Usando Samba

Robert Eckstein, David Collier-Brown, Peter Kelly
Primera Edición, Noviembre 1999
1-56592-449-5, 416 págs.
Traducción: PROYECTO S.O.B.L.

1 de octubre de 2001

Índice general

Capítulo 1

Aprendiendo Samba

Si eres el típico administrador de un sistema, entonces sabes lo que significa estar hasta arriba de trabajo. Tu rutina diaria está llena de interminables problemas con la compatibilidad del hardware, sobrecargas del sistema, problemas con las copias de seguridad, y un buen múmero de usuarios cabreados. Así que el hecho de añadir otro programa a la mezcla de herramientas que ya tienes que mantener puede sonar un poco a "más problemas". Sin embargo, si has tomado la determinación de reducir la complejidad de tu entorno de trabajo, así como la sobrecarga del mismo, Samba puede ser la herramienta que estabas esperando.

Si eres consciente de que estás teniendo problemas con tu red y estás seguro de que hay un método mejor, te aconsejamos que comiences a leer este libro. O, si has oído hablar de Samba y deseas ver qué es lo que puede hacer por ti, este es también el mejor lugar para empezar. Bien, comencemos a mostrarte el camino hacia el conocimiento de Samba y su potencial. Antes de empezar, puedes proporcionar servicios Unix a todas tus máquinas Windows -y todo sin tener que gastar toneladas de tiempo y dinero-. ¿Excitante, no? Estupendo, entonces comencemos.

1.1. ¿Qué es Samba?

Samba es una suite de aplicaciones Unix que habla el protocolo SMB (Server Message Block). Muchos sistemas operativos, incluídos Windows y OS/2, usan SMB para operaciones de red cliente-servidor. Mediante el soporte de este protocolo, Samba permite a los servidores Unix entrar en acción, comunicando con el mismo protocolo de red que los productos de Microsoft Windows. De este modo, una máquina Unix con Samba puede enmascararse como servidor en tu red Microsoft y ofrecer los siguientes servicios:

- Compartir uno o más sistemas de archivos.
- Compartir impresoras, instaladas tanto en el servidor como en los clientes.
- Ayudar a los clientes, con visualizador de Clientes de Red.
- Autentificar clientes logeándose contra un dominio Windows.
- Proporcionar o asistir con un servidor de resolución de nombres WINS.

Samba es la idea de Andrew Tridgell, quien actualmente lidera el equipo de desarrollo de Samba development desde su casa de Canberra, Australia. El proyecto nació en 1991 cuando Andrew creó un programa servidor de ficheros para su red local, que soportaba un raro protocolo DEC de Digital Pathworks. Aunque él no lo supo en ese momento, aquel protocolo más tarde se convertiría en SMB. Unos cuantos años después, él lo expandió como su servidor SMB particular y comenzó a distribuirlo como producto por Internet bajo el nombre de servidor SMB. Sin embargo, Andrew no pudo mantener ese nombre -ya pertenecía como nombre de producto de otra compañía-, así que intentó lo siguiente para buscarle un nuevo nombre desde Unix:

```
grep -i 's.*m.*b' /usr/dict/words
```

y la respuesta fue:

```
salmonberry samba sawtimber scramble
```

De ésta manera nació el nombre de Samba. Hoy, la suite Samba implica a un par de demonios que proporcionan recursos compartidos a clientes SMB sobre la red (las comparticiones son denominadas a veces también como servicios). Estos demonios son:

smbd Un demonio que permite compartición de archivos e impresoras sobre una red SMB y proporciona autentificación y autorización de acceso para clientes SMB.

nmbd Un demonio que busca a través del Windows Internet Name Service (WINS), y ayuda mediante un visualizador.

Samba se encuentra actualmente mantenido y es ampliado por un grupo de voluntarios bajo la supervisión activa de Andrew Tridgell. Al igual que el sistema operativo Linux, Samba es considerado por sus autores Open Source software (OSS), y es distribuido bajo la the GNU General Public License (GPL). Desde su concepción, el desarrollo de Samba ha sido patrocinado en parte por la Australian National University, donde Andrew Tridgell hizo su doctorado. En adición, algunas partes del desarrollo han sido patrocinadas por distribuidores independientes como Whistle and SGI. Es algo verdaderamente testimonial el que entidades tanto comerciales como no comerciales estén dispuestas a gastar dinero para dar soporte a un esfuerzo Open Source.

En el momento de la impresión de este libro, Andrew ha completado su trabajo de doctorado y ha pasado a formar parte de una compañía desarrolladora de Linux de San Francisco.

Microsoft también ha contribuido materialmente poniendo a disposición su definición de SMB y del Internet-savvy Common Internet File System (CIFS), como Public Request for Comments (RFC), y otros documentos estandar. El protocolo CIFS es el nuevo nombre de las futuras versiones del protocolo SMB que serán usadas en los productos Windows -los dos términos pueden ser usados aleatoriamente en éste libro-. De hecho, verás el protocolo escrito como "SMB/CIFS".

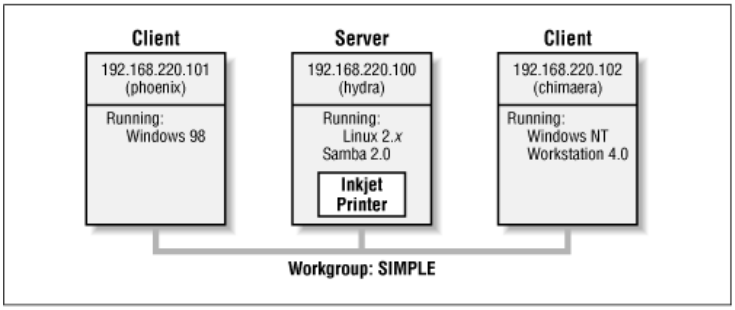
1.2. ¿Qué puede hacer Samba por mí?

