Acceso Remoto.

1. SSH

1.1. 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 encriptadas desde el primer momento de forma que si un usuario malintencionado intercepta los paquetes de datos entre el cliente y el servidor, será muy difícil que pueda extraer la información ya que se utilizan sofisticados algoritmos de encriptación.

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.

Para que un usuario se conecte a un sistema mediante ssh, deberá disponer de un cliente ssh. Desde la primera conexión, y mediante encriptación asimétrica, las comunicaciones se encriptan incluido el proceso de autentificación del usuario cuando proporciona su nombre y su contraseña. También se proporciona una clave de encriptación simétrica para encriptar las comunicaciones del resto de la sesión mediante encriptación simétrica por su menor necesidad de procesamiento.

1.2. Instalación del servidor y el cliente ssh

Para instalar el servidor y el cliente ssh debemos instalar (http://www.ubuntu-guia.com/2009/06/como-instalar-paquetes-y-programas-en.html) mediante aptitude install el paquete ssh que contiene tanto la aplicación servidora como la aplicación cliente:

// Instalación de servidor ssh y cliente ssh
root@server:# ***aptitude install install ssh***

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

1.3. 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
root@server:# ***/etc/init.d/ssh restart***
// Parar el servidor ssh
root@server:# ***/etc/init.d/ssh stop***

1.4. Conexión al servidor mediante ssh

Para conectar desde un PC cliente al servidor mediante ssh, debemos ejecutar el comando ssh seguido del nombre ó dirección IP del servidor. La conexión se realizará con el mismo nombre de usuario que estemos utilizando en el PC cliente.

Ejemplo, supongamos que juan, desde el PC llamado pc05, quiere conectarse al servidor cuya IP es 192.168.1.239:

// Conexión por ssh
juan@pc05:~$ ***ssh 192.168.1.239***
The authenticity of host '192.168.1.239 (192.168.1.239)' can't be established.
RSA key fingerprint is 51:70:3f:9c:ac:49:52:74:88:f5:45:a6:ae:f0:9c:8a.
Are you sure you want to continue connecting (yes/no)? ***yes***
Warning: Permanently added '192.168.1.239' (RSA) to the list of known hosts.
Password: ***// Introducir contraseña de juan***
juan@server:~$ ***// Ya estámos en el servidor***

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 se cortará. Si ya nos hemos conectado anteriormente otras veces y vuelve a realizar ésta 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...)

Si deseamos conectarnos al servidor utilizando un nombre de usuario diferente, debemos incluir el nombre de usuario antes del nombre o IP del servidor y separado por una arroba '@'. Ejemplo, supongamos que juan, desde el PC llamado pc05, quiere conectarse como miguel al servidor cuya IP es 192.168.1.239:

// Conexión por ssh como otro usuario
juan@cliente:~$ ***ssh miguel@192.168.1.239***
Password: ***// Introducir contraseña de miguel en el servidor***
miguel@servidor:~$ ***// Ya estámos en el servidor como miguel***

Desde PCs con Windows es posible conectarse por ssh a servidores Linux mediante el programa **Putty**. Se trata de un cliente ssh para Windows que permite acceder en modo texto al sistema Linux desde sistemas Windows.

1.5. Servicios adicionales

El paquete ssh no solamente nos proporciona conexión remota sino que proporciona otros servicios como:

- **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
X11Forwarding yes

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

// Ejecutar aplicaciones gráficas
juan@cliente:~$ ***ssh -X miguel@192.168.1.239 // -X para redirigir Xwindows.***
miguel@server:~$ ***gimp // Ejecutamos el gimp***

Desde PCs con Windows es posible conectarse por ssh a servidores Linux de forma gráfica mediante **Cygwin**. Se trata de un conjunto de programas libres que simulan un 'Unix para Windows' con servidor gráfico X y cliente ssh para Windows entre otras cosas, que permite acceder en modo gráfico al sistema Linux desde sistemas Windows. Otros servidores X gratuitos para Windows son **Xming** y Mocha.

- **Copia remota de archivos**

También se dispone de el comando scp que permite copiar archivos desde y hacia el servidor remoto desde el cliente. Ejemplo, si deseamos copiar el archivo /etc/hosts del servidor cuya IP es 192.168.1.239 e identificándonos como juan en la carpeta actual de nuestro PC, ejecutaremos el siguiente comando:

// Copiar un archivo del servidor a nuestro PC
root@cliente:~# ***scp juan@192.168.1.239:/etc/hosts .***
Password: ***// Introducimos la contraseña de juan en el servidor***
hosts 100% 443 0.4KB/s 00:00 ***// Archivo copiado***
root@cliente:~#
// Copiar un archivo de nuestro PC al servidor
// La carpeta de destino debe existir en el servidor
root@cliente:~# ***scp miarchivo.txt juan@192.168.1.239:/home/juan/pruebas/***

Password: ***// Introducimos la contraseña de juan en el servidor***
miarchivo.txt 100% 443 1.6KB/s 00:00 ***// Archivo copiado***
root@cliente:~#
// Copiar una carpeta y subcarpetas de nuestro PC al servidor
root@cliente:~# ***scp -r /datos/*.* juan@192.168.1.239:/pruebas/datos/***
Password: ***// Introducimos la contraseña de juan en el servidor***
datos/.* 100% 443 50.6KB/s 00:03* ***// Archivos copiados***
root@cliente:~#

Desde PCs con Windows es posible utilizar el programa **WinSCP** que permite copiar archivos desde y hacia el servidor. Se trata de un cliente que utiliza el protocolo ssh para acceder al sistema de archivos del servidor Linux desde sistemas Windows.

