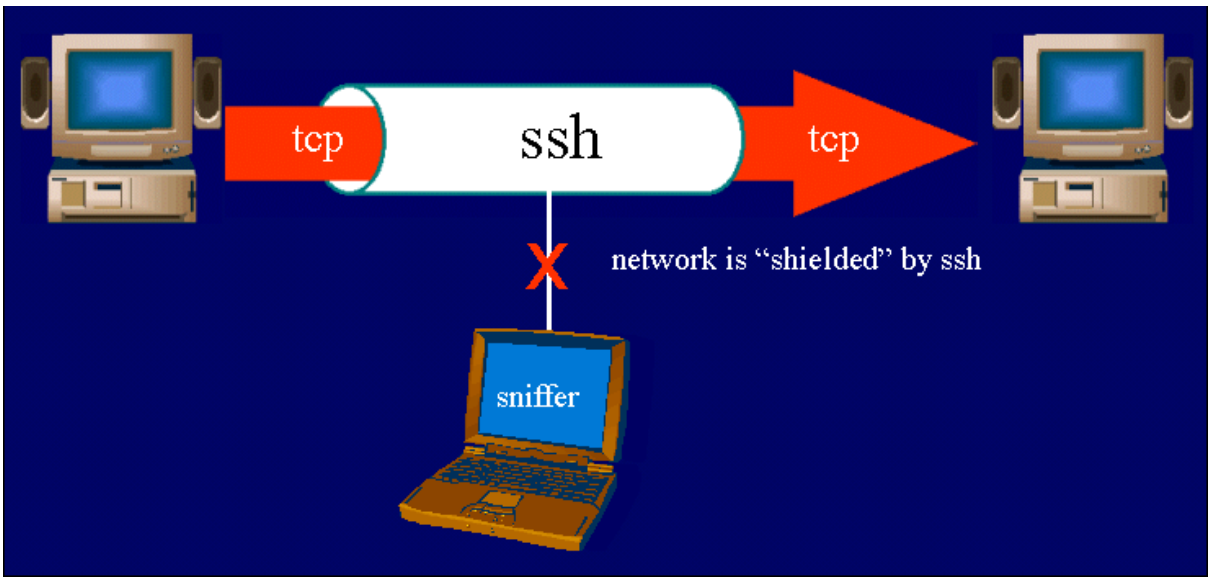
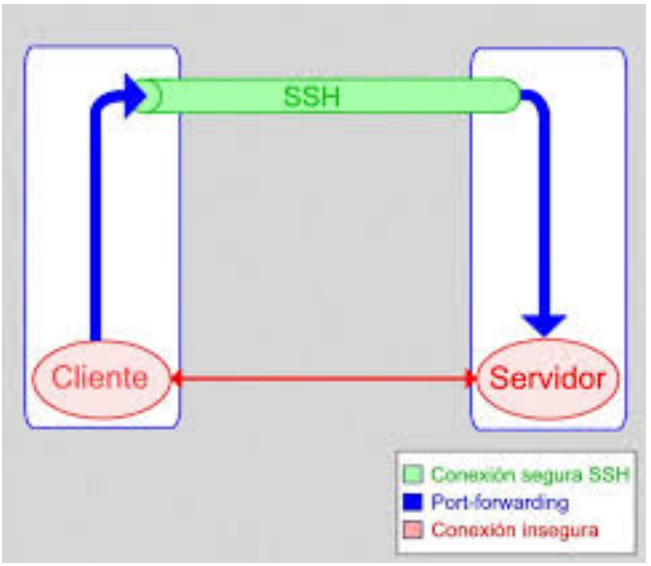
Servidor SSH Linux



Introducción

El servidor de shell seguro o SSH (Secure SHell) es un servicio muy similar al servicio telnet ya que permite que un usuario acceda de forma remota a un sistema Linux pero con la particularidad de que, al contrario que telnet, las comunicaciones entre el cliente y servidor viajan cifradas desde el primer momento de forma que si un usuario malintencionado intercepta l1os paquetes de datos entre el cliente y el servidor, será muy difícil que pueda extraer la información ya que se utilizan sofisticados algoritmos de cifrado.

