FTP

1.− Introducción
2.− Consejos para usar FTP
3.− FTP Off−Line
4.− Configuración de WS_FTP

5.− ¿En qué consiste el protocolo FTP?

6.− ¿Cuáles son los comandos más usuales?

7.− ¿Cómo se trabaja con FTP anónimo?

8.− Transferencia de ficheros

9.− ¿Cómo configurar un servidor de FTP anónimo seguro?

10.− El archivo /etc/ftpaccess

11.− El archivo /etc/ftpconversions

12.− El archivo /etc/ftphosts

13.− El archivo /var/log/xferlog

14.− Problemas de Seguridad en servidores de FTP

15.− Descripción General de cómo configurar un servidor de FTP anónimo

- **Creación usuario FTP**
- **Permisos y contenido de los directorios**
- **Cómo permitir la escritura en un servidor anónimo**
- **Extensiones del servicio de FTP**
- **Detalles de seguridad**

16.− ¿Cómo saber si un servidor es seguro?

17.− Políticas de seguridad

18.− ¿Qué hacer si nuestro servidor está siendo usado ilegalmente?

19.− ¿Dónde buscar servidores de FTP? La solución : Archie

20.− Como actualizar nuestras página Web

1. ¿Qué datos necesito conocer?

2. ¿Cómo pongo estos datos en mi programa?

- **Ya estoy conectado pero no entiendo lo que aparece en pantalla**

- **¿Cómo puedo enviar mis páginas al servidor?**
- **¿Cómo le indico al servidor cual es mi página inicial?**

- **Introducción**

FTP *(File Transfer Protocol)*, es uno de los servicios más útiles a la hora de transmitir y recibir ficheros de cualquier tipo. Funciona con protocolo **TCP/IP**, permite acceder al servidor para recibir o transmitir ficheros.

En Internet existen millones de ficheros distribuidos en miles de ordenadores, que pueden ser copiados libremente usando **FTP**. En ocasiones, necesitarás una contraseña de acceso al sistema ***(password)*** para conseguir los ficheros. En otras, no es necesaria y podrás acceder de forma libre y sis restricciones: este procedimiento recibe el nombre de **FTP *anónimo***.

Mediante **FTP** anónimo tendrás acceso a una fuente inmensa de recursos informáticos, sistemas operativos, fotos, etc. Para extraer ficheros es necesario conocer la dirección de la máquina y que ese ordenador esté funcionando con un servidor de **FTP**. También existen servidores dedicados a recoger todo tipo de software con licencia *shareware* y *freeware* para cualquier sistema operativo. Además, podemos encontrar las **FAQ** *(Frequently Asked Questions)* que tienen respuestas a las preguntas más frecuentes realizadas por los usuarios de Internet o los documentos **FYI** *(For Your Information)* con información de todo tipo sobre la Red.

2. Consejos para usar FTP

- Es imprescindible que conozcas el tipo de fichero del que se trata. Si no fuera así, puede que el fichero obtenido contenga errores que lo hagan inutilizable. Podemos distinguir dos tipos de ficheros básicos ASCII (texto estándar) y binarios. En un fichero ASCII la información está escrita en código ASCII (letras, números, signos) mientras que un fichero binario puede contener programas ejecutables, imágenes gráficas, ficheros de sonido, etc.
- Los sitios que ofrecen el servicio de **FTP** anónimo, pueden atender a un número limitado de usuarios al mismo tiempo; por lo que muchas veces no puede realizarse la conexión apareciendo el correspondiente mensaje del sistema remoto. En este caso, vuelve a intentarlo más tarde o prueba en otro anfitrión.

3. FTP *Off−Line* en Internet (mediante correo electrónico)

Aparte de los programas clientes para los distintos Sistemas Operativos , podemos hacer que un servidor ajeno a nosotros se encarge de capturar el fichero, fragmentarlo, codificarlo y enviárnoslo a nuestra dirección de correo electrónico, ahorrándonos tiempo y por supuesto dinero.

Para conseguir esto tenemos que conocer el nombre del fichero a capturar y la dirección del anfitrión que lo sirve, usando *Archie* (mirará en su índice y te mostrará las entradas que más se asemejan a la cadena de texto que le hayas indicado), o buscandolo mediante algún Robot como ; *FTP−search* en la WWW).

Una vez conocidos estos datos, solo tenemos que hacer lo siguiente:

- Enviamos un mail a ***ftpmail@ftpmail.ramona.vix.com***
- Dejamos el campo ***subject:*** vacio.
- En el cuerpo del mail ponemos lo siguiente:

connect (nombre de host)

binary

chdir (nombre de directorio)

get (nombre de archivo)

quit

- Y dependiendo del trabajo que tengan pendiente, en unas horas tendremos nuestro fichero en el buzón, sin ningún trabajo extra.

4. Configuración de WS_FTP

Una vez instalado, haciendo un doble clik en el icono correspondiente, obtendrás en la pantalla del ordenador la ventana **Propiedades de la sesión,** por encima de la ventana de aplicación. En ella, debes introducir todos los datos necesarios para realizar una conexión FTP.

Profile Name: Para guardar los datos de la conexión y poder así utilizarlos en otra ocasión, introducimos el nombre que mejor la identifique en este campo, que en nuestro caso es *cuenta personal en xana.*
Host Name: En él teclearemos la dirección del servidor con el que deseamos contactar, en este caso es *xana.valliniello.es.*
User ID: Identificación de tu terminal, en este caso es *asiv11.* Para FTP anónimo deberás activar la casilla **anonymous,** para que se completen los datos automáticamente.
Password: Introducir la contraseña, si es necesaria.Los botones **Delete** y **New** sirven para eliminar o crear respectivamente una entrada de la lista desplegable **Profile Name**. Una vez completados los campos debes hacer clic sobre el botón **Aceptar** para iniciar la conexión.

A continuación se mostrará una pantalla donde aparecerán los subdirectorios y ficheros de los ordenadores local y remoto.

5. ¿EN QUE CONSISTE EL PROTOCOLO FTP?

Una de las alternativas más importantes que nos permite Internet es la transferencia de archivos de un ordenador a otro desde cualquier parte del mundo. Para ello utilizamos el protocolo de transferencia de archivos o "ftp" (file transfer protocol).

Si a través de Telnet podemos tener acceso a archivos distantes de nosotros, mediante ftp podemos compartir (recibir y enviar) nuestros ficheros con otros ordenadores, siempre que el administrador de estos últimos nos lo permita.

Para explicar cómo se pueden compartir los archivos pongamos un ejemplo:

Imaginemos una empresa que tiene sucursales en varios lugares y quiere que todas ellas posean unos determinados archivos que se actualizan cada cierto tiempo desde la sede central de la misma. A su vez, también las sucursales que generan sus propios documentos quieren enviarlos a la sede central de la empresa.

Esto se consigue, montando un ordenador en la sede central con los archivos que quieran distribuirse y conectándolo a Internet, ofreciendo la posibilidad de efectuar la transferencia de esos ficheros mediante ftp. A su vez, las sucursales se pondrán en contacto con la sede central de la empresa utilizando la herramienta ftp de su ordenador (existen varios paquetes de software con esta herramienta comercializado por distintas empresas informáticas). La forma de conexión con la central es muy similar a la utilizada en Telnet: al contactar con la central, su ordenador pedirá la correspondiente identificación (*login y password*), tras lo cual, ya se podrá transferir cualquier fichero en ambas direcciones.