Como explicamos antes, Samba puede ayudar a las máquinas Windows y Unix a coexistir en la misma red. Sin embargo, existen algunas razones específicas por las cuales podrías desear instalar un servidor Samba en tu red:

- No quieres pagar un servidor Windows NT para obtener las funcionalidades que este proporciona.
- Puedes querer proporcionar un área común para datos o directorios de usuarios en orden a realizar una transición desde un servidor NT hacia un Unix, o viceversa.
- Puede que desees compartir impresoras a entre clientes Windows y Unix.
- Puede que quieras acceder a ficheros NT desde un servidor Unix.

Veamos ahora a Samba en acción. Asumiremos que tenemos la siguiente configuración básica de red: un servidor Samba sobre una máquina Unix, al cual le asignaremos el nombre hydra, y un par de clientes Windows, a los cuales les asignaremos los nombres phoenix y chimaera, todos conectados vía red de área local (LAN). Asumamos también que hydra también tiene una impresora de inyección conectada a ella, lp, y una compartición de disco denominada network -ambos recursos podemos ofrecerlos a las otras dos máquinas-. Un gráfico de esta red se muestra en la Figura 1.

Figura 1.1: Una configuración de red simple con un servidor Samba.



En esta red, cada una de las computadoras comparten el mismo grupo de trabajo. Un Grupo de Trabajdo es simplemente una etiqueta de nombre de grupo que identifica a una determinada colección de ordenadores y sus recursos sobre una red SBM. Pueden existir varios grupos de trabajo sobre la red al mismo tiempo, pero para nuestro ejemplo sólo tendremos uno: el grupo de trabajo *SIMPLE*.

1.2.1. Compartiendo un Servicio de Disco.

Si todo está bien configurado, deberíamos poder ver al servidor Samba server, *hydra*, a través del visualizador de red (entorno de red) de la máquina Windows llamada *phoenix*. De hecho, la Figura 2 muestra el visualizador de red de la computadora *phoenix*, incluyendo a *hydra* y a cada una de las máquinas que residen en el grupo de trabajo *SIMPLE*. Advierte el icono *Entire Network* al principio de la lista. Como mencionamos antes, pueden existir más grupos de trabajo sobre una red SMB al mismo tiempo. Si un usuario hace click sobre ese icono, verá una lista de todos los grupos de trabajo que actualmente existen en la red.

Podemos entrar en el servidor *hydra* con un doble click sobre su icono. Esta acción provoca que se contacte con *hydra* y se le solicite una lista de sus recursos compartidos -La impresora y el disco- que proporciona la máquina. En nuestro caso, existe una impresora nominada **lp** y un disco compartido llamado **network** en el servidor, como lo

muestra la Figura 3. Advierte que la ventana muestra los nombres de las máquinas con letras mayúsculas/minúsculas (*Hydra*). Las mayúsculas son irrelevantes en los nombres de host (máquinas) así que puedes leer *hydra*, *Hydra*, y *HYDRA* como salida, pero todas se referirán al mismo sistema. Gracias a Samba, Windows 98 ve al server Unix como a un servidor SMB válido, y puede acceder a la carpeta **network** como si fuese una carpeta más del sistema.

Una característica popular de Windows 95/98/NT es que puedes mapear una letra de unidad hacia un direcrotio de la red usando la opción "Conectar a Unidad de Red" desde el explorador de Window. Una vez lo hayas hecho, tus aplicaciones podrán acceder a la carpeta a través de la red con una unidad de disco estándar. Una vez llegados a este punto, podrás lamacenar datos en ella, instalar y ejecutar programas, e incluso protegerla mediante contraseña contra accesos no deseados. Mira la Figura 4 para un ejemplo de mapeado de letra de unidad sobre un directorio de red.

Echa un vistazo a la ruta: entra en la caja de diálogo de la Figura 4. Una forma equivalente de representar un directorio en una máquina de la red es usando dos barras (backslashes), seguidas del nombre de la máquina de red, otra barra (backslash), y el directorio de red de la máquina, como se muestra a continuación:

```
\\máquina-de-red \directorio
```

Esto se conoce como notación UNC (Universal Naming Convention) en el mundo Windows. por ejemplo, la caja de diálogo en la Figura 4 representa el directorio de red del servidor hydra como:

```
\\HYDRA\network
```

Si esto te suena de algo, probablemente estarás pensando en uniform resource locators (URLs), que son las notaciones que usan los navegadores web como Netscape Navigator e Internet Explorer para resolver máquinas a través de Internet. Asegúrate de no confundirte: los navegadores web usan barras inclinadas a la derecha y no a la izquierda, y están precedidas por el nombre de protocolo de transferencia de datos a usar (p.ej., ftp, http) y dos puntos (:). En realidad, URLs y UNCs son dos cosas completamente distintas.

Una vez la unidad de red está configurada, Windows y sus programas la verán y podrán usar como si ese directorio de red fuese un dico más. Si tienes aplicaciones multiusuario, puedes instalarlas sobre la unidad de red. La Figura 5 muestra la unidad de red resultante como si fuera una unidad más en el cliente Windows 98. Advierte la tubería de enlace en el icono para la unidad "G:";esto indica que es una unidad de red, en lugar de una unidad física.

Desde nuestro cliente Windows NT Workstation, chimaera, Samba aparece de forma idéntica a como lo hace en el cliente Windows 98. La Figura 6 muestra la misma vista del servidor hydra desde el explorador de red del cliente Windows NT 4.0. Configurando la unidad de red usando la opción "Conectar a Unidad de Red" en Windows NT Workstation 4.0 obtendríamos el mismo resultado.

