SERVICIOS EN RED

C.F.G.M. SISTEMAS MICROINFORMÁTICOS Y REDES

Profesor: Jorge Martín Cabello

UD 4. DE ACCESO REMOTO

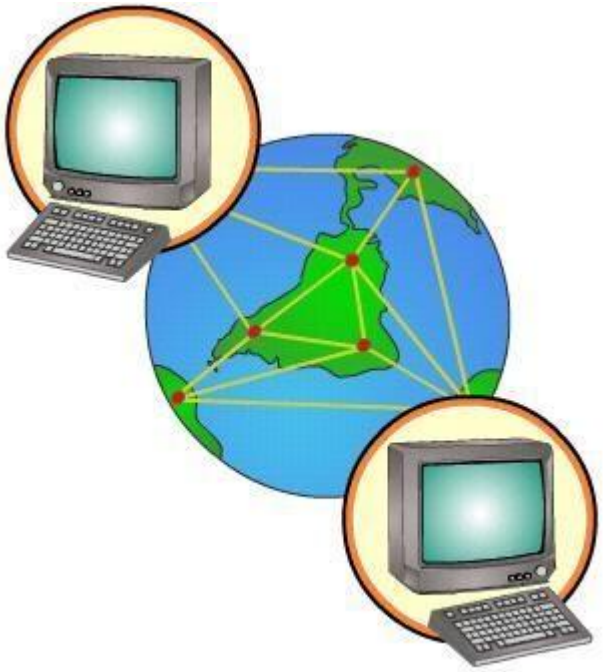


Tabla de contenido

1. TERMINAL TELNET

Telnet es un protocolo de red (Telecommunication Network) que sirve para conectarse a una máquina de una red desde otra para manejarla remotamente, como si estuviésemos sentados delante de ella.

Es un protocolo cliente-servidor que se comunica por el puerto 23. Antiguamente este tipo de acceso tenía mucho sentido, cuando los terminales o clientes eran máquinas muy lentas y los servidores máquinas muy potentes, por ejemplo, un astrónomo podía hacer cálculos en servidores de la NASA. Después, el sentido de esta herramienta se basó en poder administrar un ordenador remoto, configurarlo y solucionar errores.

Cuando hacemos un telnet a una máquina (con su nombre de dominio o IP), la máquina remota nos pide el nombre de usuario y la contraseña con la que conectarnos (algunos programas presumen que lo hacemos con el mismo usuario y contraseña de la sesión que tengamos abierta en nuestro ordenador cliente o local). El nombre de usuario, cuando son servidores públicos, suele ser: guest, visitor, new-user, etc., y la contraseña es pulsar la tecla ENTER.

Este protocolo proporciona reglas básicas que permiten vincular a un cliente (sistema compuesto de una pantalla y un teclado) con un intérprete de comandos (detallando del servidor: aplicaciones, procesador, disco duro, etc.). Al acceder, se abre un terminal gráfico en modo texto (que suele ser un Terminal Unix estándar: vt100 «virtual Terminal»).

La sintaxis es:

```
Linux:  telnet IPoNombre [puerto]
Windows:  telnet //IPoNombre
```

Ejemplo:

```
telnet  127.0.0.1
telnet  //127.0.0.1
telnet  locis.loc.gov
```

El telnet solo se usa en redes locales (de hecho la mayoría de sistemas operativos ya no lo permiten ni en LAN, como por ejemplo Windows), esto se debe a que las comunicaciones no están cifradas, es decir, no son seguras y se pueden descifrar con cualquier programa de sniffer. Además, permite conexión como superad ministrador o root.

En Windows 9x está activado el cliente por defecto. En Windows XP/2000/7 debes activarlo desde