¿Cómo se aborta una transferencia de archivo?

Para abortar una transferencia de archivo, se utiliza la clave de interrupción de terminal (generalmente Ctrl−C). Las transferencias que se esten realizando se paran inmediatamente. El protocolo ftp para el comienzo (o recepción) de transferencias enviando primero un comando ABORT del protocolo de ftp al servidor remoto y rechaza cualquier información posterior recibida. La velocidad con que esto sea logrado depende de la capacidad del servidor remoto para procesar el comando ABORT. Si el servidor remoto no tolera ABORT, el cursor (**ftp>**) no aparece hasta que el servidor remoto complete el envío del archivo solicitado.

La diferencia principal que existe entre ftp y Telnet está en que mientras con Telnet podemos ver los ficheros en la pantalla de nuestro ordenador, con ftp los tenemos que traer desde el lugar al que estamos conectados hasta nuestro ordenador y después verlos.

6. ¿CUALES SON LOS COMANDOS MAS USUALES DE FTP?

La mayoría de los servidores de ficheros trabajan en UNIX. Por ello, los ficheros y los directorios tienen nombres con la convención UNIX pero al transferirlos a un PC adoptan automáticamente el formato del sistema operativo que posea el mismo.

El nombre del servidor que comunica con ftp puede ser especificado en la línea de comando. Si se especifica el servidor, ftp abre inmediatamente una conexión con el mismo (véase el comando "open"), de lo contrario, ftp espera las instrucciones del usuario.

El protocolo de transferencia de archivos especifica parámetros de transferencia de archivo de tipo, modo, forma y estructura. El protocolo ftp soporta los tipos de archivos ASCII y binario. ASCII es el tipo ftp por defecto.

Los comandos más comunes que podemos utilizar son los siguientes :

- **!** [comando [argumentos]]
  Llama a una shell en el huésped local. La variable de entorno de la shell especifica qué programa quiere llamar. El protocolo ftp invoca a /bin/sh si la shell no se ha definido. Si se especifica el comando, la shell lo ejecuta y regresa a ftp. De lo contrario, se invoca a una shell interactiva. Cuando la shell termina, regresa a ftp.
- **account** [contraseña].
  Suministra una contraseña suplementaria requerida por el sevidor para el acceso a los recursos, una vez que el login ha sido introducido con éxito.
- **append** archivo local [archivo remoto]
  Copia el archivo local al final del archivo remoto.
- **ascii**
  Transfiere los archivos en modo ASCII. Este es el valor por defecto.
- **bell**
  Sonido de una campana después de completarse cada transferencia de archivo.
- **binary**
  Transfiere los archivos en modo binario.
- **bye**
  Cierra la conexión con el huésped servidor. También, tecleando los caracteres de final de archivo (EOF) finaliza la sesión.
- **cd** directorio remoto
  Se introduce dentro del "directorio remoto" del servidor.
- **cdup**

Retrocede al directorio padre del directorio de trabajo actual dentro del servidor.

- **chmod** modo−nombre del archivo
  Cambia los permisos del archivo "nombre del archivo" en el sistema remoto al modo indicado con la instrucción.
- **close**
  Termina la conexión con el servidor. El comando close no sale de ftp.
- **cr** Cambia el retorno de carro alterado durante la recuperación de un archivo ascii.
- **delete** archivo remoto
  Elimina el archivo remoto. El archivo remoto puede ser un directorio vacio.
- **dir** [directorio remoto] [archivo local]
  Escribe un listado del directorio remoto u opcionalmente de un archivo local. Si ni el archivo local ni el directorio remoto se especifican, lista el directorio de trabajo actual.
- **disconnect**
  Un sinónimo de close.
- **form** formato
  Establece la forma de transferencia del archivo al formato indicado.
- **get** archivo remoto [archivo local]
  Copia el archivo remoto al archivo local. Si el archivo local no se especifica, ftp utiliza el nombre del archivo remoto especificado como el nombre de archivo local.
- **hash**
  Conmuta imprimiendo un signo de fragmentación (#) cada 1024 bytes transferidos.
- **help** [comando]
  Imprime un mensaje informativo del comando de ftp llamado. Si no se especifica el comando, nos da una lista de todos los comandos de ftp.
- **lcd** [directorio local]
  Situa el directorio de trabajo en el directorio local señalado. Si el directorio local no se especifica, se situa en el directorio de trabajo local del usuario.
- **ls** [directorio remoto] [archivo local]
  Escribe un listado del directorio remoto en el archivo local. El listado incluye toda la información del sistema dependiente que el servidor quiera incluir; por ejemplo, la que la mayoría de los sistemas de UNIX producen con el comando ls −l (ver también nlist). Si ni el archivo local ni el directorio remoto se especifican, lista el directorio de trabajo remoto.
- **mdelete** [archivos remotos]
  Elimina los archivos remotos.
- **mdir** archivos remotos archivo local
  Escribe un listado de los archivos remotos en el archivo local.
- **mget** archivos remotos
  Copia los archivos remotos en el sistema local.
- **mkdir** nombre de directorio
  Crea el nombre del directorio remoto.
- **mls** archivos remotos archivo local
  Escribe un listado abreviado de archivos remotos en el archivo local.
- **mode** (modo) [nombre del modo]
  Establece el modo de transferencia de archivo de ftp en el nombre del modo.
- **modtime** archivo remoto
  Muestra la fecha de la última modificación del archivo remoto.
- **mput** archivo local
  Copia el archivo local del sistema local al sistema remoto.
- **newer** nombre de archivo Elige el archivo sólo si la fecha de modificación del archivo remoto es más reciente que el archivo del sistema actual. Si el archivo no existe en el sistema actual, el archivo remoto es considerado más reciente. Por lo demás, este comando es idéntico a "get".
- **nlist** [directorio remoto] [archivo local] Escribe un listado abreviado del directorio remoto en el

archivo local. Si el directorio remoto no ha quedado especificado, se utiliza el directorio de trabajo actual.

- **open** servidor−huésped [número de puerto]
  Establece una conexión entre servidor−huésped, utilizando el número del puerto (si se especifica). Si el auto−login está permitido, ftp intenta entrar en el servidor.
- **put** archivo local [archivo remoto]
  Copia el archivo local en el archivo remoto.
- **pwd**
  Nos informa del nombre del directorio de trabajo actual.
- **quit**
  Un sinónimo de bye.
- **quote** argumentos
  Envía argumentos, al pie de la letra, al servidor.
- **recv** archivo remoto [archivo local]
  Un sinónimo de get.
- **reget** archivo remoto [archivo local]
  reget suele actuar como get, excepto que si existe un archivo local y es más pequeño que el archivo remoto, el archivo local es supuesto como copia parcialmente transferida del archivo remoto y la transferencia continua desde el punto aparente de fallo. Este comando es útil cuando se transfieren archivos muy grandes en redes que tienden a interrumpir conexiones.
- **rhelp** [nombre del comando] Ayuda a solicitud del servidor. Si el nombre del comando es especificado, lo suministra al servidor.
- **rstatus** [nombre de archivo]
  Sin argumentos, muestra el estado de la máquina remota. Si se especifica el nombre de archivo, muestra el estado del nombre de archivo en la máquina remota.
- **rename** desde remoto a remoto
  Renombra desde remoto, que puede ser un archivo o un directorio, a remoto.
- **rmdir** directorio remoto
  Elimina el directorio remoto. El directorio remoto tiene que estar vacío.
- **send** archivo local [archivo remoto]
  Un sinónimo de put.
- **size** archivo remoto
  Muestra el tamaño del archivo remoto.
- **status**
  Muestra el estado actual de ftp.
- **system**
  Muestra el tipo de sistema operativo que posee la máquina remota.
- **type** [nombre del tipo]
  Establece el tipo de transferencia de archivo de ftp para el nombre del tipo. Si el nombre del tipo no es especificado, escribe el tipo actual. ASCII y binario son los tipos tolerados actualmente.
- **?** [comando] Un sinónimo de help. Imprime la información de ayuda del comando especificado.