La popularidad de ssh ha llegado a tal punto que el servicio telnet prácticamente no se utiliza. Se recomienda no utilizar nunca telnet y utilizar ssh en su lugar.



Instalar el servicio SSH en nuestro servidor

Vamos a instalar un servicio ssh en el lado del servidor. Este si pondrá a la escucha en el puerto 22 por defecto (si es que no decido usar otro)

Para instalar el servidor y el cliente ssh debemos instalar mediante apt-get el paquete ssh que contiene tanto la aplicación servidora como la aplicación cliente:

// Instalación de servidor ssh y cliente ssh
sudo apt install ssh

Archivos para configurar el servidor

Los archivos de configuración son:

- /etc/ssh/ssh_config: Archivo de configuración del cliente ssh
- /etc/ssh/sshd_config: Archivo de configuración del servidor ssh

Arrancando y parando el servidor

Arranque y parada manual del servidor ssh

El servidor ssh, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

// Iniciar o Reiniciar el servidor ssh
sudo /etc/init.d/ssh restart
Otra opción es: ***sudo service ssh restart***

// Parar el servidor ssh
sudo /etc/init.d/ssh stop
Otra opción es: ***sudo service ssh stop***

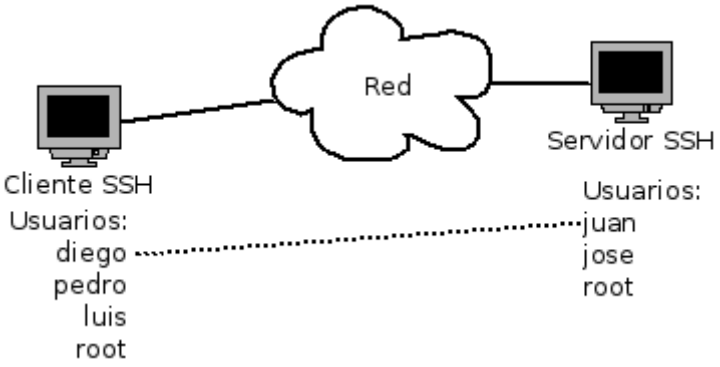
Conectándonos al servidor

Conectándonos al servidor desde un cliente Linux o Mac

ssh angel@192.168.1.136 -p <puerto>

donde:

- **angel** será el nombre de usuario con el cual queremos loguearnos en la máquina servidor
- **192.168.1.136** será la ip de la máquina servidor. Tambien la puedo sustituir por el nombre de dominio (si lo tengo). Ej: superangel.com



La primera vez que se conecte alguien desde dicho PC cliente, se instalará el certificado de autentificación del servidor, lo cual es normal si se trata de la primera vez. A la pregunta 'Are you sure you want to continue connecting (yes/no)?' debemos responder 'yes' ya que de lo contrario la comunicación finalizará.
Si ya nos hemos conectado anteriormente otras veces y vuelve a realizar esta pregunta, significa que alguien se está haciendo pasar por el servidor (nuestro servidor ha sido hackeado) o que se ha reconfigurado el servidor (cambio de nombre, IP, etc...)

Conectándonos al servidor desde un cliente Windows

Desde PCs clientes con Windows es posible conectarse por ssh a servidores Linux mediante el programa **Putty**.
http://www.putty.org/

Servicios adicionales que incorpora SSH

El paquete ssh no solamente nos proporciona conexión remota sino que proporciona otros servicios como ejecución remota de aplicaciones gráficas, servidor ftp seguro o copia remota de archivos.

Transferencia de archivos cliente-servidor y servidor cliente

También se dispone del comando **scp** que permite copiar archivos desde/hacia el servidor remoto desde un ordenador cliente.