1.2.2. Compartiendo una Impresora.

Probablemente habrás notado que la impresora lp aparece en la lista de recursos compartidos de hydra en la Figura 3. Esto indica que el servidor Unix tiene una impresora que puede ser compartida con los clientes SMB del grupo de trabajo. Los datos

Figura 1.2: El directorio del visualizador de red (Entorno de Red).

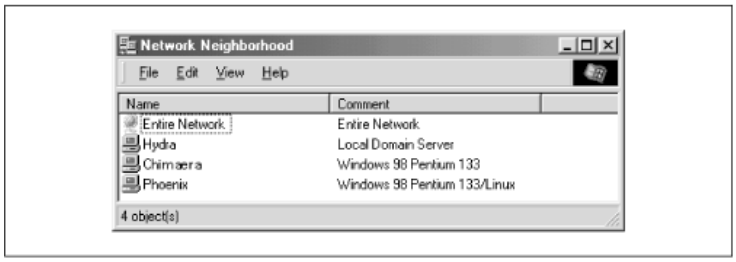


Figura 1.3: Recursos Compartidos disponibles en el servidor *hydra* vistos por *phoenix*.

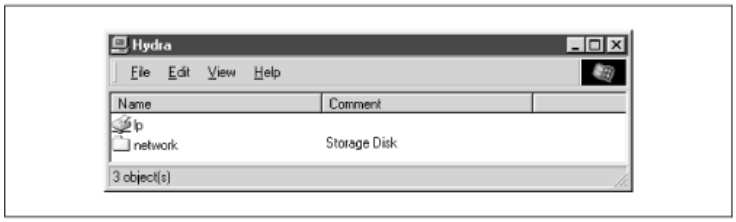


Figura 1.4: Mapeando una unidad de red con una letra de unidad Windows.

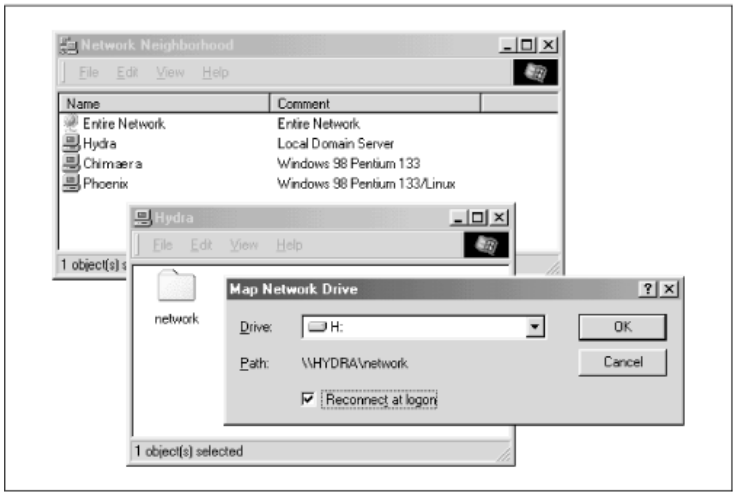
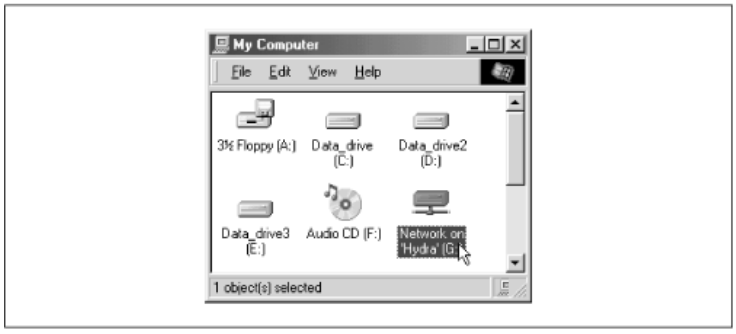


Figura 1.5: El directorio de red mapeado con letra de unidad de cliente.



enviados a la impresora desde cualquiera de los clientes será cola de impresión del servidor Unix e impresos en el orden de recepción.

La configuración de una impresora accesible a través de Samba en los clientes Windows es más sencilla que la compartición de unidades de disco. Haciendo un doble click en la impresora e identificando fabricante y modelo, puedes instalar un driver para esa impresora en el cliente Windows. Windows podrá entonces dar forma a cualquier información enviada a la impresora de red y acceder a ella como si fuese una impresora local (más adelante te mostraremos cómo hacerlo). La Figura 7 muestra la impresora de red resultante en la ventana de Impresoras de Windows 98. De nuevo, advierte la tubería bajo el icono dela impresora, que indica se trata de una impresora de red.