7. ¿COMO SE TRABAJA CON FTP ANONIMO?

Al conectarnos por ftp, nos encontramos con el problema de disponer de un identificador (login) y una clave de acceso (password) para cada máquina a la que nos conectemos. Esto se debe a protección que da el servidor a la información que posee para mantenerla fuera del alcance de intrusos. Por fortuna, los usuarios de Internet pueden realizar ftp sobre los servidores de la red configurados con el título "*anonymous ftp*" (ftp anónimo).

La comunicación por ftp anónimo es una forma de conectarse a los servidores de ficheros de Internet sin necesidad de poseer una palabra clave. Existen muchos servidores públicos que admiten conexiones ftp

anónimas llamados *ftp−sites*. Estos servidores se dedican a distribuir software de dominio público y shareware para cualquier tipo de ordenador o sistema operativo existente.

Para utilizar un ftp anónimo, el usuario debe introducir el identificador de usuario *anonymous* y como palabra clave de acceso se considera una muestra de cortesía introducir la dirección de correo electrónico auque también se suele introducir la palabra *guest* (huésped).

Para poder realizar conexiones ftp como las que hemos mencionado necesitamos disponer de soporte TCP/IP o SLIP o estar conectado a un servidor que nos ofrezca ftp. Para usuarios que no tengan contratado este servicio, existe el **FTPmail**, el cual, permite acceder mediante mensajes de correo electrónico a servidores de ficheros ftp. Utilizando un formato de dirección especial,se pueden enviar mensajes al servidor con comandos de texto para obtener directorios, moverse por el sistema de ficheros o transferir archivos. La respuesta del servidor vuelve también en forma de mensajes de correo. Este sistema es muy lento y dista mucho de la facilidad de una conexión en línea.

8. Transferencia de ficheros

En primer lugar debemos seleccionar el servidor FTP al cual conectarnos en la caja "profile name" de la ventana "session profile" y pulsar el botón connect.

Una vez conectados, debemos indicar el tipo de transferencia. ASCII o Binaria. El tipo de transferencia ASCII tiene sentido cuando los sistemas origen y destino interpretan de forma diferente el final de línea en los ficheros de texto. Por ejemplo entre un sistema Unix (que interpreta el final de una línea con el caracter ASCII 10 (*LF*), mientras que un sistema DOS/Windows interpreta el final de línea con el par de caractéres ASCII 13,10, (*CR,LF*). De no hacerlo así, tendremos el efecto escalera al recibir en un sistema Windows un fichero de texto Unix.

Para ficheros que no sean de texto (ejecutables, gráficos, sonidos, archivos comprimidos, etc), lo adecuado es seleccionar la opción de transferencia Binaria.

Una vez establecida la comunicación, el programa nos presenta una pantalla con dos ventanas, a la izquierda tenemos la ventana del sistema local, y a la derecha la del sistema remoto. En función de los derechos que tengamos, podremos enviar ficheros o sólo recibir (en acceso a un servidor FTP anónimo). Para realizar la transferencia se seleccionan los archivos a enviar o recibir y se pulsa el botón correspondiente (Flecha a la derecha para enviar, flecha a la izquierda para recibir).

Mientras dura la transferencia, aparece una ventana indicandonos el progreso de la copia. En el ejemplo vemos que estamos transfiriendo un fichero de 15 Mb del que ya ha sido descargado el

32% a una velocidad de 706 Kbytes por segundo. Se llevan 6 segundos de transferencia y se estima el tiempo restante en 14 segundos.

El resultado final de la transferencia (en este caso) fueron 15 segundos a 983 KBytes/segundo.

Hay que hacer notar que estas velocidades de transferencia son en nuestra Intranet con un ancho de banda de 10 Mbits/segundo. El acceso a servidores en Internet, lo haríamos en condiciones normales a 2 Kbytes por segundo con un módem a 28.8 Kbits/segundo y, aproximadamente a 5 Kbytes por segundo en una línea RDSI con un sólo canal activado.

Para finalizar la conexión con un servidor se debe pulsar la tecla ***close*** y ***exit*** para terminar la ejecución de WS_FTP.

9. ¿Cómo configurar un servidor de FTP anónimo seguro?

El objetivo de este documento es el de ayudar a los administradores a comprobar si su servidor de FTP está correctamente configurado y que está corriendo la version más moderna del demonio de FTP. Normalmente al instalar una versión del servidor FTP se suelen configurar muchas cosas automáticamente, tales como permisos de directorios, creación de usuarios ftp, etc. Éste documento sirve para repasar aquellas cosas que quizás se han hecho automáticamente y no siempre de la forma más correcta. Asimismo permite conocer cuales son los puntos débiles de una configuración de ftp.

10. El archivo /etc/ftpaccess

El archivo /etc/ftpaccess es el método principal para controlar la identidad y la cantidad de usuarios que acceden a su servidor. Cada línea del archivo define un atributo o fija su valor.

Estas ordenes controlan el acceso:

- **class <classname> <typelist> <addrglob>**

Define una clase de usuario con permisos para acceder a su servidor FTP.

- autogroup <groupname> <class>

Controla aún más el acceso de los usuarios anónimos, asignandoles automáticamente determinados permisos de grupo durenate el acceso.

- **deny <addrglob> <message_file>**

Permite negar el servicio a determinados hosts y mostrarles mensaje.

- guestgroup <groupname>

Usuarios reales que quiere que tengan privilegios FTP restrictivos.

- limit <class> <n> <times> <message_file>

Permite controlar el número de usuarios que accden al sistema mediante FTP según su clase y la hora del día.

- loginfails <n>

Permite fijar el número de intentos de acceso fallidos.

- private <yes ó no>

Permite que los usuarios compartan archivos mediante FTP sin necesidad de que se encuentren en un lugar público.

Estas ordenes controla la informacion que el servidor transmite a los clientes:

- banner <ruta>

Permite mostrar un mensaje en pantalla antes que el cliente introduzca su nombre y contraseña.

- email <direccion>

Permite especificar la dirección de correo electronico de la persona encargada del mantenimiento del sistema.

- message <ruta> <cuándo> <clase>

Permite crear mensajes especiales para enviarlos a los usuarios cuando accedan al sistema ó entren en un directorio concreto.