INICIO> PANEL DECONTROL> HERRAMIENTASADMINISTRATIVAS>SERVICIOS

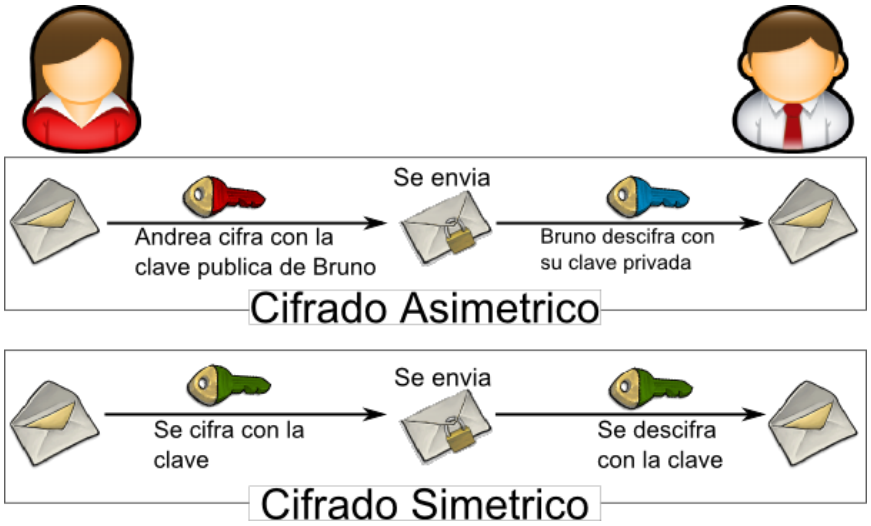
Si lo tienes instalado, en la lista debes activarlo con el botón secundario del ratón.

Pruébalo desde un navegador con

telnet://127.0.0.1

2. CRIPTOGRAFÍA.

Los sistemas de acceso remotos con seguridad utilizan algoritmos y claves de encriptación. Vamos a introducir conceptos básicos sobre encriptación y seguridad en la comunicación.



2.1. Encriptación simétrica

El cifrado mediante clave simétrica significa que dos o más usuarios, tienen una única clave secreta, esta clave será la que cifrará y descifrará la información transmitida a través del canal inseguro.

Es decir, la clave secreta la debe tener los dos usuarios, y con dicha clave, el usuario A cifrará la información, la mandará a través del canal inseguro, y a continuación el usuario B descifrará esa información con la MISMA clave que ha usado el usuario A.

Para que un algoritmo de clave simétrica sea fiable debe cumplir:

– Una vez que el mensaje es cifrado, no se puede obtener la clave de cifrado/descifrado ni tampoco el texto en claro.

– Si conocemos el texto en claro y el cifrado, se debe tardar más y gastar más dinero en obtener la clave, que el posible valor derivado de la información sustraída (texto en claro).

Debemos tener en cuenta que los algoritmos criptográficos son públicos, por lo que su fortaleza debe depender de su complejidad interna y de la longitud de la clave empleada para evitar los ataques de fuerza bruta.

La seguridad en clave simétrica reside en la propia clave secreta, y por tanto el principal problema es la distribución de esta clave a los distintos usuarios para cifrar y descifrar la información. La misión del emisor

y receptor es mantener la clave en secreto. Si cae en manos equivocadas ya no podríamos considerar que la comunicación es segura y deberíamos generar una nueva clave.

Otro problema reside en que las claves secretas a guardar es proporcional al número de canales seguros que deseamos mantener. Esto no es un problema en sí, pero debemos administrar bien las llaves para no equivocarnos. Este problema no se va a presentar en los algoritmos asimétricos porque cada usuario tiene una pareja de claves, una pública y la otra privada, independientemente del número de canales seguros que queramos establecer. Únicamente debe proteger la clave privada.

2.2. Algoritmos destacados en encriptación simétrica.

2.2.1. DES (Data Encryption Standard):

Algoritmo desarrollado por IBM EN LOS AÑOS 70 usa una clave simétrica de 64bits, los 56 primeros bits son empleados para el cifrado, y los 8 bits restantes se usan para comprobación de errores durante el proceso. La clave efectiva es de 56 bits, por tanto, tenemos 2^{56} combinaciones posibles, por lo que la fuerza bruta se hace casi imposible.

Ventajas:

– Es uno de los sistemas más empleados y extendidos, por tanto es de los más probados.

– Implementación sencilla y rápida.

Inconvenientes:

– No se permite una clave de longitud variable, es decir, no se puede aumentar para tener una mayor seguridad.

– Es vulnerable al criptoanálisis diferencial (2^{47} posibilidades) siempre que se conozco un número suficiente de textos en claro y cifrados.

– La longitud de clave de 56 bits es demasiado corta, y por tanto vulnerable. Actualmente DES ya no es un estándar, debido a que en 1999 fue roto por un ordenador.

La principal ventaja de los algoritmos simétricos es la velocidad de los algoritmos, y son muy usados para el cifrado de grandes cantidades de datos.

2.2.2. 3DES (Triple Data Encryption Standard)

Se basa en aplicar el algoritmo DES tres veces, la clave tiene una longitud de 128 bits. Si se cifra el mismo bloque de datos dos veces con dos llaves diferentes (de 64 bits), aumenta el tamaño de la clave.

El 3DES parte de una llave de 128 bits, que es divida en dos llaves, A y B.

Al recibir los datos, aplicamos el algoritmo DES con la llave A, a continuación se repite con la llave B y luego otra vez con la llave A (de nuevo).