1.2.3. Viendo cosas desde la parte Unix

Como mencionamos antes, Samba aparece como un juego de programas demonios. Puedes verlos con los comandos Unix ps y netstat, puedes también leer cualesquiera mensajes que ellos generen a través de los ficheros de depuración de Unix syslog (dependiendo de cómo hayas configurado Samba), y puedes configurarlos desde un simple fichero de configuración: smb.conf. Además, si quieres hacerte una idea de qué hace cada uno de los demonios, Samba tiene un programa llamado smbstatus que te informará en linea. Así es como trabaja:

```
# smbstatus
Samba version 2.0.4
Service uid gid pid machine
----------------------------------------------
network davecb davecb 7470 phoenix (192.168.220.101) Sun May 16
network davecb davecb 7589 chimaera (192.168.220.102) Sun May 16
Locked files: Pid DenyMode R/W Oplock Name
----------------------------------------------
7589 DENY_NONE RDONLY EXCLUSIVE+BATCH /home/samba/quicken/inet/common/system/help.bmp
Sun May 16 21:23:40 1999 7470 DENY_WRITE RDONLY NONE /home/samba/word/office/findfast.exe
Sun May 16 20:51:08 1999 7589 DENY_WRITE RDONLY EXCLUSIVE+BATCH /home/samba/quicken/lfbmp70n.dll
Sun May 16 21:23:39 1999 7589 DENY_WRITE RDWR EXCLUSIVE+BATCH  /home/samba/quicken/inet/qdata/runtime.dat
Sun May 16 21:23:41 1999 7470 DENY_WRITE RDONLY EXCLUSIVE+BATCH /home/samba/word/office/osa.exe
Sun May 16 20:51:09 1999 7589 DENY_WRITE RDONLY NONE /home/samba/quicken/qversion.dll
Sun May 16 21:20:33 1999 7470 DENY_WRITE RDONLY NONE /home/samba/quicken/qversion.dll
Sun May 16 20:51:11 1999
Share mode memory usage (bytes):
1043432(99%) free + 4312(0%) used + 832(0%) overhead = 1048576(100%) total
```

El informe de status de Samba que ves arriba proporciona tres grupos de datos, cada uno de ellos dividido en secciones separadas. La primera sección te dice qué sistemas han conectado al servidor Samba, identificando a cada cliente por su nombre de máquina (phoenix y chimaera) y dirección IP. La segunda sección reporta el nombre y status de los ficheros que están actualmente en uso en una compartición del servidor, incluyendo el status lectura/estritura y los bloqueos de los ficheros. Finalmente, Samba reporta la cantidad de memoria que actualmente está dedicada para los recursos que administra, incluyendo la cantidad activamente usada por los recursos más la restante de overhead. (Advierte que esta no es la misma que la cantidad total de memoria que los procesos smbd o nmbd están usando).

No te preocupes si no entiendes estas estadísticas; te serán más fáciles de entender a medida que profundices en este libro.

1.3. Familiarizándonos con una Red SMB/CIFS.

Ahora que ya tienes una breve visión de Samba, tomémonos algún tiempo para familiarizarnos con el entorno que ha adoptado Samba: una red SMB/CIFS. Trabajar con redes SMB es significativamente diferente a trabajar con redes Unix TCP/IP, debido a que hay bastantes conceptos nuevos que aprender y mucha información a cubrir. Primero, discutiremos los conceptos básicos existentes tras una red SMB, seguido de algunas implementaciones de Microsoft a SMB, y finalmente te mostraremos dónde puede encajar un servidor Samba y dónde no.

1.3.1. Comprendiendo NetBIOS

Para comenzar, volvamos al pasado. En 1984, IBM diseñó un simple "application programming interface" (API) para conectar en red sus computadoras, llamado Network Basic Input/Output System (NetBIOS). El API NetBIOS proporcionaba un diseño rudimentario para que una aplicación se conectara y compartiese datos con otras computadoras.

Es útil pensar en el API NetBIOS como en extensiones de red para llamadas de la API BIOS estándard. Con BIOS, cada llamada de bajo nivel está confinada al hardware de la máquina local y no necesita ayuda para viajar a su destino. NetBIOS, sin embargo, originalmente tenía que intercambiar instrucciones con computadoras de redes IBM PC o Token Ring. Exigión por consiguiente un protocolo de tansporte de bajo nivel para llevar las peticiones de una computadora a la siguiente.

A finales de 1985, IBM lanzó dicho protocolo, el cual unión con el API NetBIOS para convertirse en NetBIOS Extended User Interface (NetBEUI). NetBEUI fue diseñado para redes de área local (LANs), y permitía a cada máquina usar un nombre (de hasta 15 caracteres) que no estuviera siendo usado en la red. Entendemos por pequeña LAN, a una red de menos de 255 nodos -¡Esto se consideraba un restricción práctica en 1985!-.

El protocolo NetBEUI se volvió muy popularen las aplicaciones de red, incluyendo a las que corrían bajo Windows para Grupos. Más tarde, emergieron también implementaciones de NetBIOS sobre protocolos IPX de Novell, los cuales competían con NetBEUI. Sin embargo, los protocolos de red escogidos por la comunidad de Internet eran TCP/IP y UDP/IP, y las implementaciones de las APIs NetBIOS sobre dichos protocolos pronto se convirtió en una necesidad.

Ten en cuenta que TCP/IP usa números para representar direcciones de computadoras, tales como 192.168.220.100, mientras que NetBIOS usa sólo nombres. Este fue el mayor problema a solucionar a la hora de hacer relacionarse a los dos protocolos. En 1987, El Internet Engineering Task Force (IETF) publicó una serie de documentos de estandarización, titulados RFC 1001 y 1002, que perfilaban cómo NetBIOS podría trabajar sobre una red TCP/UDP. Este juego de documentos todavía gobiernan a cada una de las implementaciones que existen hoy en día, incluyendo aquellas proporcionadas por Microsoft para sus sitemas operativos, así como a la suite Samba.

Desde entonces, la norma que estos documentos gobiernan se ha conocido como NetBIOS sobre TCP/IP, o NBT para abreviar. El estándar NBT (RFC 1001/1002) actualmente establece un trio de servicios sobre una red:

- Un Servicio de Nombres
- Dos Servicios de Comunicación:
  - Datagramas.
  - Sesiones.

El servicio de nombres resuelve el problema nombre-a-dirección comentado antes; pemite a cada computadora declarar un nombre específico en la red que pueda ser convertido a una dirección IP de máquina, como hacen hoy en día los DNS en Internet. Los servicios de datagramas y sesiones son ambos protocolos secundarios de comunicación, usados para transmitir datos desde y hacia máquinas NetBIOS a través de la red.