- readme <ruta> <cuándo> <clase>

Permite especificar las condiciones bajo las cuáles se avisará a los clientes de la última modificación de un determinado archivo.

Estas órdenes controlan la posibilidad de acceso:

- log commands <lista>

Cada uno de los comandos ejecutados por los clientes es registrado en el archivo correspondiente.

- log transfers <lista> <direcciones>

Permite mantener un registro sólo de los archivos transferidos por los clientes.

Las siguientes ordenes son de diferentes tipos:

- alias <cadena> <directorio>

Permite definir alias de directorios para sus clientes FTP.

- cdpath <directorio>

Similarmente a path.

- compress <yes ó no> <classglob>

Permite que los usuarios compriman ó descompriman un archivo antes de transmitirlo.

- tar <yes ó no> <classglob>

Es idéntica a la orden anterior.

- shutdown <ruta>

Indica al servidor que debe buscar un archivo de forma periódica para saber si el servidor se va a apagar.

Los permisos se fijan utilizando las siguientes ordenes:

- chmod <yes ó no> <lista>

Determina si un cliente tiene permiso para modificar los permisos de los archivos del servidor.

- delete <yes ó no> <lista>

Determina si un cliente puede eliminararchivos que residen en el servidor.

- overwrite <yes ó no> <lista>

Controla si los usuarios pueden sustituir archivos ya existentes en el servidor.

- rename <yes ó no> <lista>

Determina si los usuarios pueden renombrar archivos.

- umask <yes ó no> <lista>

Determina si los clientes pueden modificar los permisos que tienen asignados por defecto.

- passwd−check <rigurosidad> <obligacion>

Es de buena educación utilizar su dirección de correo electrónico cómo contraseña cuando se accede a un servidor FTP anónimo. Esta orden permite determinar cuán estricto quiere ser con la cadena proporcionada como dirección de correo electrónico de un usuario anónimo.

- path−filter <lista> <mensaje> <exp_permitida> <exp_no_permitida>

Si permite que los usuarios transfieran archivos a su servidor mediante FTP, path−filter permite determinar que archivos pueden transferir y cuáles no.

- upload <directorio> <dirglob> <yes ó no> <propietario> <grupo>

<modo> <mkdir>

Especifica los permisos que tiene el cliente para transferir archivos a determinados directorios, así cómo los

permisos que tendrán los archivos una vez que hayan sido colocados en dichos directorios.

<lista>, es una lista separada por comas de los tipos de usuarios a los que afecta la orden. Los posibles usuarios son anonymous, guest, y real.

11. El archivo /etc/ftpconversions

El formato del archivo /etc/ftpconversions es:

<1>:<2>:<3>:<4>:<5>:<6>:<7>:<8>

donde <1> es el prefijo, <2> el sufijo, <3> un prefijo adicional, <4> un sufijo adicional, <5> es la orden externa que se ejecutará para realizar la conversion, <6> el tipo de archivo, <7> la información de las opciones utilizadas al acceder y <8> una descripcion de la acción.

¿Confundido? No tiene por qué. En realidad cada una de estas opciones es bastante simple. En la siguiente sección las veremos de una en una.

El prefijo

El prefijo es la cadena, situada al principio del nombre de un archivo, que debe ser eliminada cuando se trae un nuevo archivo. Por ejemplo, si quiere que se realice una acción especial en todos los archivos que comiencen con *discography.*, en la que dicho prefijo se deberá eliminar después de realizar la acción, deberá especificar *.discography.orb* y un cliente ejecuta la orden get orb, el servidor ejecuta la orden opcional sobre dicho archivo y transfiere los resultados al cliente.

El sufijo

El sufijo es la cadena, situada al final del nombre del archivo, que deberá ser eliminada cuando se transfiera el archivo. El sufijo suele utilizarse para eliminar la extensión .gz de un archivo que está siendo descomprimido, antes de ser transferidode nuevo al cliente.

El prefijo adicional

El prefijo adicional es la cadena que se inserta antes del nombre del archivo cuando se transfiere un archivo desde ó hacia el servidor. Por ejemplo, podría introducir la cadena mayúsculas en todos los archivos que, al ser transferidos desde el servidor, conviertan su nombre a mayúsculas.

El sufijo adicional

El sufijo adicional es la cadena que se añade a un nombre de archivo cuando se ha realizado alguna operación con él. Este tipo de sufijo suele utilizarse cuando el cliente ejecuta la orden get *archivogrande.gz* y, en realidad, el nombre del archivo tan sólo es *archivogrande*, en este caso, el servidor comprime el archivo usando gzip y luego realiza la transferencia.

La orden externa

El componente más importante de cada línea es la orden externa. Esta entrada especifica el programa que debe ejecutarse cuando se transfiere un archivo desde ó hacia un servidor. Mientras se transfiere el archivo, es filtrado por el programa, desde donde los archivos enviados al cliente van a la salida estándar. Por ejemplo, si quiere descomprimir con gzip los archivos que se están transfiriendo, la entrada debería parecerse a ésta:

gzip −dc −%s

La cadena *%s* de ésta línea indica al servidor que debe sustituirla por el nombre del archivo solicitado por el usuario.

El tipo de archivo

El campo dedicado al tipo de archivo en /etc/ftpconversions incluye una lista con los posibles tipos de archivos sobre los que puede actuar, cada uno de ellos separado por un símbolo |. Los tres tipos de archivos que son reconocidos son T_REG, T_ASCII y T_DIR, que representan archivos normales, archivos ASCII y directorios, respectivamente. Un ejemplo de ésta tabla sería T_REG | T_ASCII.

Opciones

El campo de opciones de /etc/ftpconversions es parecido al campo dedicado al tipo de archivo, ya que está formado por una lista de nombres separados por el símbolo |. Los tres tipos de opciones que se pueden utilizar son O_COMPRESS, O_UNCOMPRESS y O_TAR, que se especifican si la orden comprime archivos, los descomprime o utiliza la orden tar. Una entrada habitual sería O_COMPRESS | O_TAR, que indica que el archivo será comprimido y después se ejecutará tar.

La descripción

El último parámetro de /etc/ftpconversions, la descripción de la conversión, es una entrada de formato libre en la que puede describir el tipo de conversión que se está realizando.

12. El archivo /etc/ftphosts

El archivo /etc/ftphosts establece reglas para cada usuario en las que se define si podrá acceder desde determinados host ó si se le negará el acceso cuando intenten acceder desde un host diferente. Cada línea del archivo puede incluir una de éstas ordenes:

- Allow <nombre_usuario> <dirglob>

Permite que el usuario especificado acceda mediante FTP desde las direcciones incluidas explícitamente en el parámetro <dirglob>.

- Deny <nombre_usuario> <dirglob>

Niega al usuario especificado el acceso desde cualquiera de la direcciones incluidas en <dirglob>.

13. El archivo /var/log/xferlog

A pesar de que /var/log/xferlog no es un archivo de configuración, es bastante importante. En él se guardan todos los registros generados por el servidor FTP. Cada línea del registro consta de los siguientes elementos:

- **current−time: la hora actual en formato DDD MMM dd hh:mm:ss YYYY, done DDD es el día de la semana, MMM el mes, dd el día del mes, hh:mm:ss la hora en formato militar y YYYY el año.**

- transfer−time: el tiempo total necesario para transferir el archivo en segundos.
- remote−host: nombre del host del cliente que inició la transferencia.