3DES aumenta de forma significativa la seguridad del sistema de DES, pero requiere más recursos del ordenador.

Existe una variante del 3DES, conocida como DES-EDE3, con tres claves diferentes y una longitud de 192bits, consiguiendo un sistema mucho más robusto.

.

2.2.3. IDEA (International Data Encriptión Algorithm)

Desarrollado por la escuela politécnica de Zurich en 1990, aplica una clave de 128 bits sin paridad a bloques de datos de 64 bits, y se usa tanto para cifrar como para descifrar.

Según numerosos expertos criptográficos, IDEA es el mejor algoritmo de cifrado de datos existente en la actualidad ya que existen 2^{128} claves privadas que probar mediante el ataque de fuerza bruta.

2.2.4. AES (Advanced Encryption Standard)

Fue estandarizado por el gobierno de EEUU en el año 2001. Este algoritmo es el más conocido entre los usuarios de routers, ya que el protocolo WPA opera con AES como método de cifrado. Este cifrado puede implementar tanto en sistemas hardware como en software. El sistema criptográfico AES opera con bloques y claves de longitudes variable, hay AES de 128bits, de 192 bits y de 256 bits.

2.3. Criptografía de clave asimétrica.

Esta criptografía se basa en dos claves distintas (de ahí el nombre de criptografía asimétrica). Una de las claves se denomina pública y la otra privada.

La clave pública (como su nombre indica) puede hacerse pública, por el contrario la clave privada sólo es conocida por el propietario de la misma.

Cuando una persona quiere firmar digitalmente un mensaje usa su clave privada, de esta forma cualquier persona que posea la clave pública del remitente podrá comprobar que el mensaje ha sido firmado correctamente.

Para cifrar un mensaje se usa la clave pública del destinatario, así cuando éste reciba el mensaje podrá usar su clave privada para descifrarlo y por tanto sólo él puede ver el contenido del mensaje.

2.4. Algoritmos de clave asimétrica RSA y DSA.

Cuando se trata de elegir entre generar una clave RSA o una DSA, la principal diferencia son las operaciones de encriptación que quieres realizar. Las claves RSA pueden encriptar todo el documento y la firma de archivo, mientras que una DSA solo se usa para firmar documentos. La RSA permite a los usuarios generar pares de claves cuyos tamaños son mayores de 1024 bits, que es el tamaño máximo de una clave DSA. Puesto que las claves son más difíciles de forzar cuando se aumenta su tamaño, quienes estén más preocupados por la seguridad deberían considerar usar claves RSA para sus firmas de archivo así como encriptación de documentos.

3. SSH (Secure Shell)

3.1. Introducción.

El SSH o Intérprete de Comandos Seguro es el protocolo que viene a resolver los problemas de seguridad de telnet.

La gran diferencia con telnet es que sí cifra las conexiones.

Normalmente, el protocolo SSH cifra con claves de los algoritmos RSA, pero puede hacerlo con claves y algoritmos DSA (Digital Signature Algorithm).

Las características básicas del protocolo SSH son:

- Autenticación: se autentifica el usuario mediante nombres de usuario y contraseñas (de usuario o host, públicas y privadas).
- Confidencialidad: se cifran las conexiones.
- Integridad: si por el camino el paquete es alterado, se puede detectar. Otras ventajas de este sistema son:
- No rechazo: si se contesta a esta comunicación no podemos negar quiénes somos.
- Evita programas sniffer.
- Evita la suplantación de host o Man in The Middle.
- Existen versiones para todos los sistemas operativos.
- Permite tunelización para FTP, SMTP, Messenger, etc.
- Permite la compresión de los paquetes.

El funcionamiento del SSH:

- Empieza cuando el cliente abre una conexión TCP en el puerto 22.
- El servidor y el cliente negocian la versión de SSH, el tipo de cifrado (RSA/DSA), etc.

- El servidor envía su clave pública al cliente.
- El cliente la compara con la lista de claves que tiene. Si es la primera vez, es el usuario el que indica si es válida o no. Esta fase es crítica, pues es el único momento en el que se puede suplantar a este equipo. Para evitarlo tenemos acceso a las claves en mano o desde intranet.
- El cliente genera una clave de sesión aleatoria, que es enviada al servidor dentro de un paquete cifrado con el algoritmo seleccionado y la clave pública.
- A partir de este momento, la comunicación se basa en el algoritmo simétrico de encriptación seleccionado.

Algunos conceptos a tener en cuenta :

- Tunelización: Acción de encapsular un protocolo por otro. Se usa para saltarse los cortafuegos o encriptar protocolos inseguros.