Ejemplo, si el usuario jessica desea copiar el archivo /etc/hosts del servidor cuya IP es 192.168.1.239 a la carpeta actual de nuestro PC, ejecutá el siguiente comando:

Copiar un archivo que está en el servidor a nuestro ordenador cliente

El siguiente comando cogerá el archivo almacenado en la ruta /home/angel/fichero.txt del servidor y lo guardará en nuestra carpeta actual de nuestro cliente.
No olvides el punto del final
scp angel@192.168.1.136:/home/angel/fichero.txt .

Enviar un archivo desde nuestro ordenador al cliente al servidor

El siguiente código guardará un fichero del ordenador cliente llamado mifichero.txt dentro de servidor en concreto en la ruta /home/angel/prueba
scp mifichero.txt angel@192.168.1.136:/home/angel/prueba

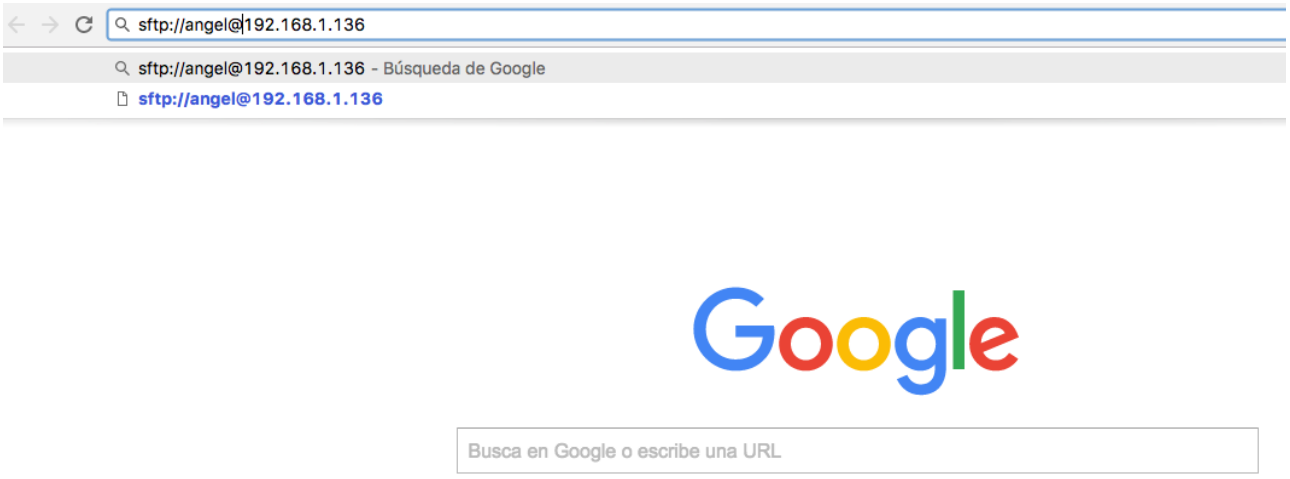
Copiar una carpeta entera

También se permite guardar de forma recursiva carpetas completas (con lo que tengan dentro de su interior)
scp -r /datos/ angel@192.168.1.136:/home/angel/pruebas/datos/

Servidor FTP seguro

El paquete ssh también incorpora un servidor ftp seguro y un cliente ftp seguro. Para activar el servidor ftp seguro tan solo hay que tener arrancado el servidor ssh.

El cliente ftp seguro es el comando sftp que funciona igual que el comando ftp. También podemos utilizarlo desde el navegador Nautilus escribiendo sftp://nombre-del-usuario@nombre-del-servidor por ejemplo en la url: sftp://angel@192.168.1.136



Ejecución remota de aplicaciones gráficas

Mediante ssh existe la posibilidad de ejecutar aplicaciones gráficas en el servidor y manejarlas y visualizarlas en el cliente. El servidor ssh deberá tener activada la redirección del protocolo X, es decir, deberá tener el siguiente parámetro en el archivo de configuración /etc/ssh/ssh_config:

// Habilitar la redirección X en ***/etc/ssh/sshd_config***
ForwardX11 yes

Ejemplo: supongamos que en nuestro terminal tenemos Damn Small Linux (que no dispone del gimp, programa "parecido" a photoshop) y deseamos conectarnos a otro PC que sí que tiene instalado el editor gráfico gimp, los pasos que haremos serán:

// Ejecutar aplicaciones gráficas
bea@cliente:~$ ***ssh -X angel@192.168.1.136*** // -X para redirigir Xwindows
angel@angel-desktop:~$ ***gimp*** // Ejecutamos el gimp

Identificación por certificado de clave pública

Ventajas:

- Las claves que se usan son brutalmente largas y muy muy seguras.

