# DESPLIEGUE DE APLICACIONES WEB

Servicio de acceso y control remoto

# CONTENIDOS

1. ¿Qué es el servicio de acceso y control remoto?
2. El servicio SSH
3. ¿Cómo funciona SSH?
4. ¿Qué es un cliente SSH?
5. ¿Qué es un servidor SSH?
6. Acceso remoto con FreeNX
7. Servidor SSH bajo Windows 2008 Server
8. Servicios de Terminal Server



The McGraw-Hill Companies

# 1. ¿Qué es el servicio de acceso y control remoto?

Los servicios de acceso y control remotos permiten, mediante la utilización de determinadas aplicaciones de software, establecer conexiones con equipos a distancia y administrarlos de manera centralizada sin necesidad de acceder a ellos.

En el caso de que se disponga de equipos que no tienen teclado o pantalla, o bien de servidores apilados en un rack o que no estén físicamente presentes, es muy importante contar con mecanismos que permitan administrarlos remotamente de forma cómoda, rápida y segura.

Lógicamente esta función, que parece útil e inofensiva, puede tener consecuencias impredecibles si no se lleva a cabo con unas condiciones de seguridad bien definidas. Cualquier agujero de seguridad que presenten dichas herramientas puede permitir el acceso de terceros no deseados a informaciones confidenciales.

Las herramientas de administración remota más utilizadas actualmente son:

1. En modo texto: telnet, rlogin y Secure Shell (SSH).

2. En modo gráfico: VNC en entornos Unix GNU/Linux, NX y los servicios de Terminal Server en Windows.



The McGraw-Hill Companies

# 2. El servicio SSH

Esta herramienta permite establecer conexiones seguras entre máquinas remotas. Su funcionamiento se describe en el RFC 4251.

Las principales características del servicio SSH son las siguientes:

- Utiliza el puerto 22 (TCP y UDP), el protocolo SSH y sigue el modelo cliente-servidor.
- Permite la autenticación de los usuarios mediante contraseña o un sistema de claves.
- Permite su integración con otros sistemas de autenticación como Kerberos, PGP o PAM.
- Está implementado para la mayoría de sistemas operativos y plataformas.

## 2.1. Ventajas de utilizar SSH

- Después de la primera conexión, el cliente puede saber que se conectará al mismo servidor en futuras sesiones. (Por *cliente,* se entiende la máquina, el equipo o el ordenador desde donde se lanza la orden SSH correspondiente.)
- El cliente transmite al servidor la información necesaria para su autenticación (usuario y contraseña) en formato cifrado.
- Todos los datos que se envían y se reciben durante la conexión se transfieren cifrados.
- El cliente puede ejecutar aplicaciones gráficas desde el *shell* (intérprete de órdenes) de forma segura.



The McGraw-Hill Companies

# 2. El servicio SSH

## 2.1. Ventajas de utilizar SSH

Con la utilización de SSH se evita:

1. La interceptación de la comunicación entre dos sistemas por parte de una máquina tercera que copia la información que circula entre ellas y puede introducir modificaciones y reenviarla a la máquina de destino.

2. La suplantación de un host o enmascaramiento, es decir, que una máquina finja que es la máquina de destino de un mensaje, en cuyo caso el cliente no se da cuenta de que está siendo engañado y continúa la transmisión.

   Ambos problemas se evitan con el cifrado de paquetes mediante claves que solo son conocidas por el sistema local y el remoto.

| Versiones SSH | SSH1 | SSH2 |
|---|---|---|
| Contiene | Algoritmos de encriptación patentados, algunos caducados, y algún agujero de seguridad. | Versión más segura. |
| Claves que utilizan | Dos claves: una clave pública y otra que se genera de forma aleatoria al solicitar el inicio de sesión. | Dos claves, que dependen del algoritmo de encriptación utilizado. |
| Inconvenientes | 1. La generación aleatoria de clave consume mucho tiempo de CPU.<br>2. Tiene un fallo de seguridad que compromete la clave justo después de ser generada. | Ninguno, de momento. |
| Algoritmos de encriptación soportados | RSA. | RSA y DSA. |



The McGraw-Hill Companies

# 3. ¿Cómo funciona SSH?

**Las claves de una correcta conexión remota son las siguientes:**

- No transmitir las contraseñas en texto plano por la red.
- Proceso de autenticación con garantías.
- Ejecución segura de las órdenes remotas, como son las transferencias de archivos.
- Sesiones gráficas X11 seguras (entornos GNU/Linux).