- file−size: el tamaño del archivo transferido.
- filename: el nombre del archivo transferido.
- transfer−type: el tipo de transferencia realizado, utilizando *a* para una transferencia ASCII y *b* para una transferencia binaria.
- special−action−flag: una lista de las acciones realizadas por el servidor en ese archivo, en la que C significa que ha sido comprimido, T que se utilizó con tar y − que no se realizó acción alguna.
- direction: una señal que indica si el archivo entraba ó salía, representado por *o* ó por *i*, respectivamente.
- acces−mode: el tipo de usuario que realizó la acción, siendo *a* anónimo, *g* guest y *r* real.
- username: el nombre de usuario local, si el usuario era del tipo real.
- service−name: el nombre del servicio llamado (casi siempre FTP).
- authentication−name: el tipo de autentificación utilizado: 0 indica que no se realizó ninguna (usuario anónimo) y 1 que el usuario fue validado por el Protocolo de Autentificación del Servidor RFC 931.
- authenticated−user−id: el nombre de usuario usado para autentificar la transferencia.

14. Problemas de Seguridad en servidores de FTP

Los servidores de FTP anónimo están sometidos a abusos de entre los que destacan principalmente dos:

- Distribución de software o material con copyright a través del servidor.
- Fallo de seguridad en el servidor debida a una mala configuración del servidor de FTP.

Los archivos anónimos se pueden conseguir de muchas formas. La mayoria de las veces es a través de servidores FTP anónimos pero no hay que olvidar otros medios como FSP o NFS. Algunos servidores permiten la existencia de zonas con permisos de escritura (directorios *incoming*) para que los usuarios puedan dejar archivos. Pero si los usuarios anónimos tienen permiso para leer de ese directorio la probabilidad de abuso por parte de dichos usuarios aumenta.

Ciertos usuarios consiguen y distribuyen listas (listas *warez*) que enumeran sitios con vulnerabilidad y la información que contienen. Éstas listas suelen incluir los nombres de directorios con permisos de escritura y la localización de software pirata, así como archivos de password u otra importante información.

Estos usuarios se aprovechan de la poca importancia que dan los administradores a que este tipo de abuso exista, o bien de la confianza que tienen dichos administradores en la configuración de su servidor.

A parte del problema del pirateo informático, existe el riesgo de que un mal configurado servidor puede proporcionar a alguién la capacidad de ejecutar procesos con UID (identificador de usuario) del demonio de FTP.

15. Descripción General de cómo configurar un servidor FTP anónimo.

15.1. Creación usuario ftp.

Crear un usuario ftp en el archivo */etc/passwd* cuyo directorio home sea *~ftp*. Éste será el directorio raíz a partir del cual los usuarios de ftp podrán ver los archivos. El crear este usuario permite el acceso anónimo al servidor. Asociar al usuario ftp un grupo no importante (sin especiales privilegios), así como un password y una shell incorrectos. Ejemplo:

ftp:*:400:400:Anonymous FTP:/home/ftp:/bin/true

Un password incorrecto es una sucesión de carácteres que no pueden representar la encriptación de ningún password, como por ejemplo 'x' o '*'. Una shell incorrecta puede ser por ejemplo un simple script que muestre

por pantalla que el usuario anónimo de ftp no puede conectarse a través de otros servicios como telnet o login. Conviene añadir la shell al archivo /etc/shells. Un ejemplo de este tipo de shell puede ser:

15.2. Permisos y contenido de los directorios.

La norma más importante a seguir a la hora de configurar los permisos de los directorios del servidor de ftp es que el directorio raíz (*~ftp*) y sus subdirectorios no deben ser poseidos por el usuario ftp. Éste es el problema de configuración más corriente.

Si no quiere perder mucho tiempo en este capítulo de los permisos le recomendamos que haga que el directorio raíz y los subdirectorios del servidor sean propiedad de **root** y del grupo **system**, y que sólo root tenga permiso de escritura. Un ejemplo de posible directorio del servidor de FTP anónimo sería:

drwxr−xr−x 7 root system 512 Jan 10 11:30 ./

drwxr−xr−x 25 root system 512 Jan 10 11:30 ../

drwxr−xr−x 2 root system 512 Jan 10 11:30 bin/

drwxr−xr−x 2 root system 512 Jan 10 11:30 etc/

drwxr−xr−x 10 root system 512 Jan 10 11:30 pub/

Si hubieramos dado la propiedad de los directorios al usuario ftp, dejandole además permiso de escritura, un intruso podría ser capaz de añadir archivos como *.rhosts* o modificar otros archivos.

Los archivos y librerías, especialmente los usados por el demonio de FTP y los que se encuentran en los directorios *~ftp/bin* y *~ftp/etc* deben tener las mismas protecciones que estos directorios.

Hasta ahora se ha hecho una descripción somera de los permisos. Si desea afinar más los permisos reduciendolos al máximo siga los consejos que se van a exponer a continuación.

El directorio *~ftp* debe poseerlo **root** (No ftp. Esto es un agujero de seguridad) y el mismo grupo al que pertenece ftp. El permiso debe ser 555 (read, nowrite, execute).

El directorio *~ftp/bin* y cualquier archivo incluido en él debe tener los permisos 111(noread, nowrite, execute), pertencer a root y a un grupo de root, por ejemplo whell. Además debe contener el programa *ls* para que de esta forma un usuario de FTP pueda listar los archivos.

El directorio *~ftp/etc* debe tener los mismos permisos que *~ftp/bin*. Hay que incluir en *~ftp/etc* una copia de los archivos */etc/passwd* y */etc/group* no sin antes eliminar de ellos todo lo necesario para que sólo incluyan las lineas correspondientes a los usuarios *root, daemon, uucp,* y *ftp* en passwd y al grupo de ftp en group. No olvidar eliminar todos los passwords de estos archivos sustituyendolos por un * (de esta forma evitamos que un usuario tenga acceso a password encriptados susceptibles de ser atacados por un programa craqueador) y el resto de informacion como se ve en el ejemplo:

root:*:0:0:Ftp maintainer::

ftp:*:400:400: Anonymous ftp::

De todas formas estos archivos no son necesarios y su ausencia sólo se notará en que al hacer un listado de archivos a traves de ftp éste no mostrará los nombres de usuario ni de grupo de los archivos y en su lugar

mostrará su número equivalente.

El directorio *~ftp/pub* debe ser propiedad del administrador y del grupo de ftp con permisos 555. Se puede poner además la opcion set−group−id (2555) para que los nuevos archivos que se creen tengan al mismo grupo como propietario. Los archivos que se incluyan tambien con permiso 555.

Debido a que las nuevas versiones de los servidores de ftp permiten ejecutar comandos como *chmod*, es necesario configurarlos para evitar que se puedan cambiar los permisos. Las opciones recomendables son:

all the following default to "yes" for everybody

delete no guest,anonymous # delete permission?

overwrite no guest,anonymous # overwrite permission?

rename no guest,anonymous # rename permission?

chmod no anonymous # chmod permission?

umask no anonymous # umask permission?

15.3. ¿Cómo permitir la escritura en un servidor anónimo?.