- No es necesario introducir las contraseñas, esto es crucial si tengo un sistema que hace copias de seguridad en el servidor todos los días a las 3:00 de la mañana.

Para evitar tener que introducir continuamente la contraseña cuando deseamos conectar con un servidor remoto por ssh, existe la posibilidad de autenticarse por certificado, para ello debemos:

1. Crear un certificado de usuario en el PC cliente
2. Copiar el certificado en el PC servidor

Para que el servidor ssh acepte la autentificación por medio de certificado, deberá tener activada la opción PubkeyAuthentication yes, es decir, deberá tener el siguiente parámetro en el archivo de configuración ***/etc/ssh/sshd_config***:
// Permitir autentificación por certificado
PubkeyAuthentication yes

Crear un certificado en el PC cliente

Para crear un certificado que permita autentificar al usuario, debemos ejecutar el comando ssh-keygen. Dicho comando creará dentro de nuestra carpeta home, en una carpeta llamada '.ssh', dos archivos:

- uno llamado id_rsa que será la clave privada de nuestro certificado
- y otro llamado id_rsa.pub que será la clave pública de nuestro certificado. Este último archivo será el que hay que copiar en el servidor remoto.

// Creación de un certificado
ssh-keygen -t rsa

Generating public/private rsa key pair.
Enter file in which to save the key (/home/angel/.ssh/id_rsa):
// Archivo del certificado. Podemos dejar el que viene por defecto
Created directory '/home/angel/.ssh'.

Enter passphrase (empty for no passphrase): // Opcional
Enter same passphrase again:
Your identification has been saved in /home/angel/.ssh/id_rsa.
Your public key has been saved in /home/angel/.ssh/id_rsa.pub.
The key fingerprint is:
c8:a4:fe:0c:19:78:8e:7d:05:5b:13:df:37:17:e8:ea angel@cebem.com

El modificador -t rsa indica el tipo de sistema critográfico que vamos a usar para generar las claves, en concreto RSA (Rivest, Shamir y Adleman) es un sistema criptográfico de clave pública desarrollado en 1977.

Veamos qué datos ha generado:
cat /home/angel/.ssh/id_rsa.pub
cat /home/angel/.ssh/id_rsa

Copiar el certificado en el ordenador servidor

Para poder identificarse en el servidor como angel desde el cliente, debemos copiar el archivo **id_rsa.pub (solo este, nunca el otro)** que hemos creado en el cliente, en la carpeta home de angel en el servidor dentro de una carpeta llamada '.ssh' en un archivo llamado authorized_keys. Para copiar dicho archivo del cliente al servidor, podemos hacerlo con scp. Supongamos que el cliente se llama 'cliente' y el servidor se llama 'servidor':

// Copia del certificado y prueba de la conexión
// Nota: el símbolo ~ en Linux es la carpeta home del usuario es equivalente a $HOME
scp ~/.ssh/id_rsa.pub angel@192.168.1.136:/home/angel/.ssh/authorized_keys
id_rsa.pub 100% 242 0.2KB/s 00:00 // Copiado

También copiamos el fichero con nuestra clave publica en el servidor
scp ~/.ssh/id_rsa.pub angel@192.168.1.136:/home/angel/.ssh/

Copiando varios certificados en el ordenador servidor

Opcionalmente si vamos a usar múltiples certificados podemos usar el siguiente código.

cat *~/.ssh/id_rsa.pub* | *ssh angel@192.168.1.136 "mkdir -p ~/.ssh && cat >> /home/angel/.ssh/authorized_keys"*

Conectándonos al servidor usando ahora una clave pública

ssh angel@192.168.1.136 // Probamos la conexión
angel@192.168.1.136:~$ // Ya estamos en el servidor sin necesidad de contraseña

NOTA: Si cuando generamos las claves con el comando ***ssh-keygen*** pusimos contraseña *passphrase* nos pedirá esta contraseña.