**El servicio SSH garantiza todos estos puntos y funciona según el proceso siguiente:**

1. La máquina cliente abre una conexión TCP sobre el puerto 22 del servidor.
2. La máquina cliente y el servidor se ponen de acuerdo en la versión de SSH que van a utilizar. En este momento se determina el algoritmo de cifrado (simétrico) a utilizar para la transferencia de datos.
3. El servidor tiene dos claves (pública y privada). El servidor envía su clave pública al cliente.
4. El cliente recibe la clave pública y la compara con la que tiene almacenada para verificar si es auténtica. La primera vez (como no dispone de esta clave pública), SSH pide que el usuario la confirme. En este envío se podría producir un cambio y sustituirla por otra. Se trata de su punto más débil. De hecho, se trata de un tipo de ataque bastante común conocido como *man in the middle*. En las ocasiones siguientes, cuando el cliente reciba la clave pública del servidor, la comparará con la que ya tiene almacenada. Se pueden prevenir ataques *man in the middle* contra SSH en una intranet fácilmente. Basta con publicar un listado con las claves de los servidores de la intranet, para que los usuarios puedan verificarlas antes de aceptarlas.



***(Continúa…)***

# 3. ¿Cómo funciona SSH?

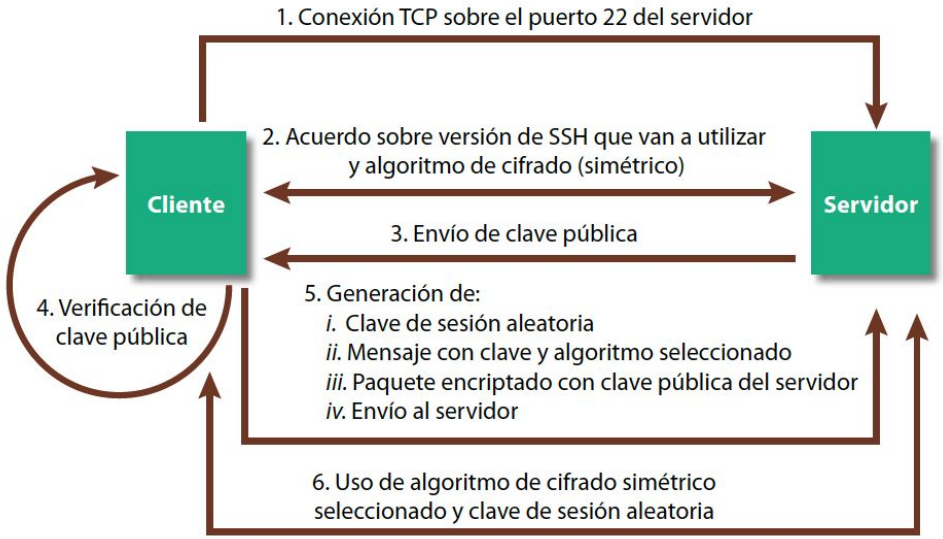
***(Continuación)***

5. El cliente genera una clave de sesión aleatoria y crea un mensaje que contiene la clave aleatoria generada y el algoritmo seleccionado, todo ello encriptado haciendo uso de la clave pública del servidor. El cliente envía este paquete cifrado al servidor.

6. Para el resto de la sesión remota se utiliza el algoritmo de cifrado simétrico seleccionado y clave de sesión aleatoria.

7. Llegados a este punto se autentica el usuario y aquí pueden usarse varios mecanismos, algunos de los cuales veremos más adelante.

8. Por último se inicia la sesión de usuario.






The McGraw-Hill Companies

# 3. ¿Cómo funciona SSH?

## 3.1. ¿Qué es un túnel SSH?

En su mayoría, los protocolos que se emplean en las comunicaciones están basados en diseños de hace casi 30 años, cuando la seguridad en redes no era un problema.

Como ya se comentó, Telnet, FTP, POP3 son protocolos muy comunes pero que descuidan la seguridad y confidencialidad de los datos que envían. ¿De qué sirve proteger los servidores, utilizar una buena política de contraseñas y actualizar las versiones de las aplicaciones servidoras, si luego cuando un usuario de POP3, por ejemplo, quiere ver su correo electrónico desde su centro de estudios, envía su usuario y contraseña en texto plano (sin encriptar) por la red?

Para evitarlo, hay dos posibles soluciones:

• Crear o utilizar protocolos seguros.

• Modificar los protocolos inseguros de forma que se comporten como protocolos seguros.

De estas soluciones, la segunda es mucho más viable, ya que aprovecha gran parte de los servidores y clientes existentes en el mercado. El objetivo será convertir los protocolos clásicos en protocolos seguros. A continuación, veremos qué tiene que ver el servicio SSH con ello.

El procedimiento consiste en crear un túnel por el que viajen los datos de manera segura.