1.6. Fortificación del servicio ssh

Este servicio dispone de una configuración segura por defecto, aunque también es posible mejorarla y ajustarla a los requerimientos y políticas de cualquier organización, minimizando de esta forma el número potencial de riesgos.

Todas las opciones que se muestran se han de modificar en el fichero de configuración, generalmente /etc/ssh/sshd_conf

Port (por defecto: 22)

Especifica el número del puerto en el que el servicio escuchará. El mayor número de ataques se producen por herramientas automáticas, que en busca de este demonio, probarán usuarios y contraseñas comunes. Modificando el puerto a un número no estándar, por ejemplo 8822 se evitarán en su gran mayoría, ahorrando de esta forma la generación de cientos de registros inútiles al día. Tiene especial importancia en servidores expuestos en Internet.

Un truquito para elegir el puerto es utilizar el que ya conocemos, en este caso el 22 y añadirle algún número para que sea más fácilmente memorizable.

// Conexión por ssh como otro usuario por el puerto 8822
juan@pc05:~$ ***ssh prado@10.2.23.105 –p 8822***

Protocol (por defecto: 1,2)

Especifica la versión del protocolo con la que un cliente puede conectarse.
Debido a las vulnerabilidades conocidas de ataques de hombre en el medio en la versión 1, este parámetro se ha de establecer únicamente a "2".

PermitRootLogin (por defecto: yes)

El usuario administrador del sistema no debería realizar la autenticación directamente contra el equipo. Es recomendable modificar este parámetro dejándolo en "no" y en caso de

necesidad por el uso de scripts o aplicaciones automáticas, utilizar el valor "forced-commands-only", que permitirá la conexión del usuario root solo si es mediante certificado.

ListenAddress (por defecto: all)

La directiva ListenAddress especifica en que direcciones IP se abrirá el puerto para ofrecer el servicio. Si el sistema dispone de más de una dirección IP, ya sea IPv4 o IPv6, es conveniente limitar esta escucha únicamente a las necesarias. Se permite el uso repetido de este parámetro, en caso de por ejemplo, querer escuchar en dos de las tres direcciones IP de un equipo.

AllowUsers (por defecto: todos los usuarios)

AllowUsers concreta que usuarios tienen permitido el uso del servicio SSH, se permite especificar distintos usuarios separados por coma y opcionalmente, el host desde el que conectan con el formato "usuario@host".

Esta directiva complementa el grupo "Allow/Deny" compuesto por otros parámetros, en concreto, según su orden de procesado: AllowUsers, DenyUsers, AllowGroup y DenyGroup.

MaxAuthTries (por defecto: 6)

Con el parámetro MaxAuthTries se especifica el máximo número de intentos de autenticación fallida por conexión. Cuando se alcanza la mitad de este número los sucesivos intentos serán registrados en los correspondientes logs. Se recomienda su configuración a un valor más bajo como 3.

Más información en:
http://www.securitybydefault.com/2009/08/12-fortificacion-del-servicio-ssh.html

1.7. Identificación por certificado

Para evitar tener que introducir continuamente la contraseña cuando deseamos conectar con un servidor remoto por ssh, existe la posibilidad de autentificarse 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. Éste último archivo será el que hay que copiar en el servidor remoto.

// Creación de un certificado
miguel@cliente:~$ ***ssh-keygen -t rsa***
Generating public/private rsa key pair.
Enter file in which to save the key (/home/miguel/.ssh/id_rsa):
// Archivo del certificado. Podemos dejar el que viene por defecto
Created directory '/home/miguel/.ssh'.
Enter passphrase (empty for no passphrase): ***// Opcional***
Enter same passphrase again:
Your identification has been saved in /home/miguel/.ssh/id_rsa.
Your public key has been saved in /home/miguel/.ssh/id_rsa.pub.
The key fingerprint is:
c8:a4:fe:0c:19:78:8e:7d:05:5b:13:df:37:17:e8:ea miguel@cliente