- Spoofing: Suplantación de personalidad (de usuario y/o host en este caso)

3.2. Instalación de Clientes SSH en terminal en Ubuntu.

En Linux podemos instalar el cliente y servidor SSH

```
Servidor :# apt-get install ssh
Cliente: # apt-get install openssh-client
```

Los tipos de clientes SSH permiten:

- SSH: encapsular otros protocolos, tunelizar, etc.
- SFTP: la transferencia de archivos seguros.
- SSMTP: el envío seguro de correo electrónico.
- Genéricos: navegadores que permiten los protocolos HTTPS, SFTP, SSMTP, etc

Los parámetros de conexión en el cliente son:

```
ssh [-p puerto] [usuario@] ip_del_servidor
Ejemplo1: ssh –p 22 admin@127.0.0.1
Ejemplo2: ssh root@127.0.0.1
```

Los parámetros básicos de configuración del cliente se encuentran en “ /etc/ssh/ssh_config”

y son (primero especificamos los valores por defecto):

• Host * / 127.0.0.1 / nombre-dominio : restringe a qué servidores podemos conectarnos, el comodín asterisco (*) permite conectarnos a todos. Se puede especificar *.org; 192.168.0.* (toda esa subred) o 10.0.0.[1-9] (solo de ese equipo, terminado en 1 hasta el 9, ambos inclusive).

• Port 22/otro_puerto: puerto que podemos cambiar.

• Protocol 2/1: versión del protocolo SSH.

• PubkeyAuthentication yes/no : autenticación por clave pública.

• PasswordAuthentication yes/no : autenticación por contraseña.

• IdentifyFile ruta/archivo ~/.ssh/id_rsa y ~/.ssh/id_dsa : por defecto, identifica al usuario con clave RSA o DSA.

• ForwardX11 no/yes : activar el sistema gráfico XWindows versión 11.

• Compression no/yes : activa o desactiva la compresión.

• HostbasedAuthentication no/yes : autenticación adicional por host (rhost).

• RhostsRSAAuthentication no/yes: autenticación adicional por host en la versión RSA.

• RSAAuthentication yes/no : activar la autenticación con el algoritmo de encriptación asimétrica RSA.

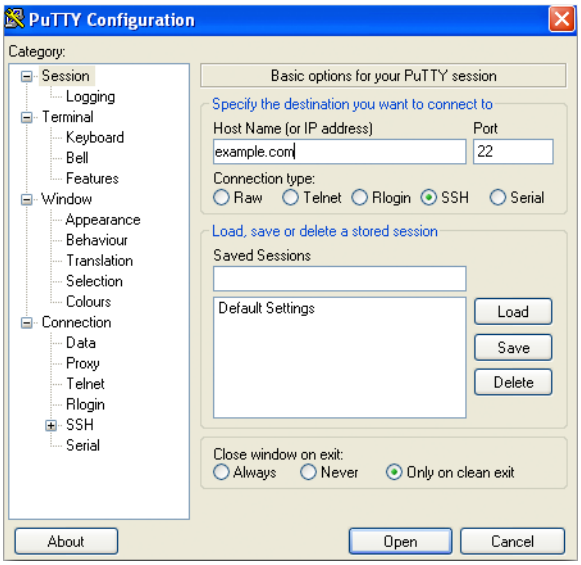
Los parámetros de control y autenticación anteriores se pueden configurar desde el terminal con la orden ssh:

```
ssh [–o orden parametro] [user@]nombre-host
Ejemplo: #ssh –o Compression yes root@127.0.0.1
```

3.3. Cliente SSH en Windows.

Para conectarnos a servidores SSH, aun en modo texto, debemos instalar, en los sistemas Windows, programas adicionales como: PuTTY (el más usado), FileZilla (para SFTP), OpenSSH,etc..

PuTTY permite realizar TELNET, RLOGIN, pero por defecto está configurado el SSH por el puerto 22, con el algoritmo IDEA y la versión 2. Podemos descargarlo en www.putty.org



3.4. Instalación de Servidor SSH.

En Linux podemos instalar el paquete del repositorio ssh ó openssh-server

Una vez instalado podemos arrancar o parar el servicio con #service ssh (stop, start, restart)

En Windows podemos acceder a www.OpenSSH.org y bajar el instalador de OpenSSH.

También podemos instalar otros servidores SSH para Windows como FreeSSHd .

