquemos, en el ejemplo la 217.158.45.23

3. **Acceso de un usuario desde una determinada red indicada**

   AllowUser pedro@217.158.45.*

   Con esta forma, que es muy similar al caso anterior, lo que hacemos es indicar al usuario que podrá acceder desde cualquier dirección IP que forme parte de la red indicada.

4. **Acceso de un usuario desde un determinado equipo de una red**

   AllowUser pedro@pedropc.miempresaweb.es

   En este caso estamos indicando que el usuario "pedro" sólo podrá acceder al servidor desde el equipo "pedropc.miempresaweb.es".

5. **Acceso de un usuario desde cualquier equipo perteneciente a la red**

   AllowUser pedro@*.miempresaweb.es

   En este caso, el usuario "pedro" podrá acceder vía SSH desde cualquier equipo que pertenezca a la red.

Al igual que hemos hecho en el punto 1, en una misma línea se puede indicar más de un usuario que tendrá acceso a la máquina vía SSH.