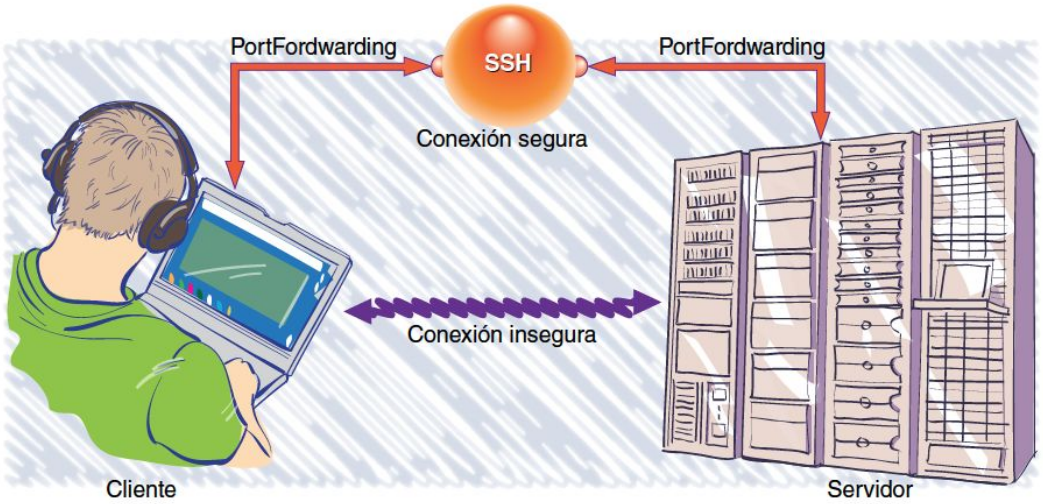
En los extremos de dicho túnel se están ejecutando servicios que utilizan protocolos no seguros, como POP3 o FTP. SSH es capaz de asegurar la comunicación mediante la criptografía y haciendo uso de la técnica del reenvío de puertos *(PortForwarding)*. SSH toma los datos que el cliente envía en un extremo del túnel y los reenvía por el canal seguro creado a partir de dicho túnel, hacia el otro extremo donde se recogen los datos que son reenviados al servidor.



The McGraw-Hill Companies

# 3. ¿Cómo funciona SSH?

## 3.1. ¿Qué es un túnel SSH?




El reenvío de puertos puede ser interesante para los siguientes propósitos:

- Para acceder a servicios TCP internos de una LAN con direcciones privadas.
- Para no enviar la clave en texto plano de FTP, Telnet, Messenger, POP3, IMAP o SMTP.
- Para atravesar un cortafuegos (Firewall) donde solo está permitido SSH.
- Para disponer de servicios X a través de una red insegura.

El epígrafe 5.3 muestra una aplicación práctica del uso de los túneles. En concreto el reenvío por TCP/IP.



The McGraw-Hill Companies

# 4. ¿Qué es un cliente SSH?

El cliente SSH es la herramienta de software que permite al usuario, desde una máquina remota, solicitar el establecimiento de una conexión segura con el servidor SSH.

La conexión SSH se puede llevar a cabo mediante herramientas gráficas o bien desde una consola, en modo de línea de comandos.

Por ejemplo, la herramienta Webmin ofrece la posibilidad de establecer conexiones SSH. Para utilizar el cliente SSH de Webmin si el navegador empleado es Firefox y las pruebas se hacen con la distribución Ubuntu GNU/Linux, hay que tener instalado el complemento para Java-console (JRE). En la pantalla inicial, si se selecciona la categoría *Otros* y, a continuación *Conexión SSH*, se abrirá una ventana de conexión.

Es posible que muestre un mensaje de seguridad relacionado con el certificado diciendo que lo ha emitido una fuente que no es de confianza. Bastará con contestar que se confía.

Antes de abrir la conexión se pueden modificar algunos parámetros de la conexión. Para ello, habrá que ir a *Configuración de módulo*.


El proceso o demonio que se ejecuta en el cliente es ssh y se encuentra en el directorio /usr/bin.

Dentro de las opciones gráficas, también pueden utilizarse la herramienta FreeNX o el cliente PuTTY.

Si el acceso al servidor se hace desde una terminal de texto, la sintaxis es la siguiente: `ssh [usuario@]host`, donde usuario es el login de conexión del usuario y host, la IP de la máquina servidor SSH o su nombre si se tiene configurado un servidor DNS.



The McGraw-Hill Companies

# 4. ¿Qué es un cliente SSH?

## 4.1. Transferencia segura de archivos

### La orden `scp`

Permite realizar transferencias simples desde la línea de comandos. Funciona como el comando cp que se utiliza para copiar en local, pero de forma remota y, a diferencia del `rcp`, de forma segura.

Con esta orden pueden hacerse copias seguras de archivos, con conexión segura y encriptada, entre distintas máquinas, así como directorios completos con el carácter asterisco (*).