OpenSSH crea su carpeta de archivos en c:/Archivos de programas/OpenSSH/etc

Podemos iniciar y parar el servicio en

Panel de Control/Herramientas Administrativas/Servicios/OpenSSH

3.5. Ficheros y parámetros de configuración.

El fichero de configuración del servidor es sshd_config, y se encuentra en la Carpeta / etc/ssh/ de Linux y en Windows en: C:\Archivos de programa\Open SSH\etc.

Sus parámetros de configuración básicos son (omitimos los que coinciden con los del cliente):

•PermitRootLogin yes|no: permite conectarse a la máquina remota como superusuario (root).

•ListenAddress 0.0.0.0: direcciones IP que escucha. 0.0.0.0 es todo internet con IPv4; :: es todo internet con IPv6.

•HostKey /etc/ssh/ssh_host_rsa_key : lugar donde guarda las claves RSA o DSA (ssh_host_dsa_key).

•KeyRegenerationInterval 1h: si queremos forzar la regeneración de la contraseña, ponemos los segundos, minutos, horas, etc.

•ServerKeyBits 768: longitud en bits de la clave, 512 es insegura y 1024 eleva las necesidades de computación.

•AuthorizedKeysFile .ssh/authorized_keys: lista de claves de usuarios autorizados.

•X11Forwarding no|yes: activar el sistema gráfico XWindows versión 11.

En modo gráfico de Linux podemos usar Webmin:

- Autenticación (Authentication) : permiso para conectarse como root, si solo deja conectarse a máquinas reconocidas, etc.
- En Red (Networking): direcciones IP que atenderá, si se activa el reenvío, etc.
- Control de Acceso (Access Control) : usuarios permitidos, etc.
- Opciones Varias (Miscellaneous Options): activación de las X11, intervalo de re-generación de clave, etc.
- Opciones de máquina-cliente (Client Host Options): configuración particular para los usuarios (compresión, etc.)

3.6. Autenticación de usuarios

Las claves de usuario están en los archivos:

ssh_host_rsa_key , ssh_host_rsa_key.pub, ssh_host_dsa_key y ssh_host_rsa_key.pub.

En el servidor, el archivo ssh/autho rized_keys tiene las claves de los usuarios permitidos, que siempre serán un subconjunto de los usuarios locales del servidor.

En Windows debemos crear los usuarios, para ello desde la carpeta \bin utilizamos la orden mkpassd.

```
mkpassd –l [ –u usuario] >> ..\etc\passwd
Ejemplo: mkpassd –l –u joaquin >> ..\etc\passwd
```

3.7. Túneles SSH

Los túneles encriptan otros protocolos (FTP, SMTP, etc.) y los convierten en seguros o los centralizan para pasarlos por el cortafuegos, etc. Desde línea de comandos podemos crearlo con:

```
ssh –L puertolocal:servidor:puertoservidor [usuario@]servidor
Ejemplo: ssh –L 10443:smtp.gmail.com:443 usuario@smtp.gmail.com
```

3.8. Transferencia segura de archivos usando SSH (sftp y scp)

Cuando queremos transferir archivos de forma segura (cosa más que recomendable para evitar el envío de virus a servidores, por privacidad, etc.) utilizamos la orden sftp (existe en todos los sistemas operativos al instalar el cliente SSH). Su sintaxis es:

```
sftp [usuario@]servidor
```

Para copiar archivos podemos usar scp, cuya sintaxis es:

```
scp origen destino
```

Donde ORIGEN y DESTINO son el path de los archivos en el local o el servidor, por ejemplo en el local /etc/imagen1.jpg y en el remoto usuario@servidor:/path/ archivo . El comando scp funciona igual que cp , con su sintaxis, comodines, etc.

Ejemplo que copia del local al servidor:

```
scp /etc/*.jpg joaquin@gmail.com:/images
```

O al revés, si queremos copiar del servidor al cliente:

```
scp joaquin@gmail.com:/images/*.jpg /etc/
```

Para saber más sobre el comando sftp utiliza la ayuda en Linux:

$man sftp

y para el comando scp:

$man scp

Para servidores sftp, puedes consultar en internet el siguiente enlace:

http://www.openbsd.org/cgi-bin/man.cgi?query=sftp-serve

Para sftp podemos usar las órdenes que ya conocemos sobre administración de archivos y que ampliaremos en el siguiente tema:

ls, chmod , cp, exit, get, put, help, lls, pwd, lpwd, mkdir, lmkdir, quit, rename, rm, rmdir, …