Deshabilitando el acceso mediante cuenta root

Abrimos el archivo de configuración
sudo pico /etc/ssh/sshd_config

Editamos la linea:
PermitRootLogin no

Reiniciamos el servidor ssh
sudo /etc/init.d/ssh restart
O tambien: ***sudo service ssh restart***

Entendiendo la criptografía de clave pública

El propósito de usar la autenticación mediante claves públicas es sacar el mayor partido posible a SSH mediante el uso de clave privada y pública. De forma que al compartir esta última con el servidor, podamos identificarnos automáticamente, sin necesidad de utilizar el esquema clásico de usuario y contraseña, al que estamos acostumbrados

Autenticación de clave pública

El cifrado de clave pública utiliza un algoritmo matemático con un par de claves pública/privada para cifrar y descifrar datos. Una de estas claves es una clave pública, que puede distribuirse libremente entre los participantes de la comunicación, y la otra es una clave privada, que el propietario de la clave debe guardar en un lugar seguro. Los datos cifrados con la clave privada sólo pueden descifrarse con la clave pública, mientras que los datos cifrados con la clave pública sólo pueden descifrarse con la clave privada.

Cuando se utilizan claves para la autenticación, la parte que se está autenticando crea una firma digital utilizando la clave privada de un par de claves pública/privada. El receptor deberá utilizar la clave pública correspondiente para comprobar la autenticidad de la firma digital. Es decir, el receptor deberá tener una copia de la clave pública de la otra parte y confiar en la autenticidad de dicha clave.

¿Qué algoritmos de cifrado usar?

SSH permite usar los algoritmos RSA y DSA, pero… ¿cuál de ellos nos conviene más?

- **RSA** – Es un algoritmo asimétrico cifrador de bloques, que utiliza una clave pública, la cual se distribuye y otra privada, guardada en secreto por su propietario. Su funcionamiento reside en el uso de expresiones exponenciales dentro de la aritmética modular. Obteniendo una completa seguridad, debido a que aún no se conocen formas óptimas de factorizar un número grande en sus factores primos utilizando ordenadores personales. El RSA se basa en dos problemas matemáticos: el problema de factorizar números grandes y el problema RSA. El descifrado completo de un texto cifrado con RSA es computacionalmente intratable..
- **DSA** – (Digital Signature Algorithm o Algoritmo Estándar de Firmado) es el algoritmo de firmado digital incluido en el DSS (Digital Signature Standard o Estándar de Firmas Digitales) del NIST Norteamericano. Está basado en el problema de los logaritmos discretos y únicamente puede emplearse para las firmas digitales. A diferencia del RSA, que puede emplearse también para encriptar.La elección de este algoritmo como estándar de firmado generó multitud de críticas puesto que perdía bastante flexibilidad respecto al RSA.

El algoritmo DSA es más rápido para generar la firma que para verificarla, al contrario de lo que sucede con RSA. Por lo que para realizar la autenticación en nuestro servidor SSH usaremos este último.

Además, si comparamos el tamaño de las llaves generadas por ambos algoritmos, comprobaremos cómo las utilizadas por RSA son superiores a las de DSA.

Acceso mediante Shell gráfica por ssh

https://www.linuxito.com/gnu-linux/nivel-medio/550-lanzar-aplicaciones-graficas-desde-una-sesion-ssh

Bibliografía

- https://es.wikipedia.org/wiki/Secure_Shell
- http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m5/instalacin_de_servidor_de_ssh.html
- https://linuxcode.wordpress.com/2009/08/08/autentificacion-mediante-claves-publicas-en-ssh/