- **Copiar el certificado en el PC servidor**

Para poder identificarse en el servidor como miguel desde el cliente, debemos copiar el archivo id_rsa.pub que hemos creado en el cliente, en la carpeta home de miguel en el servidor dentro de una carperta 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*
// Si el usuario del servidor no ha usado nunca ssh, la carpeta .ssh no existirá y habrá que crearla de forma manual en un paso previo.(ssh miguel@servidor mkdir ruta/carpeta)
miguel@cliente:~$ ***scp ~/.ssh/id_rsa.pub miguel@servidor:~/.ssh/authorized_keys***
Password: ***// Va a ser la última vez que introduzcamos la contraseña***
id_rsa.pub 100% 242 0.2KB/s 00:00 ***// Copiado***
miguel@cliente:~$ ***ssh miguel@servidor // Probamos la conexión***
miguel@servidor:~$ ***// Ya estamos en el servidor sin necesidad de contraseña***
// si queremos aceptar más clientes habría que hacerlo de la siguiente forma:
miguel@cliente2:~$ ***scp /home/miguel/.ssh/id_rsa.pub miguel@servidor:~/***
miguel@cliente2:~$ ***ssh miguel@servidor***
*miguel@servidor****:~$ cat ~/id_rsa.pub >> ~/.ssh/authorized_keys***

*//Borra el fichero id_rsa***.**pub del servidor: rm id_rsa.pub

//Una vez hecho esto, no olvides darle a esta carpeta y a este archivo los permisos apropiados, de otra manera otro usuario en el mismo sistema que tu podría agregar una llave a tu archivo de llaves permitidas y podría entonces conectarse como tú por medio de SSH.

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

2. Otros accesos remotos

2.1. VNC

VNC es un servicio que crea servidores gráficos sobre pantallas o displays virtuales y permite establecer conexiones remotas desde otros PCs de la red al servidor, de forma gráfica de manera similar a si fuera un servidor de terminales.

La diferencia más significativa con respecto a un servidor de terminales Xwindow como el que hemos visto en el punto anterior es que mientras cuando hacemos una conexión Xwindow el cliente debe disponer de un servidor gráfico, cuando hacemos la conexión con VNCServer, la imagen gráfica se genera en el servidor y básicamente lo que fluye por la red son pantallazos jpg, de esa forma el cliente puede ser más ligero pero la carga del servidor es mucho mayor.

Para que pueda funcionar es necesario instalar y ejecutar el servidor VNC. Este servidor atenderá las peticiones de los clientes. El terminal deberá disponer del cliente de VNC llamado vncviewer del que hay versiones para todos los sistemas operativos incluidos MS-DOS, Linux y Microsoft Windows.

Cuando ejecutamos el servidor de VNC, se crea un nuevo escritorio (nuevo display X) al cual se puede acceder de forma remota con el cliente de VNC. Se pueden ejecutar tantos servidores VNC como permita la memoria del sistema, pudiendo varios usuarios acceder de forma simultánea, cada uno a su escritorio independiente.

En la estación de trabajo donde se ejecute el visor de VNC, éste aparece como una ventana en el entorno de escritorio local, presentando la interfaz de usuario; todas las funciones del S.O., así como las aplicaciones, se ejecutan en el servidor.

Aunque hay muchos clientes como el xvncviewer o el xtightvncviewe y servidores como tightvncserver, las últimas versiones de Ubuntu, ya vienen con el Cliente VNC (Aplicaciones, Internet, **Visor de Escritorio Remoto**) y el servidor (Sistema, Preferencias, **Escritorio Remoto**).

http://www.ubuntu-guia.com/2010/03/escritorio-remoto-en-ubuntu.html#more

Como se usa el mismo protocolo, clientes y servidores de distintos sistemas operativos son compatibles.

2.2. Free Nx

NX es una tecnología para manejar conexiones remotas a X Window de forma suficientemente rápida incluso sobre un módem de 56K. Para ello utiliza compresión de datos y mecanismos de caché, que le proporcionan un rendimiento netamente superior al de otras soluciones de este tipo como VNC. También emplea SSH para cifrar la conexión entre servidor y cliente. Además de permitir a los usuarios loguearse en una máquina remota accediendo al escritorio, permite también suspender y recuperar sesiones. NX es un producto de la empresa NoMachine, que dispone de licencia GPL sobre la propia tecnología NX, existiendo múltiples implementaciones, tanto comerciales como gratuitas, y tanto libres como propietarias, de servidores y clientes.

Más información sobre Free NX

http://es.kioskea.net/faq/sujet-2595-instalar-nx-server

http://miblockdenotix.wordpress.com/2008/09/04/nomachine-nx-freenx-acceso-remoto-en-ubuntu/

2.3. Rdesktop

rdesktop (Remote Desktop Protocol Client) es un cliente del protocolo RDP de Microsoft, escrito en C (trabaja directamente sobre las X's, X11) y publicado bajo licencia GPL (GNU General Public License) que funciona sobre Linux, BSD y otros sistemas derivados de UNIX.

La implementación original es la aplicación Windows Terminal Services, y tras sucesivas actualizaciones ha pasado a llamarse Windows Remote Desktop (Escritorio Remoto de Windows).

Normalmente viene preinstalado (cliente y servidor) en la familia de productos Windows 2000 Server, XP, 2003 Server, 2008 Server, Vista, 7 y 8

¿Cómo lo conseguimos?

En Debian, Ubuntu y otras derivadas: ***rdesktop***

sudo apt-get install rdesktop

¿Cómo funciona?

rdesktop dirección_del_equipo (parámetros opcionales)

El puerto por defecto para rdp es **3389/tcp**,