1.3.2. Obteniendo un Nombre

Para un ser humano, tener un nombre es sencillo. Sin embargo, para una máquina sobre una red NetBIOS, esto puede ser algo más complicado. Veamos algunos de esos problemas.

En el mundo NetBIOS, cuando cada máquina se vuelve activa, quiere reclamar un nombre para sí; esto se denomina registro de nombre. Sin embargo, dos máquinas en

el mismo grupo de trabajo podrían solicitar el mismo nombre; esto causaría problemas de confusión para cualquier máquina que quiera comunicar con una de esas dos. Hay dos aproximaciones diferentes para asegurarnos de que esto no ocurra:

- Usar un Servidor de Nombres NetBIOS (NBNS) para controlar el registro de nombres NetBIOS de las máquinas.
- Permitir a cada máquina de la red defender su nombre en el caso de que otra máquina intente usarlo.

La Figura 8 ilustra un registro de nombre (negado), con y sin Servidor de Nombres NetBIOS.

En adición, debe haber una forma de resolver un nombre NetBIOS hacia una dirección IP específica como ya mencionamos antes; esto es conocido como resolución de nombre. Hay dos formas diferentes también aquí con NBT:

- Haber reportado cada máquina su dirección IP cuando "escucha" una petición broadcast para su nombre NetBIOS.
- Usar el NBNS para resolver nombres NetBIOS a direcciones IP.

La Figura 9 ilustra los dos tipos de resolución de nombre.

Como te puedes imaginar, tener un NBNS en tu red te puede ayudar enormemente. Para ver exáctamente por qué, veamos el método sin-NBNS.

Aquí, cuando una máquina cliente arranca, manda un mensaje broadcast declarando que desearía registrar un nombre NetBIOS específico para ella. Si nadie objeta nada ante el uso de ese nombre tras múltiples intentos de registro, obtiene el nombre. En la otra parte, si otra máquina en la red está actualmente usando ese nombre, enviará un mensaje de respuesta al cliente solicitante indicando que ese nombre ya está siendo usado. Esto es conocido como defender el nombre de host. Este tipo de sistema es útil cuando un cliente ha caído inesperadamente de la red -otro puede tomar su nombre-, pero se incurre en un importante aumento del tráfico de la red para algo tan simple como el registro de nombre.

Con un NBNS, ocurre lo mismo, pero con la diferencia de que la comunicación se está confinada a la máquina solicitante y al servidor de nombres NBNS. No ocurre broadcasting cuando la máquina desea registrar el nombre; el mensaje de registro es simplemente enviado desde el cliente hacia el servidor NBNS, y este NBNS responde si el nombre está o no libre. Esto es conocido como comunicación punto-a-punto, y es beneficioso en redes con más de una subred. Esto se debe a que los routers suelen estar preconfigurados para bloquear paquetes entrantes que son mensajes de difusión (broadcast) para todas las máquinas de la red.

Los mismos principios se aplican a la resolución de nombres. Sin un NBNS, la resolución de nombres NetBIOS podría realizarse mediante un mecanismo broadcast. Todos los paquetes se enviarían a cada una de las computadoras de la red, con la esperanza de que alguna máquina que se vea afectada por la petición responda directamente a la máquina solicitante. En éste punto, queda claro que usar un servidor de nombres NBNS y una comunicación punto-a-punto para este propósito carga mucho menos la red que usar boradcasts para cada una de las peticiones de resolución de nombres que se produzcan.

Figura 1.6: Recursos disponibles en *hydra* (vistos por *chimaera*).

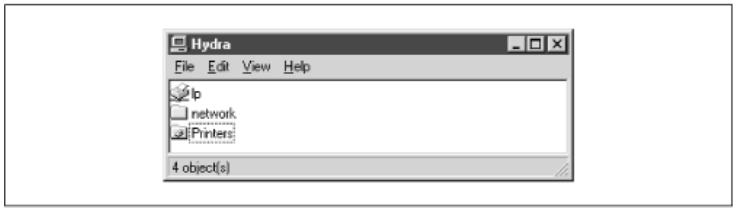
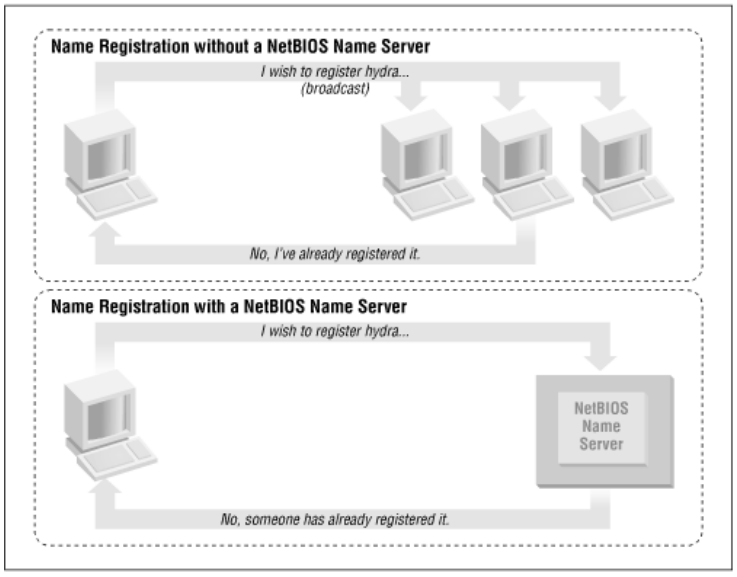


Figura 1.7: Una impresora de red disponible en *hydra* (vista por *chimaera*).