4. TERMINALES EN MODO GRÁFICO: ESCRITORIO REMOTO

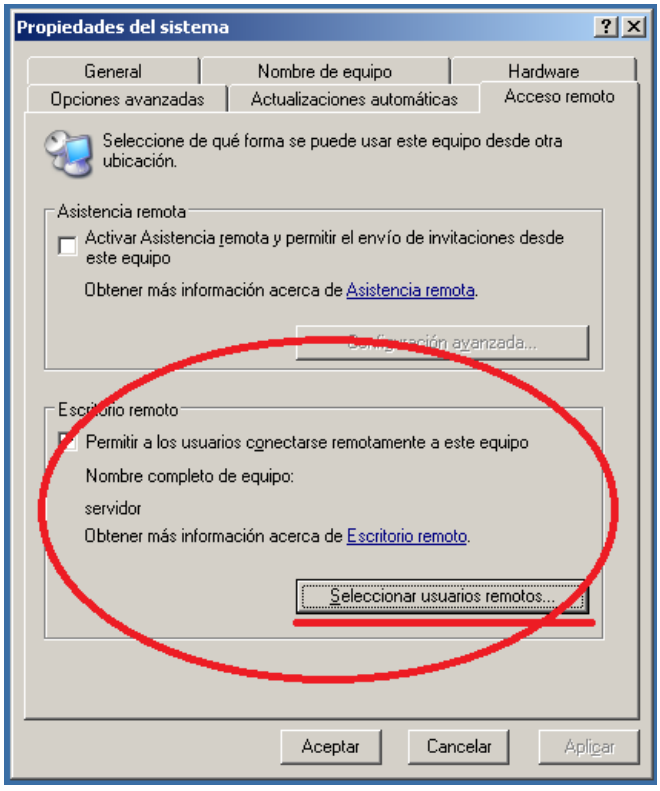
4.1. Terminales en modo gráfico:

Escritorio remoto Existen diversos programas que nos permiten manejar un equipo desde otro, de foma remota. Unos envían como fichero de imagen (generalmente en formato jpg) lo que sucede en la pantalla del servidor y otros solo envían las coordenadas X e Y del cursor y simulan el sistema operativo anfitrión. Hay algunos muy versátiles y otros específicos (como los que se utilizan en teleformación, etc.).

4.2. Protocolo RDP (Remote Desktop Protocol)

Remote Desktop Protocol (RDP) es un protocolo propietario desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre un terminal (mostrando la información procesada que recibe del servidor) y un servidor Windows (recibiendo la información dada por el usuario en el terminal mediante el ratón ó el teclado).El modo de funcionamiento del protocolo es sencillo. La información gráfica que genera el servidor es convertida a un formato propio RDP y enviada a través de la red al terminal, que interpretará la información contenida en el paquete del protocolo para reconstruir la imagen a mostrar en la pantalla del terminal.

Este servicio utiliza por defecto el puerto TCP 3389 en el servidor para recibir las peticiones. Una vez iniciada la sesión desde un punto remoto el ordenador servidor mostrará la pantalla de bienvenida de windows, no se verá lo que el usuario está realizando de forma remota. Para activar la posibilidad de permitir el acceso al escritorio remoto de nuestro servidor Windows 2003,, para posibilitar su administración de forma remota tenemos que seguir las siguientes opciones:



Inicio>Panel de control>Sistema

Pulsa en la solapa Remoto y marca la casilla *Permitir que los usuarios se conecten de manera remota* a este equipo, tal como se muestra en la siguiente figura.

Además desde esta pantalla podemos “Seleccionar usuarios remotos”, los usuarios que podrán administrar de forma remota el servidor.