Sigue los mismos esquemas de comprobación que SSH. Su funcionamiento dependerá de la manera en que cada usuario haya configurado SSH; es decir, si dispone de clave propia o no y si la clave tiene frase de paso o no.

| Sintaxis scp | Descripción |
|---|---|
| `$ scp nombre _ usuario@máquina _ origen:archivo _ origen`<br>`Nombre _ usuario@máquina _ destino:archivo _ destino` | Copia el `archivo _ origen` desde la `máquina _ origen` al `archivo _ destino` en la `máquina _ destino`. |
| `scp archivo _ local nombre _ usuario@ maquina _ remota:archivo _ copia` | Copia el `archivo _ local` a la máquina remota. |
| `scp nombre _ usuario@maquina _ remota:archivo _ remoto archivo _ copia _ local` | Copia `archivo _ remoto` a la máquina local. |
| `scp /directorio/* nombre _ usuario@ nombre _ maquina:/directorio _ destino/` | Copia directorio local copiado a la máquina remota. |

**Ejemplos**

```
alumno1@pc11:~$ scp archivo pc12:~
alumno1@pc11:~$ scp pc12:/path/archivo _ remoto path/archivo _ local
alumno1@pc11:~$ scp archivo alumno@pc12:~
```



The McGraw-Hill Companies

# 4. ¿Qué es un cliente SSH?

## 4.1. Transferencia segura de archivos

**La orden `sftp`**

Intenta emular la forma de uso de un cliente FTP ordinario para abrir una sesión segura e interactiva de este tipo.

La sintaxis es la siguiente:

```
alumno1@pc11:~$ sftp nombre_usuario@ nombre_maquina
```

Una vez establecida la conexión, puede utilizarse una serie de comandos específicos en FTP. No obstante, solo está disponible en OpenSSH 2.5 y versiones superiores.

```
alumno1@pc11$ sftp maquina_remota

sftp>help
```

Este último comando permite visualizar la lista de órdenes ftp disponibles. En esta conexión se asume que el usuario *alumno1* se conecta como *alumno1* en la máquina_remota y, por lo tanto, dicho usuario debe existir en ella.



The McGraw-Hill Companies

# 4. ¿Qué es un cliente SSH?

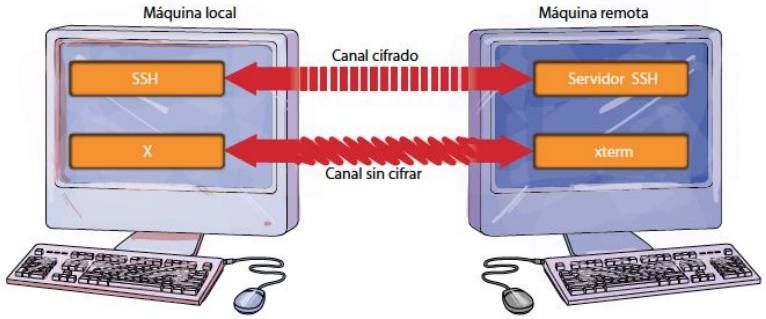
## 4.2. Reenvío X11

Una de las funciones de SSH es establecer una línea de órdenes segura, aunque también se pueden abrir sesiones X por un canal SSH.

Cuando se ejecuta un programa X Window desde una shell segura, el cliente y el servidor SSH crean un nuevo canal seguro dentro de la conexión SSH actual. Los datos del programa X Window se envían a la máquina cliente a través de dicho canal como si se realizase una conexión al servidor X a través de un terminal local.

**La figura muestra el esquema de este proceso que consta de los pasos siguientes:**

1. Establecer conexión con el servidor remoto usando ssh (canal *cifrado*).
2. Por otro lado, desde la máquina local se ejecuta *xterm,* que es el terminal virtual utilizado en el sistema gráfico X Window.
3. *Xterm* se comunica directamente con el servidor X que está corriendo en la máquina (comunicación *sin cifrar*).



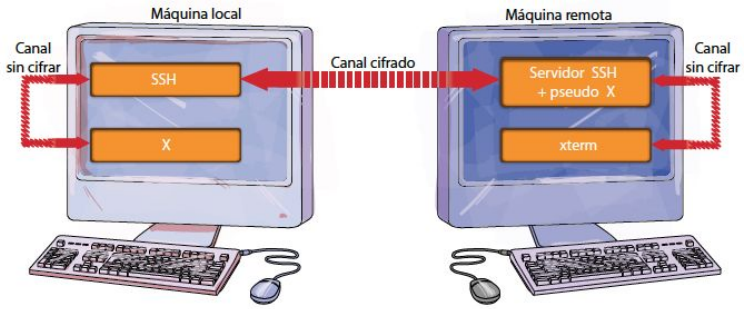



The McGraw-Hill Companies

# 4. ¿Qué es un cliente SSH?

## 4.2. Reenvío X11

A partir de ahora, todo lo que se teclea en la ventana de xterm (contraseñas incluidas) se transmite en texto plano por la red, sin aprovechar las ventajas de utilizar SSH. Sería preferible que se pudiera utilizar el mismo canal seguro que se establece con SSH para los clientes X. El reenvío (forwarding) de X11 es un mecanismo que permite la utilización del canal SSH. El esquema de funcionamiento es el que se muestra en la figura.