Figura 1.8: Registro de Nombre NBNS contra no-NBNS.



1.3.3. Tipos de Nodos

¿Y cómo le digo a los clientes qué estrategia deben seguir para realizar el registro de nombre y la resolución? Cada máquina en una red NBT aprende una de las siguientes designaciones, dependiendo de cómo se maneje el registro y la resolución de nombre: b-node, p-node, m-node y h-node. Las conductas de cada tipo de nodo se resumen en la Tabla 1.

Cuadro 1.1: Tipos de Nodos NetBIOS

Papel	**Valor**
b-node	Usa registro broadcast y sólo resolución.
p-node	Usa registro punto-a-punto y sólo resolución.
m-node	Usa broadcast para registro. Si tiene éxito, notifica al servidor NBNS el resultado. Usa broadcast para resolución; usa servidor NBNS si el broadcast no tiene éxito.
h-node (hybrid)	Usa servidor NBNS para registro y resolución; usa broadcast si el servidor NBNS no responde o no está operativo.

En el caso de los clientes Windows, los encontrarás listados normalmente como h-nodes o hybrid nodes. Incidentalmente,los h-nodes fueron inventados más tarde por Microsoft, como un tipo de nodo más tolerante a fallos de rutas, y no aparece en el RFC 1001/1002.

Puedes averiguar el tipo de nodo para cada máquina Windows tecleando el comando ipconfig /all y buscando la línea que pone Node Type.

```
C:\>ipconfig /all
Windows 98 IP Configuration
...
Node Type . . . . . . . . . . . : Hybrid
...
```

1.3.4. ¿Qué hay en un Nombre?

Los usos de creación de nombres NetBIOS son diferentes a los de los nombres tipo DNS a los que a lo mejor estarás más acostumbrado. Primero, los nombres NetBIOS existen en un espacio único. En otras palabras, no existen cualificadores del tipo ora.com o samba.org para definir secciones dentro de los nombres; sólo hay un nombre único para representar a cada computadora. Segundo, los nombres NetBIOS sólo pueden contener hasta 15 caracteres, no pueden comenzar con asterisco (*), y pueden consistir sólo en caracteres alfanuméricos estandard (a-z, A-Z, 0-9) y los siguientes:

```
! @ # $ % ^ & ( ) - ' { } . ~
```

Aunque puedes usar el punto (.) en un nombre NetBIOS, no te lo recomendamos, debido a que esos nombres puede que no funcionen en las futuras versiones de NetBIOS sobre TCP/IP.

No es una coincidencia que todos los nombres válidos DNS también sean válidos en NetBIOS. De hecho, el nombre DNS para un servidor Samba es frecuentemente reusado como su nombre NetBIOS. Por ejmplo, si tienes una máquina phoenix.ora.com , su nombre NetBIOS podría ser PHOENIX (seguido por 8 espacios en blanco).

1.3.4.1. Nombres de Recursos y Tipos

Con NetBIOS, una máquina no sólo advierte de su presencia, sino que también le dice a las otras máquinas qué tipo de servicios ofrece. Por ejemplo, phoenix puede indicar que no es sólo una estación de trabajo, sino que también es un servidor de ficheros y puede recibir mensajes WinPopup. Esto se hace añadiendo un byte (el 16) al final del nombre de máquina (recurso), llamado tipo de recurso, y registrando el nombre más de una vez. Mira la Figura 10

El tipo de recurso de 1 byte indica el único servicio que la máquina ofrece. En este libro, frecuentemente verás el tipo de recurso marcado entre símbolos de mayor/menor (<>) tras el nombre NetBIOS, como a continuación:

```
PHOENIX<00>
```

Puedes saber qué nombres están registrados para una máquina NBTdeterminada usando el comando de Windows NBTSTAT. Debido a que estos servicios son únicos (no puede haber más de uno registrado), los verás listados como tipo UNICO (UNIQUE) en la salida. Por ejemplo, la siguiente salida describe al servidor hydra:

```
D:\>NBTSTAT -a hydra
NetBIOS Remote Machine Name Table

Name            Type                        Status
---------------------------------------------------------
HYDRA       <00> UNIQUE    Registered
HYDRA       <03> UNIQUE    Registered
HYDRA       <20> UNIQUE    Registered
...
```

Esto indica que el servidor ha registrado el nombre NetBIOS hydra como nombre de máquina (estación de trabajo), un recipiente para mensajes WinPopup y un servidor de ficheros. Algunos de los posibles atributos que un nombre puede tener se listan en la Tabla 2.

Cuadro 1.2: Tipos de Recursos Unicos NetBIOS.

Nombre Recurso	**Hexidecimal Byte Value**
Standard Workstation Service	00
Messenger Service (WinPopup)	03
RAS Server Service	06
Domain Master Browser Service (associated with primary domain controller)	1B
Master Browser name	1D
NetDDE Service	1F
Fileserver (including printer server)	20
RAS Client Service	21
Network Monitor Agent	BE
Network Monitor Utility	BF

Figura 1.9: Resolución de nombre con-NBNS versus sin-NBNS.