Si se quiere que usuarios anónimos puedan dejar archivos, hay que crear un directorio para ello, por ejemplo *~ftp/pub/incoming* poseido por *root* y con permisos 733. Además ejecutar el comando: *chmod +t ~ftp/pub/incoming*

De esta forma aunque normalmente el servidor de ftp no permitirá sobreescribir en un archivo ya existente, un usuario del sistema aunque quisiera no podrá. Por tanto el directorio tendrá los permisos 1733.

Con *wuftpd* se puede configurar el servidor para que los archivos que se creen tengan como permisos 600 y sean poseidos por root o cualquier otro usuario. Para evitar abusos en el uso de este directorio, sobre todo cuando su uso se aleja del fin deseado, se suele quitar el permiso de lectura de dicho directorio.

A veces lo que los usuarios anónimos hacen es crear directorios ocultos dentro de ese directorio para evitar ser descubiertos. Pero esto puede evitarse con modernos servidores como *wuftpd* que restringen el conjunto de caracteres permitidos para nombres de ficheros y directorios. Las lineas a incorporar en el archivo de configuración son:

specify the upload directory information

upload /var/spool/ftp * no

upload /var/spool/ftp /incoming yes ftp staff 0600 nodirs

path filters

path−filter anonymous /etc/msgs/pathmsg ^[−A−Za−z0−9_\.]*$ ^\. ^−

path−filter guest /etc/msgs/pathmsg ^[−A−Za−z0−9_\.]*$ ^\. ^−

Aún así si se quieren evitar abusos existen las siguientes soluciones:

- Utilizar una versión del demonio de FTP modificada expresamente para prevenir el uso no deseable de las zonas con permiso de escritura. Se sugiere:

    - Implementar una política en la que un archivo introducido en la zona de escribible no puede ser accedida por los demás usuarios anónimos hasta que el administrador del sistema examine el archivo y lo ponga en un directorio público.
    - Limitar la cantidad de datos transmisible en una sesión.
    - Limitar la cantidad total de datos transferibles en función del espacio libre del disco del sistema.
    - Aumentar la información grabada en los archivos log.

- Utilizar un método basado en ocultar los archivos en los que se tiene permiso de escritura para que sólo los usuarios legítimos puedan acceder.

Para ello, proteger el directorio *~ftp/incoming* dandole sólo permito de ejecución al usuario anónimo (*chmod 751 ~ftp/incoming*). Esto permitirá a los usuarios anónimos cambiar de directorio (*cd*), pero no ver los contenidos del mismo:

drwxr−x−−x 4 root system 512 Jun 11 13:29 incoming/

A continuación, crear directorios en *~ftp/incoming* usando nombres sólo conocidos por los usuarios locales o por aquellos usuarios anónimos a los que se les quiere dar permiso de escritura. Hay que tener, por tanto, el mismo cuidado que cuando se eligen passwords a la hora de buscar nombres a los directorios. Por ejemplo:

drwxr−x−wx 10 root system 512 Jun 11 13:54 aAbBcC/

drwxr−x−wx 10 root system 512 Jun 11 13:54 0a1b2c/

De esta forma se previene que un usuario anónimo casual puede escribir ficheros en el sistema. Hay que señalar que la seguridad de este método se basa en la protección de los nombres de los directorios, por lo que en el momento que estos se hagan públicos desaparece la medida de seguridad.

- Dedicar al directorio *~ftp/incoming* una partición de disco o incluso un disco exclusivamente. De esta forma se limita el tamaño de información que puede ser introducida, evitandose el posible ataque de intentar llenar el disco del servidor.

El administrador debería estar pendiente del uso de este directorio, por si se está desperdiciando en exceso memoria.

15.4. Extensiones del servicio de FTP.

Si quieres soportar algunas extensiones de ftp como comprimir/descomprimir o crear archivos tar de jerarquias de directorios es necesario poner en el directorio *~ftp/bin* los correspondientes programas y editar el fichero */etc/ftpconversions* para que contenga las siguientes lineas:

#strip prefix:strip postfix:addon prefix:addon postfix:external command:

#types:options:description

:.Z: : :/bin/compress −d −c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS

:−z: : :/bin/compress −d −c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS

: : :.Z:/bin/compress −c %s:T_REG:O_COMPRESS:COMPRESS

: : :.tar:/bin/tar cf − %s:T_REG|T_DIR:O_TAR:TAR

: : :.tar.Z:/bin/pipe /bin/tar cf − %s | /bin/compress
−c:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS

: : :.tar:/bin/gtar −c −f − %s:T_REG|T_DIR:O_TAR:TAR

: : :.tar.Z:/bin/gtar −c −Z −f − %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS

: : :.tar.gz:/bin/gtar −c −z −f − %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP

Las anteriores lineas hacen referencia a un programa llamado *pipe* que permite encadenar la ejecución de dos programas. Es utilizado para encadenar los programas tar y compress y debe ser incluido en el directorio *~/ftp/bin*.

15.5. Detalles de seguridad.

Para conseguir una mayor seguridad, otras cosas a hacer son:

touch ~ftp/.rhosts
touch ~ftp/.forward
chmod 400 ~ftp/.rhosts
chmod 400 ~ftp/.forward

de esta forma nos aseguramos que estos archivos tienen longitud cero (si es que antes no existían, si éste es el caso, borrarlos previamente) y son poseidos por root.

Si se montan discos de otras m'aquinas sobre la jerarquía de ftp, montarlo como de sólo lectura. Por ejemplo una entrada del archivo */etc/fstab* podría ser:

other:/u1/linux /home/ftp/pub/linux nfs ro,noquota,nosuid,intr,bg 1 0

16. ¿Cómo saber si un servidor es seguro?

Pasos a seguir por un administrador para asegurarse que la seguridad del sistema no está siendo comprometida:

1.− Asegurarse que el servidor de ftp no tiene el comando SITE EXEC haciendo telnet al puerto 21 y tecleando SITE EXEC. Si lo tuviera asegurarse que es la version más moderna (es decir Wu−FTP 2.4). En antiguas versiones esto permitía ganar a cualquiera una shell a través de ese puerto.

2.− Comprobar que nadie puede conectarse ni crear archivos ni directorios en el directorio principal. Si cualquiera puede conectarse como anonymous FTP y crear archivos como .rhosts y .forward, se garantiza acceso instantaneo a cualquier intruso.

3.− Asegurarse que el directorio principal no esta poseido por ftp. Si lo estuviera un intruso podria ejecutar SITE CHMOD 777 en el directorio principal y luego en los archivos para darle acceso. El comando SITE CHMOD debería ser eliminado porque usuarios anónimos no necesitan ningun privilegio extra.

4.− Mirar que ningún archivo ni directorio está poseido por ftp. Si lo está un intruso podráa reemplazarlos con

su propia version troyana.

5.− Hay varios errores en antiguos demonios por lo que es importante asegurarse que se tiene la version más moderna.

17. Políticas de seguridad.

1.− Comprobar la no existencia de archivos o directorios inusuales en el directorio *incoming* especialmente busca por archivos ocultos que normalmente no son mostrados mediante "*ls*".