En este segundo caso se procede de este modo:

1. Conectar al servidor remoto usando `ssh` (comunicación cifrada). El servidor de SSH crea un pseudoservidor de X en la máquina remota.
2. Desde la sesión en la máquina remota, se ejecuta `xterm`.
3. Xterm se comunica con el pseudoservidor de X. Al hallarse en la máquina remota, se produce una comunicación no cifrada pero que no viaja por la red.
4. El pseudoservidor de X se comunica con el cliente de SSH a través del canal cifrado.
5. El cliente de SSH transmite los datos de xterm al servidor X real (comunicación sin cifrar, pero es local a nuestra máquina).

La orden que hay que lanzar para establecer la terminal xterm en la máquina remota bajo SSH es:

```
$ ssh -n usuario@maquina_remota xterm &
```

En estas circunstancias la aplicación ejecutada consumirá CPU de la máquina remota (servidor SSH) y en la máquina local (cliente SSH) solo servirá la pantalla.



The McGraw-Hill Companies

# 4. ¿Qué es un cliente SSH?

## 4.3. Reenvío por TCP/IP

Se basa en la asignación de un puerto local del cliente a un puerto remoto del servidor. De este modo, la información cuyo destino es un puerto de la máquina local se puede enviar a otro puerto de una máquina remota. Si el reenvío por TCP/IP se configura para escuchar por puertos inferiores a 1024, es preciso acceder como root.

Cuando se utiliza esta técnica, el servidor SSH se convierte en un túnel encriptado para el cliente SSH. El usuario tan solo ha de tener una cuenta en el sistema remoto. Esta técnica también se conoce como «mapeado de puertos».

El reenvío por TCP/IP puede usarse para aquellos protocolos que no tienen soporte nativo para comunicaciones cifradas y autenticadas, como por ejemplo POP, IMAP, FTP (cuando no se pueda utilizar el cliente sftp), etcétera. Los servicios basados en estos protocolos pueden hacerse más seguros mediante estos túneles SSH.

La sintaxis para crear un túnel de reenvío local por TCP/IP es la siguiente:

```
ssh -L puerto_local:maquina_local:puerto_remoto nombre_usuario@maquina_remota
```

Donde la opción -L indica que se trata de un reenvío local.

El protocolo IMAP es un buen ejemplo a la hora de probar el túnel SSH. Los clientes de correo envían constantemente el login y la contraseña del usuario al servidor de correo para sondear si se han recibido nuevos mensajes. Toda esta información viaja en texto plano, por lo que puede ser capturada y analizada, y el usuario perdería la confidencialidad de su correo.



The McGraw-Hill Companies

# 4. ¿Qué es un cliente SSH?

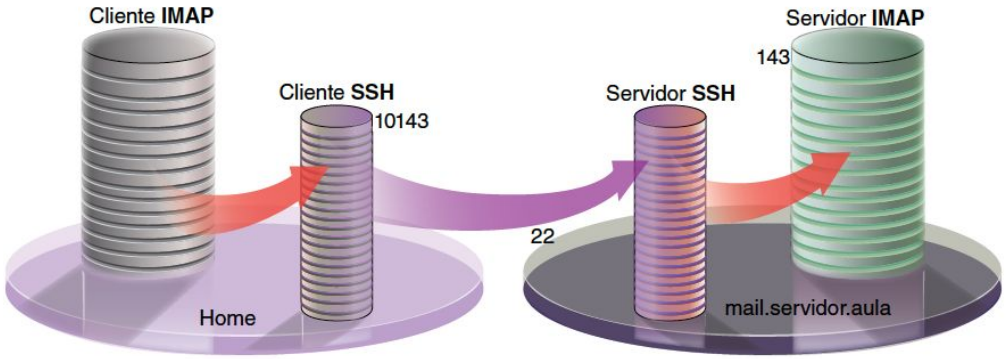
## 4.3. Reenvío por TCP/IP

Tenemos el siguiente ejemplo: el usuario *alumno1* quiere comprobar su correo en el servidor (remoto) mail.servidor.aula mediante el uso de IMAP a través de una conexión segura. Para ello reenviará sus peticiones IMAP desde el puerto 10143 local a través de la conexión SSH al puerto 143 de la máquina remota: `$ ssh -L 10143:localhost:143 alumno1@mail.servidor.aula`

Esta orden funcionará solo en el caso de que en el servidor remoto esté ejecutándose un servidor SSH y el servicio al que se quiere acceder. Indica que cualquier petición enviada al puerto 10143 en el sistema local será dirigida de forma segura al servidor mail.

También se puede crear el túnel, aunque la propia máquina remota no tenga el servidor SSH ejecutándose, siempre que se tenga una máquina intermedia, ya sea el propio cortafuegos u otra máquina detrás del DMZ con el puerto 22/tcp abierto. En este caso hay tres máquinas implicadas: el origen y dos destinos, uno de ellos, el que tiene el servidor SSH activo, hace de intermediario. Sin embargo, la comunicación solo sería segura en el primer tramo, es decir, hasta la máquina intermedia.