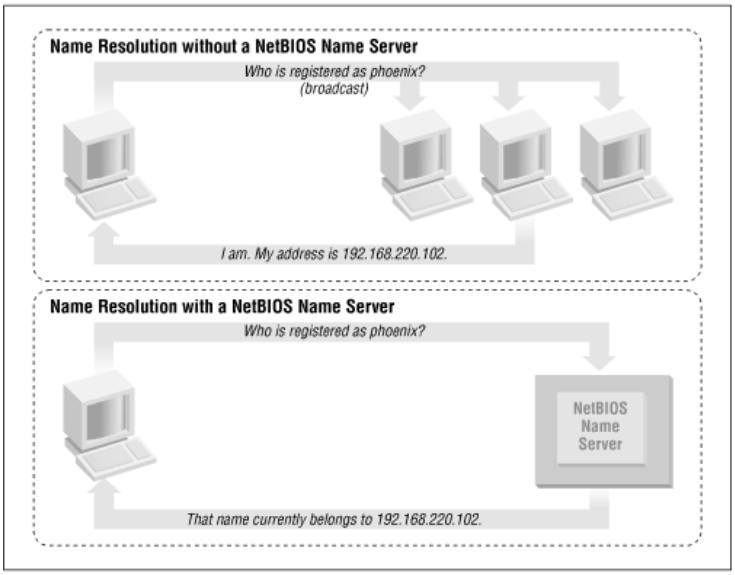
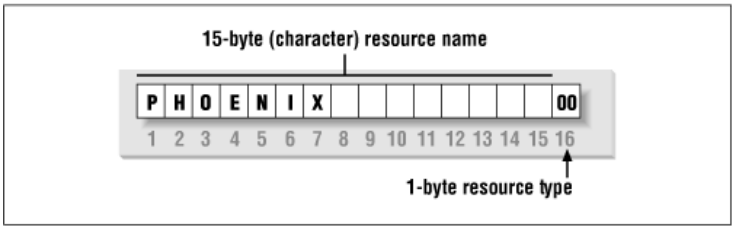


Figura 1.10: Estructura de un Nombre NetBIOS.



Advierte que debido a que los nombres DNS no tiene tipos de recursos, los diseñadores intencionadamente pusieron un valor hexadecimal 20 (un espacio en blanco) por defecto para el tipo de servidor de ficheros.

1.3.4.2. Nombres de Grupos y Tipos

SMB también usa el concepto de grupos. Anteriormente mencionamos que las máquinas de nuestro ejemplo pertenecían a un grupo de trabajo, el cual es una partición de másquinas en la misma red. Por ejemplo, una empresa podría tener fácilmente un grupo de trabajo ADMINISTRACION y otro VENTAS, cada uno con diferentes servidores e impresoras. En el mundo Windows, un grupo de trabajo y un grupo SMB son la misma cosa.

Continuando con nuestro ejemplo de NBTSTAT, el servidor Samba hydra es también un miembro del grupo de trabajo SIMPLE (el atributo GROUP hex 00), y estará disponible para ser elegido como visualizador maestro (atributo GROUP 1E). Mira la salida de NBTSTAT:

```
NetBIOS Remote Machine Name Table, continued
Name                   Type              Status
---------------------------------------------------------------
SIMPLE                 <00> GROUP        Registered
SIMPLE                 <1E> GROUP        Registered
..__MSBROWSE__.        <01> GROUP        Registered
```

Los posibles atributos de grupo que puede tener una máquina se ilustran en la Tabla 3. Para más información, Windows NT in a Nutshell de Eric Pearce, publicado por O'Reilly.

Cuadro 1.3: Tipos de Recursos de Grupo NetBIOS.

Nombre Recurso	**Valor Hexadecimal Byte**
Standard Workstation group	00
Logon Server	1C
Master Browser name	1D
Normal Group name (used in browser elections)	1E
Internet Group name (administrative)	20
<01><02>__MSBROWSE__<02>	01

La entrada final, **__MSBROWSE__** , se usa para anunciar un grupo a otros visualizadores maestros. Los caracteres no impresos en el nombre se muestran como guiones bajos en una salida de NBTSTAT. No te preocupes si no comprendes todos los recursos o tipos de grupos. Algunos de ellos no los necesitarás con Samba, y sobre los otros verás más el resto del capítulo. Lo importante aquí es recordar la lógica del mecanismo de nombres.

1.3.5. Datagramas y Sesiones.

Llegados a este punto, hagamos una introducción sobre otra responsabilidad de NBT: proporcionar servicios de conexión entre dos máquinas NetBIOS. Existen ac-

tualmente dos servicios ofrecidos por NetBIOS sobre TCP/IP: el servicio de sesiones y el servicio de datagramas. El comprender cómo funcionan estos dos servicios no es esencial para usar Samba, pero te va a dar una idea sobre cómo trabaja NBT y cómo arreglar problemas cuando Samba no funcione.

El servicio de datagramas ofrece una conexión no estable entre una máquina y otra. Los paquetes de datos son simplemente enviados o difundidos (broadcasting) de una máquina a otra, sin considerar el orden en que estos llegan al destino, o si han llegado todos. El uso de datagramas no incrementa tanto el trafico de la red como el uso de sesiones, aunque pueden echar abajo una red si se usan indebidamente (¿Te acuerdas de la difusión de la resolución de nombres de antes?) Los datagramas, por tanto, son empleados para enviar rápidamente sencillos bloques de datos a una o más máquinas. El servicio de datagramas comunica usando las primitivas simples mostradas en la Tabla 4.

Cuadro 1.4: Primitivas de Datagramas.

Primitiva	**Descripción**
Send Datagram	Envía paquete datagrama a máquina o grupos de máquinas.
Send Broadcast Datagram	Difunde (broadcast) datagrama a cualquier máquina, esperando un datagrama de acuse de recibo.
Receive Datagram	Recibe un datagrama de una máquina.
Receive Broadcast Datagram	Espera por un datagrama de difusión.