2.− Revisar periódicamente los archivos log de ftp buscando actividades inusuales del servidor, como gran cantidad de conexiones, identificadas como "puts" y "gets", por ejemplo, en un breve intervalo de tiempo. Esto puede mostrar un uso del servidor como intercambidor de software pirateado, que a la larga puede provocar: que legítimos usuarios no puedan acceder al sistema por haber demasiadas conexiones, la caida del sistema o el consumo de espacio de disco en el sistema.

Conviene por tanto desarrollar herramientas de análisis de los archivos logs, buscando "STOR" y "RETR" sesiones, adaptadas a la actividad del servidor.

3.− FSP es un servidor de archivos anónimo que es similar a FTP. Está basado en UDP y a menudo utuliza el puerto privilegiado 21. Sin embargo, existen casos en que usuarios o intrusos han establecido su propio servicio de FSP sobre un puerto UDP no privilegiado. Aunque FSP no es en sí un problema, es susceptible de sufrir los mismos abusos que FTP.

Por tanto, si no va a ofrecer un servico FSP, examine el sistema mirando los servicios que se ofrecen a través del puerto 21. El problema aparece cuando un usuario ofrece un servicio FTP o FSP no autorizado en un puerto no privilegiado, debido a la dificultad de detección, haciendose necesario un análisis de tráfico de los puertos.

4.− Asegurese de que no se ha producido ninguna modificación de algún fichero existente. Muestre especial interes en el propio demonio de ftp y en archivos con directo impacto en la seguridad del sistema como *~ftp/.rhosts*. Se conocen casos de servidores en los que se reemplazó el demonio de ftp por una versión del caballo de Troya.

5.− Utilice herramientas como *Tripwire* que permiten la comprobación de la integridad de sus archivos.

18. ¿Qué hacer si nuestro servidor está siendo usado ilegalmente?

1.− Si se detecta el uso del servidor como distribuidor de software pirateado, debe revisar los directorios y archivos creados como resultado de dicho abuso, y tome las medidas oportunas de acuerdo a las normas de su organización.

2.− Si descubre entre el material descubierto cualquier referencia a otros sitios, como por ejemplo listas warez, haga lo siguiente:

- Determine desde dónde se produjo el acceso no autorizado, pues lo más probable es que la seguridad de estos sitios esté siendo igualmente comprometida.
- Revise los contenidos de los archivos o directorios en busca de posibles referencias a cuentas o passwords de otros servidores.
- Notifique cualquier descubrimiento a la organización correspondiente.

19. ¿Dónde buscar servidores de FTP?. La solución : Archie

Para buscar servidores FTP es bueno recurrir a servidores archie para que efectuen la búsqueda. Si quieres que tu servidor anónimo se añadido a la lista de buscadores archie manda un mail a archie−updates@bunyip.com

Servidores Archie más importantes:

archie.ac.il 132.65.20.254 (Israel server)

archie.ans.net 147.225.1.10 (ANS server, NY (USA))

archie.au 139.130.4.6 (Australian Server)

archie.doc.ic.ac.uk 146.169.11.3 (United Kingdom Server)

archie.edvz.uni−linz.ac.at 140.78.3.8 (Austrian Server)

archie.funet.fi 128.214.6.102 (Finnish Server)

archie.internic.net 198.49.45.10 (AT&T server, NY (USA))

archie.kr 128.134.1.1 (Korean Server)

archie.kuis.kyoto−u.ac.jp 130.54.20.1 (Japanese Server)

archie.luth.se 130.240.18.4 (Swedish Server)

archie.ncu.edu.tw 140.115.19.24 (Taiwanese server)

archie.nz 130.195.9.4 (New Zealand server)

archie.rediris.es 130.206.1.2 (Spanish Server)

archie.rutgers.edu 128.6.18.15 (Rutgers University (USA))

archie.sogang.ac.kr 163.239.1.11 (Korean Server)

archie.sura.net 128.167.254.195 (SURAnet server MD (USA))

archie.sura.net(1526) 128.167.254.195 (SURAnet alt. MD (USA))

archie.switch.ch 130.59.1.40 (Swiss Server)

archie.th−darmstadt.de 130.83.22.60 (German Server)

archie.unipi.it 131.114.21.10 (Italian Server)

archie.univie.ac.at 131.130.1.23 (Austrian Server)

archie.unl.edu 129.93.1.14 (U. of Nebraska, Lincoln (USA))

archie.univ−rennes1.fr (French Server)

archie.uqam.ca 132.208.250.10 (Canadian Server)

archie.wide.ad.jp 133.4.3.6 (Japanese Server)

20. Como actualizar nuestras páginas web.

Lo primero que tendremos que hacer es solicitar este servicio en las oficinas de nuestro servidor, pues aunque no se cobra por él, no se asigna por defecto al darnos de alta.

Una vez que ya lo tenemos disponible, para actualizar nuestras páginas tendremos que volcarlas en el servidor. Para ello deberemos configurar nuestro programa de ftp de una forma determinada. Aquí vamos a poner de ejemplo el software *WS_FTP* pues es uno de los mas difundidos, pero la configuración será identica para cualquier otro programa.

20.1. ¿Que datos necesito conocer?

1. El nombre del servidor
• Nuestro login ante el equipo • Nuestra password
• El directorio de trabajo

20.2. ¿Como pongo estos datos en mi programa?

Pantalla de configuración del *WS_FTP*	Es importante pinchar el botón ***New*** antes de introducir los datos y el botón ***Save*** despues para conservarlos en futuras conexiones. Una vez configurado, pulsando el botón ***Ok*** nos conectaremos sin ningún problema.

20.3. Ya estoy conectado pero no entiendo lo que aparece en pantalla.

Aspecto normal tras establecerse la conexión.	La pantalla principal se divide en dos partes claramente diferenciadas: *Local System* (nuestro equipo) *Remote System* (el servidor) En cada una de ellas la parte superior se refiere a directorios y la inferior a ficheros.

20.4. ¿Como puedo enviar mis páginas al servidor?

Lo primero que tenemos que saber es que se transmiten de forma distinta los ficheros de texto (.html, .txt, ...) que los binarios (.gif, .jpg, ...). Esto se le indica marcando en la parte inferior de la pantalla ***ASCII*** o ***Binary*** según sea el caso.

Lo siguiente que tenemos que hacer es seleccionar los ficheros a enviar. Para esto *pincharemos* sobre ellos de manera que queden marcados.

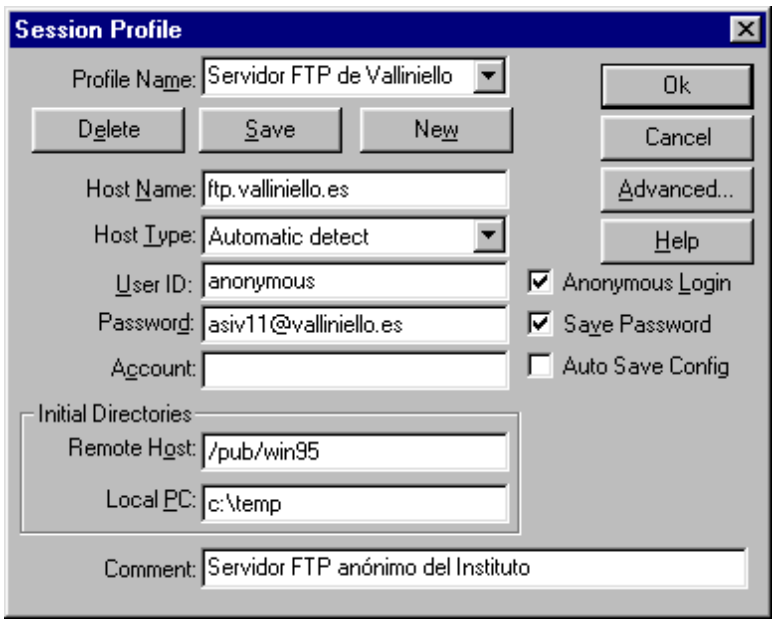
Aspecto de la pantalla tras enviar traba1.html, traba2.html, traba3.html y traba4.html.	Despues de enviar cualquier fichero al servidor, este tiene que verse en pantalla.