Si el administrador del sistema no quiere dejar activa la posibilidad de utilizar el reenvío de puertos tiene que, en el servidor SSH, deshabilitar la opción `AllowTcpForwarding` localizada en el archivo `/etc/ssh/sshd_config` y reiniciar el servicio SSH.






The McGraw-Hill Companies

# 5. ¿Qué es un servidor SSH?

El servidor SSH facilita el establecimiento de conexiones remotas que permiten la transmisión segura de cualquier tipo de datos: archivos, contraseñas, ejecución de órdenes de administración en un sistema remoto, sesiones de login, sesiones gráficas, etcétera. Una vez ha sido descrita la funcionalidad básica de SSH en general, a partir de este punto se va a tomar como herramienta base OpenSSH, que es la implementación libre más utilizada del servicio SSH.

**Las características más importantes de OpenSSH son:**

1. Proyecto de Código Abierto, disponible para su descarga de Internet.

2. Tiene licencia libre que permite su utilización para cualquier propósito, incluido el comercial.

3. Es compatible con los protocolos SSH1 y SSH2.

4. Está disponible para plataformas GNU/Linux y Windows, así como Unix, Mac, Solaris, AIX y otras.

5. Reenvío por puertos.

6. Reenvío por agente.

7. Soporte para cliente y servidor de SFTP en los protocolos SSH1 y SSH2.

8. Compresión de datos.



The McGraw-Hill Companies

# 5. ¿Qué es un servidor SSH?

## 5.1. Instalación del servidor SSH (GNU/Linux) con Webmin

Webmin dispone de un módulo específico que permite administrar las conexiones SSH. Este módulo es estándar, se llama webmin-sshd y está disponible directamente en la web oficial de Webmin.

## 5.2. Archivos de configuración del servidor SSH

| Archivo | Descripción |
|---|---|
| `sshd_config` | Describe la configuración del servidor SSH. Permite configurar opciones como el puerto de escucha, la versión del protocolo, dónde se encuentra la clave privada de la máquina o si esta ha sido generada con RSA o DSA. Si se permite la autenticación de usuarios mediante clave pública, hay que activar la opción *PubkeyAuthentication* e indicar dónde están guardadas (opción *AuthorizedKeysFile*). Normalmente se encuentra en `~/.ssh/authorized_keys`. |
| `ssh_config` | Describe la configuración del cliente SSH. Es posible que un mismo cliente tenga opciones de conexión diferentes en función de la máquina destino. Esto se indica mediante diferentes secciones host. |
| `ssh_host_rsa_key` | Clave RSA privada de la máquina. |
| `ssh_host_rsa_key.pub` | Clave RSA pública de la máquina. |
| `known_hosts` | Claves públicas de otras máquinas. |
| `ssh_host_dsa_key` | Clave DSA privada de la máquina. |
| `ssh_host_dsa_key.pub` | Clave DSA pública de la máquina. |



The McGraw-Hill Companies

# 5. ¿Qué es un servidor SSH?

## 5.3. Autenticación de usuarios

Existen varios métodos de autenticación de usuarios. A continuación, se explican dos métodos mutuamente excluyentes, es decir, el servidor y el cliente deben utilizar el mismo, por lo que automáticamente se descarta el otro.

• **Autenticación por contraseña**

SSH permite autenticar a un usuario utilizando su contraseña. Para ello, cada vez que el usuario quiera establecer una conexión, se le pide una contraseña que se envía al servidor. Este comprueba que el usuario existe y que la clave introducida es correcta. La validación utilizada en el servidor se basará en el archivo /etc/shadow, el procedimiento típico para los sistemas Unix o basados en él.

Este método, aunque tiene el inconveniente de requerir al usuario su contraseña cada vez que quiera establecer una sesión, por lo menos no realiza el envío del login y la contraseña en texto plano.

• **Autenticación por clave pública**

La segunda alternativa de autenticación utiliza un esquema de clave pública/privada generadas por el usuario. También se conoce como clave asimétrica y se aplica al usuario. En este caso se recurre a los elementos siguientes:

1. Una clave pública, que se copia a todos los servidores a los que el usuario quiere conectarse.

2. Una clave privada que solo posee el usuario. Para mayor seguridad, está cifrada con una frase de paso.



The McGraw-Hill Companies

# 5. ¿Qué es un servidor SSH?

## 5.3. Autenticación de usuarios

Ambas claves poseen una característica importante: un texto cifrado con la clave pública solo puede descifrarse mediante la clave privada correspondiente, mientras que un texto cifrado con la clave privada solo puede descifrarse mediante su clave pública asociada.