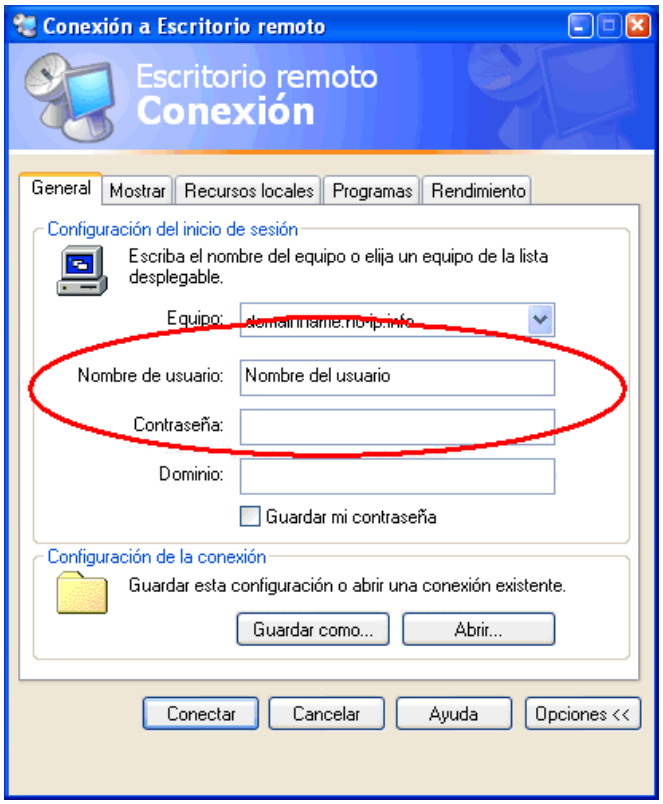
4.3. Clientes de escritorio remoto

Los clientes de escritorio remoto son programas que se conectan a un servidor para administrarlo, ejecutar programas en máquinas más potentes, etc. Se suelen utilizar para la administración remota y así evitar desplazamientos, para acceder a servidores que no tienen pantalla, para teleformación, para solucionar problemas de configuración, etc.

Ejecutaremos el programa Conexión a Escritorio remoto, el cual está disponible accediendo desde

Inicio>Todos los programas>Accesorios>Comunicaciones.

En la primera ficha (General) podemos indicar el nombre del usuario y el password, tal como se muestra en la figura. Si no indicamos estos datos, al hacer login en el equipo, se nos pedirán, y para que la contraseña se incluya, tendremos que marcar la opción Guardar mi contraseña, algo que no es recomendable hacer salvo que nos conectemos desde un equipo de nuestra propiedad. Cosas de la seguridad...

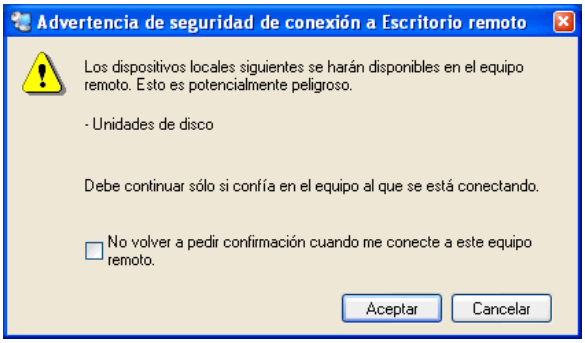


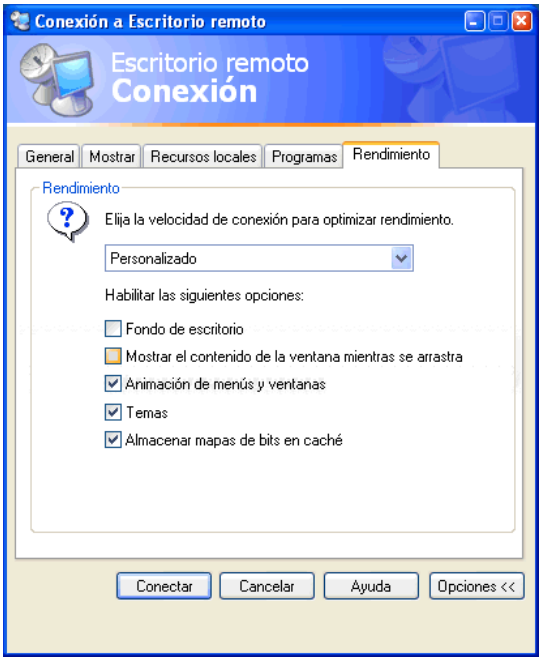
Mediante la segunda ficha (**Mostrar**), podemos indicar la resolución que usaremos además del número de colores, pero, tal como se indica en dicha ficha, esos colores dependerán de la configuración del equipo remoto.

Si llevamos el indicador del tamaño a usar hasta la parte derecha (**Más**), se mostrará a pantalla completa.

En la ficha **Recursos locales** podemos indicar que es lo que queremos hacer con los recursos de el equipo remoto y los locales. De forma que podamos "traer" el sonido del equipo remoto a nuestro equipo o que podamos compartir con el equipo remoto nuestras unidades de disco, impresoras y puertos de serie.

Si marcamos la opción de conectar las unidades locales, al conectar nos preguntará si estamos seguros de hacerlo, ya que puede suponer un problema de seguridad.