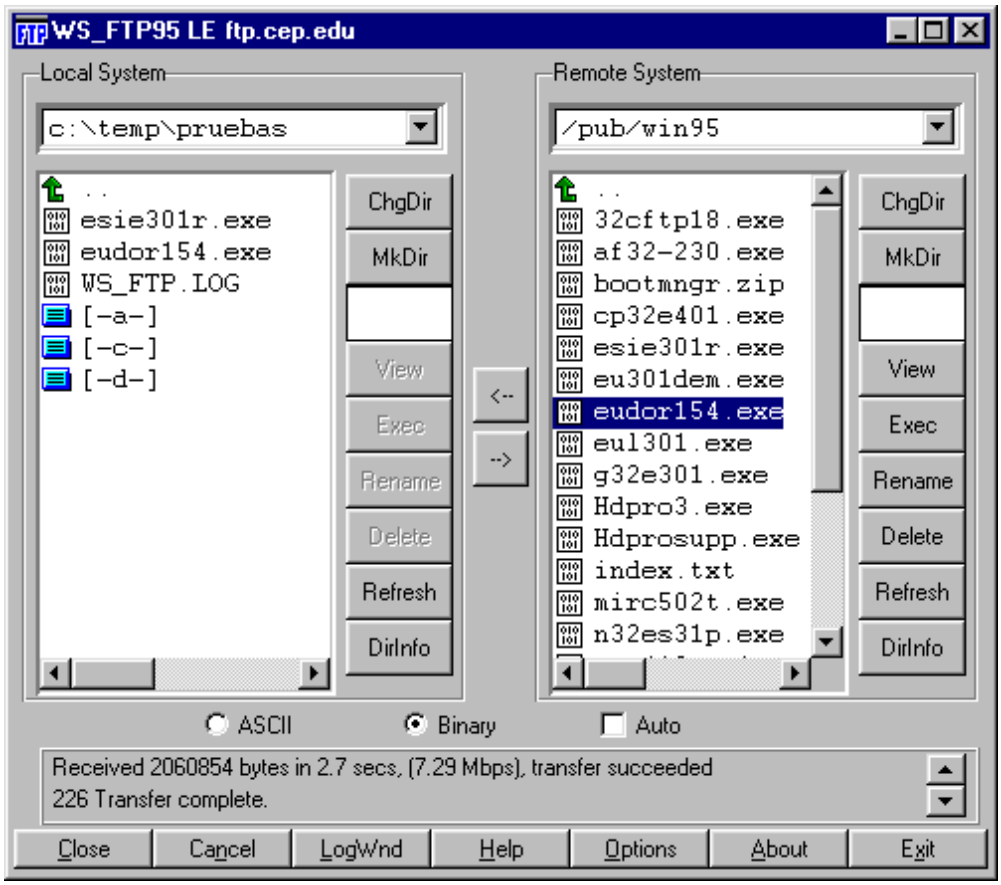
20.5. ¿Como le indico al servidor cual es mi página inicial?

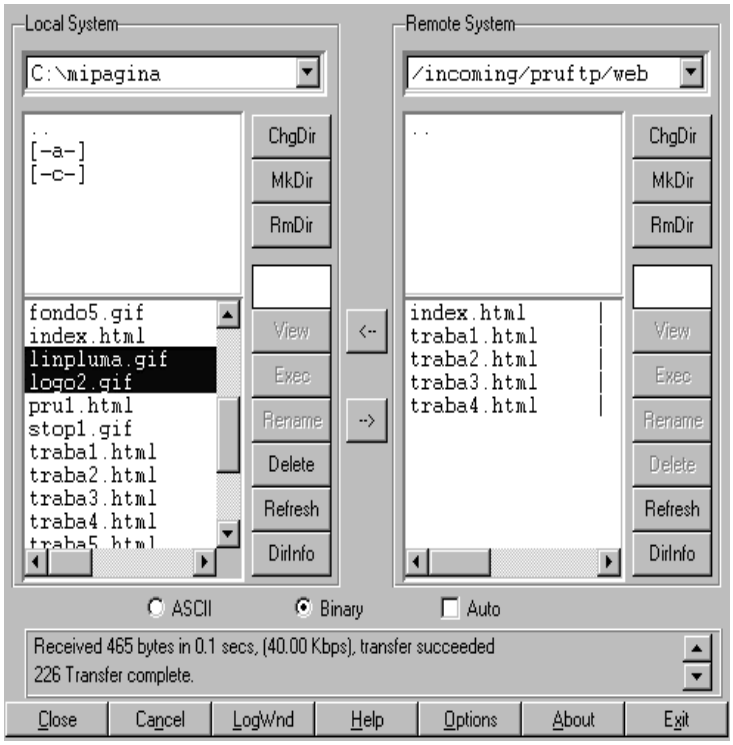
Nuestro servidor web esta configurado para poner como página principal de cada usuario la que se llama ***index.html***. En caso de que no exista, mostraria los ficheros del directorio ***web*** en forma de lista.

1

Página 33 de 33




WS_FTP95 LE ftp.cep.edu
Local System
c:\temp\pruebas
..
esie301r.exe
eudor154.exe
WS_FTP.LOG
[-a-]
[-c-]
[-d-]
ChgDir
MkDir
View
Exec
Rename
Delete
Refresh
DirInfo
<--
-->
Remote System
/pub/win95
..
32cftp18.exe
af32-230.exe
bootmngr.zip
cp32e401.exe
esie301r.exe
eu301dem.exe
eudor154.exe
eul301.exe
g32e301.exe
Hdpro3.exe
Hdprosupp.exe
index.txt
mirc502t.exe
n32es31p.exe
ASCII
Binary
Auto
Received 2060854 bytes in 2.7 secs, (7.29 Mbps), transfer succeeded
226 Transfer complete.
Close
Cancel
LogWnd
Help
Options
About
Exit


Local System
C:\mipagina
..
[-a-]
[-c-]
fondo5.gif
index.html
linpluma.gif
logo2.gif
pru1.html
stop1.gif
traba1.html
traba2.html
traba3.html
traba4.html
traba5.html
ChgDir
MkDir
RmDir
View
Exec
Rename
Delete
Refresh
DirInfo
<--
-->
Remote System
/incoming/pruftp/web
..
index.html
traba1.html
traba2.html
traba3.html
traba4.html
ASCII
Binary
Auto
Received 465 bytes in 0.1 secs, (40.00 Kbps), transfer succeeded
226 Transfer complete.
Close
Cancel
LogWnd
Help
Options
About
Exit