**¿Cómo se aplica esta propiedad al proceso de autenticación del usuario?**

1. Una vez establecida la conexión, el servidor genera un número aleatorio que se conoce con el nombre de «desafío» (challenge), cifrado con la clave pública del usuario mediante el algoritmo RSA o DSA. Este texto cifrado se envía al usuario.

2. El usuario debe descifrarlo con la clave, privada correspondiente y devolver la respuesta cifrada al servidor. De esa forma, demuestra que el usuario es quien dice ser.

3. El servidor descifra el texto de respuesta con la clave pública del usuario.

4. El servidor compara el texto resultante con el texto original. Si coinciden el servidor acepta al usuario como correctamente autenticado.



The McGraw-Hill Companies

# 5. ¿Qué es un servidor SSH?

## 5.4. Autenticación SSH por contraseña

La configuración de la autenticación de usuario por contraseña está disponible en la opción de menú:

*Webmin > Servidores > Servidor SSH > Autenticación*

Desde allí se establecen los procedimientos de autenticación de usuarios.

El archivo de claves públicas autorizadas es ~/.ssh/authorized_keys y, como su path indica, se encuentra en el directorio home del usuario.

Es preferible siempre tener desactivada la opción de archivos .rhosts. En realidad el servicio SSH apareció para solucionar los problemas de seguridad de Telnet y los comandos «r».

Activando las opciones correspondientes se puede configurar la autenticación por clave pública desde Webmin.




The McGraw-Hill Companies

# 5. ¿Qué es un servidor SSH?

## 5.5. Otras opciones del módulo SSH de Webmin

***Servidor SSH > Control de Acceso***
Desde *Servidor SSH* también se puede acceder a *Control de Acceso,* una opción desde la que se puede conceder o denegar el paso a los usuarios.


***Servidor SSH > Opciones varias***
De ellas las más significativas son la que activa el reenvío X y la que permite seleccionar el sistema de log que se utilizará.

***Servidor SSH > Opciones de máquina cliente***
Establece opciones para uno, varios o todos los clientes SSH: puertos de reenvío local y remoto, compresión del tráfico, número de intentos de conexión, protocolos SSH a probar, etcétera.


***Servidor SSH > Configuración de clave de SSH***
Permite establecer una configuración de SSH por defecto para nuevos los usuarios que se vayan creando en el sistema. Por ejemplo, se podría establecer que los nuevos usuarios no tengan que usar ssh-keygen antes de usar SSH, si se permite utilizar la contraseña como frase de paso, el tipo de clave, etcétera.



The McGraw-Hill Companies

# 5. ¿Qué es un servidor SSH?

## 5.6. Utilización básica de SSH

- El demonio servidor de OpenSSH es /usr/sbin/sshd.
- Por lo general, se activa durante el proceso init.
- El cliente de OpenSSH es /usr/bin/ssh.

**La orden ssh permite iniciar sesiones y ejecutar comandos de forma segura en máquinas remotas.**

La utilización básica del cliente es:

```
ssh [nombre_usuario@]maquina_remota
```

Otro modo habitual de utilizar ssh es el siguiente:

```
ssh -l usuario maquina_remota
```



The McGraw-Hill Companies

# 5. ¿Qué es un servidor SSH?

## 5.7. El agente de autenticación ssh-agent

El agente de autenticación permite simplificar el proceso de conexión a una máquina remota en caso de que se utilice autenticación por clave pública. El agente ssh-agent actúa como almacén de las claves privadas y las suministra al cliente SSH cada vez que este las necesita.

La forma de lanzar el agente, si se está utilizando sh o bash, es la siguiente:

```
alumno1@pc11:~$ eval `ssh-agent`

Agent pid 6194
```

La orden `eval` (eval [arg1 [arg2] ...]) expande sus argumentos siguiendo las normas de expansión de la shell, separándolos por espacios e intenta ejecutar la cadena que resulta como si fuera una orden.

Inicialmente el depósito de claves del agente está vacío. Para añadir nuestra clave privada RSA a este depósito se utiliza el comando `ssh-add`:

```
alumno1@pc11:~$ ssh-add .ssh/id_rsa

Enter passphrase for .ssh/id_rsa:

Identity added: .ssh/id_rsa

alumno@pc11:~$ ssh alumno@servidor.aulaSER.com

.....................................................................

alumno@servidor:~$
```



The McGraw-Hill Companies

# 7. Servidor SSH bajo Windows 2008 Server

Como se indicó en la introducción a esta unidad, la solución a los problemas de seguridad de Telnet se llama Secure Shell (SSH), en este caso para Windows. Existen varias herramientas SSH para Windows en el mercado, muchas propietarias y alguna libre. De ellas se ha elegido la opción libre freeSSHd ya que el nivel de complejidad de OpenSSH para Windows 2008 Server no lo hace recomendable.