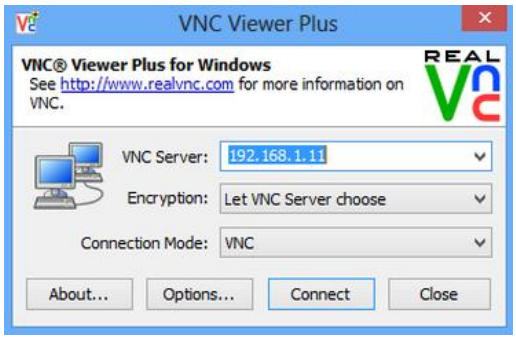


En la ficha **Rendimiento** indicaremos las opciones "gráficas" que queremos habilitar al conectar remotamente. Dependiendo de la velocidad de conexión que tengamos, podemos seleccionarlas todas o solo las que nos interesen.

5. VIRTUAL NETWORK COMPUTING VNC

5.1. Funcionamiento y características.

El VNC o programa de Computación en Red Virtual es otro programa de administración remota de código libre. El servidor utiliza el puerto 5900, los clientes de Windows el puerto 5800, y los de Linux y Mac OS el terminal que no estén usando (por ejemplo, si solo hemos activado un terminal, el siguiente puede acceder desde el puerto 5801, si tenemos activados cuatro debemos conectarnos por el puerto 5804, etc.).



Las características más importantes de VNC son:

- Permite crear pantallas virtuales en Linux y Mac OS, pero solo puede compartir la pantalla actual en Windows.
- En algunas versiones puede compartir la pantalla con varios clientes a la vez.
- Permite codificación IDEA de hasta 128 bits para encriptar y RSA de hasta 2048 bits para autenticar.
- Permite compartir impresoras.
- Permite FTP seguro.
- Permite chat seguro.
- Puede compartir aplicaciones con servidores Windows.

5.2. Clientes VNC

Existen distintas versiones de clientes VNC (RealVNC, TightVNC, UltraVNC,etc.) que están diseñadas para todos los sistemas operativos. RealVNC es uno de estos programas, que podremos bajar de

http://www.realvnc.com y sirve para todos los sistemas operativos.

Desde Linux podemos bajarnos el paquete vncviewer.

En el cliente RealVNC, nos pide el SERVER (nombre o IP), y en OPCIONES podemos elegir número de colores, pantalla completa, etc. (desde Linux podemos usar #vncviewer IP)

.

5.3. Servidores VNC

Desde terminal Linux podemos instalar vncserver *#apt-get install vncserver*

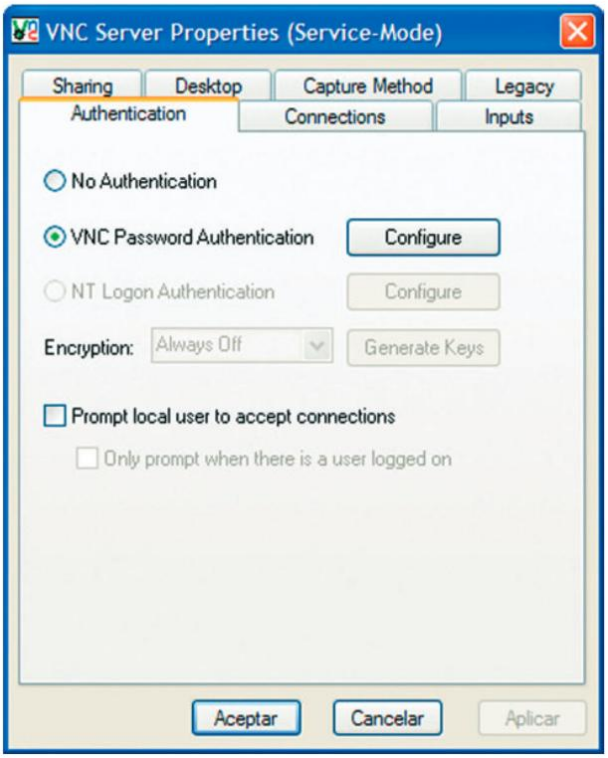
En el caso de Linux, Mac OS y Windows, RealVNC tiene una versión de servidor que se queda residente al instalar. Sobre el icono VNC (junto al de los programas residentes, junto a la hora, ver imagen al margen) podemos:

•ADDNEWCLIENT: añadir nuevos clientes.

•DISCONNECTCLIENTS: desconectar a los clientes.

•CLOSEVNC SERVER: parar el servidor.

•OPTIONS: modificar las opciones básicas (contraseñas, puertos, si se visualiza elescritorio o la opción «por defecto», etc.).Como mínimo debemos entrar en VNC PASSWORD AUTHENTICATION y poner una contraseña.



Otras opciones interesantes son:

•DESKTOP: para conservar el fondo de escritorio, etc.

•CONNECTIONS: si queremos configurar los puertos de conexión.

•INPUTS: donde aceptamos entrada desde el cliente de teclados, ratón, imprimir pantallas, etc.