**Las principales características de freeSSHd son las siguientes:**


- Permite la utilización de SSH sobre las interfaces de red que asignemos.
- Soporta varios métodos de autenticación, incluyendo autenticación integrada en Active Directory (NTLM).
- Soporta varios métodos de cifrado AES y otros (3DES, Blowfish, etcétera).
- Permite establecer túneles seguros.
- Selecciona aquellos usuarios a los que se le permite la conexión SSH, incluso la utilización de túneles y FTP seguro.
- Selecciona máquinas/redes a las que se les permite o deniega la conexión.
- Lleva un registro de todas las operaciones realizadas y permite controlar todas las sesiones abiertas.



The McGraw-Hill Companies

# 8. Servicios de Terminal Server

Por lo general, los servicios de terminal de Windows 2008 permiten que varios usuarios puedan iniciar sesión simultáneamente en el servidor.

Si varios usuarios se conectan de forma remota al servidor se está permitiendo la ejecución de aplicaciones y el control sobre los escritorios en dispositivos físicamente distantes Y el administrador puede tener también disponibles varias sesiones remotas y acceder desde ellas a los dispositivos locales y a las unidades de otros equipos Windows.

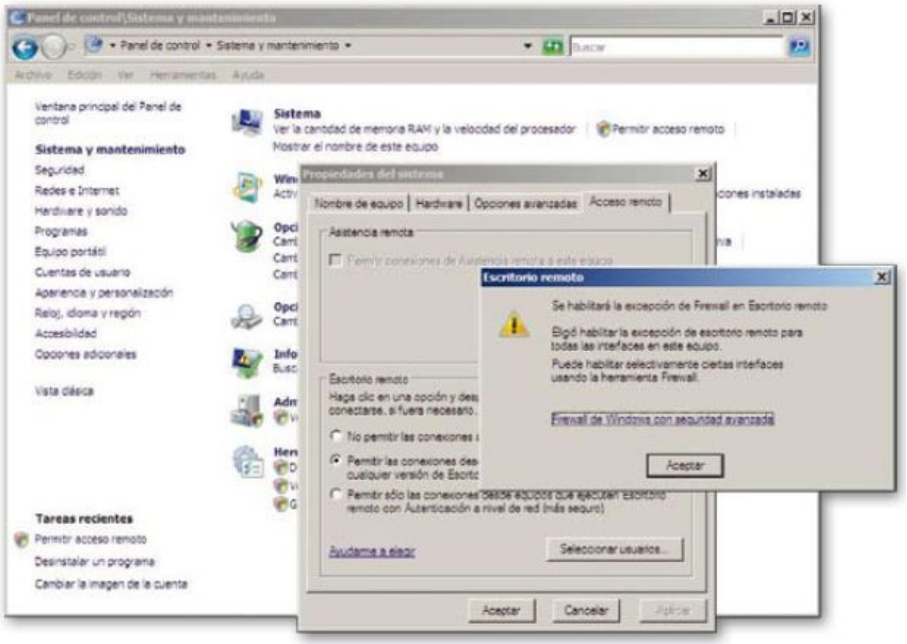
Dentro de los servicios de terminal se encuentra **Remote Desktop**, que proporciona una interfaz gráfica de usuario para el control remoto de los escritorios dentro de una red de área local.

**Activación de Remote Desktop: servidor**

El servicio de terminal Remote Desktop se instala por defecto con Windows 2008 Server, pero no queda activado.

Para activarlo hay que acceder a las propiedades del sistema; para ello, es preciso seguir este itinerario:

*Inicio > Panel de control > Sistema y mantenimiento > Permitir acceso remoto*






The McGraw-Hill Companies

# 8. Servicios de Terminal Server

**Activación de Remote Desktop: cliente**

Los servicios de Terminal Server para el cliente se pueden utilizar de diferentes formas.

El procedimiento más habitual de conexión al servidor es mediante la consola MMC (Microsoft Management Console) o también ejecutando el programa *mstsc.exe*.

Una vez hecha la validación del usuario, se inicia la sesión remota. Si el usuario que se conecta tiene privilegios de administrador, tendrá la oportunidad de realizar tareas de administración de forma remota.




The McGraw-Hill Companies

# 8. Servicios de Terminal Server

**La herramienta rdesktop**

Otra herramienta para el control remoto es *rdesktop*, un cliente Open Source para los servicios de Terminal Server de Windows. La herramienta rdesktop se comunica de forma nativa con el protocolo de escritorio remoto (RDP) para ofrecer un escritorio a los usuarios de Windows

**La aplicación tsclient**

http://gnomepro.com/tsclient

Disponible para Windows y GNU/Linux, actúa como interfaz gráfica de rdesktop. El acceso se lleva a cabo desde el menú de Ubuntu siguiendo este itinerario:

*Aplicaciones > Internet > Cliente de Terminal Server*

Desde *tsclient* es posible llevar a cabo conexiones remotas con cualquiera de los protocolos vistos hasta ahora, como RDP, VNC y XDMCP.




The McGraw-Hill Companies