El servicio de sesiones es más complejo. Las sesiones son un método de comunicación que, en teoría, ofrece la capacidad de detectar conexiones problemáticas o inoperativas entre dos aplicaciones NetBIOS. Esto lleva a pensar en una sesión NBT en términos de una llamada telefónica. Una conexión full-duplex es abierta entre una máquina que llama y una máquina que es llamada, y la conexión debe permanecer abierta durante la duración de la conversación. Cada parte implicada conoce a la otra máquina, y pueden comunicar con las primitivas que se muestran en la Tabla 5.

Cuadro 1.5: Primitivas de Sesiones.

Primitiva	**Descripción**
Call	Inicia una sesión con una máquina que está a la escucha bajo un nombre específico.
Listen	Espera una llamada de un llamante conocido o cualquier otro.
Hang-up	Termina una llamada.
Send	Envía datos a la otra máquina.
Receive	Recibe datos de la otra máquina.
Session Status	Obtiene información sobre sesiones pedidas.

Las sesiones son el troncal de la compartición de recursos en una red NBT. Son normalmente usadas para establecer conexiones estables desde máquinas clientes a unidades de disco o impresoras compartidas en un servidor. El cliente "llama" e inicia la conversación, enviando información del tipo qué ficheros desea abrir, qué datos quiere intercambiar, etc. Estas llamadas pueden durar mucho tiempo -horas, incluso

días- y todo esto ocurre dentro del contexto de una única conexión. Si se produce un errorerror, el software de sesión (TCP) retransmitirá hasta que los datos sean recibidos correctamente, a diferencia del "envía-y-reza" del servicio de datagramas (UDP).

En realidad, mientras que las sesiones se supone están para manejar comunicaciones problemáticas, normalmente no lo hacen. Como probablemente habrás descubierto al usar redes Windows, es un serio problema el usar sesiones NBT. Si la conexión es interrumpida por la razón que sea, la información de sesión que está abierta entre dos computadoras puede fácilmente volverse inválida. Si esto ocurre, la única forma de restablecer la sesión para las dos mismas máquinas es llamar de nuevo y comenzar desde ceero.

Si deseas más ínformación sobre cada uno de estos servicios, te recomendamos mires el RFC 1001. Sin embargo, hay dos cosas importantes a recordar aquí:

- Las sesiones siempre ocurren entre dos máquinas NetBIOS -ni más ni menos-. Si un servicio de sesión es interrumpido, se supone que el cliente ha almacenado la suficiente información de estado como para restablecer la comunicación. Sin embargo, en la práctica, es raro el caso.

- Los datagramas pueden ser difundidos a múltiples máquinas, pero son inestables. Dicho de otro modo, no hay forma para el emisor de saber si los datagramas que ha enviado han llegado correctamente a los destinatarios.

1.4. Implementaciones de Microsoft.

Con todo lo anterior de fondo, ahora podemos hablar sobre algunas de la implementaciones de Microsoft sobre los anteriores conceptos del mundo de las redes CIFS/SMB. Y como te esperas, también introduciremos sobre algunas más complejas extensiones.

1.4.1. Dominio Windows.

Recuerda que un grupo de trabajo es una colección de computadoras SMB, las cuales residen todas en la misma subred y se encuentran suscritas al mismo grupo SMB. Un Dominio Windows va un paso más allá. Es un grupo de trabajo de máquinas SMB que tienen una añadido: un servidor que actúa como controlador de dominio. Debes tener un controlador de dominio para poder tener un dominio Windows [6]. Por otra parte, se trata sólo de un grupo de trabajo. Mira la Figura 11. [6] Los dominios Windows son llamados "Dominios Windows NT" por Microsoft porque ellos asumen que serán máquinas Windows NT las que asuman el papel de controladoras de dominio. Sin embargo, como Samba puede realizar ésta función también, nosotros simplemente hablaremos de "Dominios Windows" para evitar confusiones.

Hay actualmente dos protocolos separados usados por un controlador de dominio (logon server): uno para comunicar con máquinas Windows 95/98 y otro para comunicar con máquinas Windows NT. Mientras que Samba actualmente imlementa el protocolo controlador de dominio para máquinas Windows 95/98 (lo cual nos permite actuar como controlador de dominio para máquinas Windows 9 x), todavía no está completamente soportado el protocolo para máquinas Windows NT. Sin embargo, el equipo de desarrollo de Samba promete que dicho soporte para el protocolo controlador de dominio para máquinas Windows NT estará en la versión Samba 2.1.

¿Por qué tanta complejidad? El protocolo que el controlador de dominio de Windows usa para comunicar con sus clientes y con otros controladores de dominio es propietario y no ha sido liberado por Microsoft. Esto ha forzado al equipo de Samba a utilizar una ingeniería inversa sobre el protocolo controlador de dominio para ver qué códigos realizan qué taréas.

1.4.1.1. Controlador de Dominio.

El controlador de dominio es el centro nervioso de un dominio Windows, tal como un servidor NIS lo es del servicio de información de una red Unix. Los controladores de dominio tienen una serie de responsabilidades. Una de las que te va a implicar a ti es la autentificación. La autentificación es el proceso de garantizar o denegar a un usuario el acceso a recursos compartidos o a otra máquina de la red, normalmente a través del uso de una password.

Cada controlador de dominio usa un security account manager (SAM) para mantener una lista de combinaciones nombre_usuario-contraseña. El controlador de dominio entonces forma una central repositoria de passwords que están enlzadas a nombres de usuarios (una password por usuario), lo cual es más eficiente que mantener en cada máquina cliente centenares de passwords para cada recurso de red disponible.

En un dominio Windows, cuando un cliente no autorizado solicita acceso a los recursos compartidos de un servidor, el servidor actúa y pregunta al controlador de dominio si ese usuario está autentificado. Si lo está, el servidor establecerá una conexión de sesión con los derechos de acceso correspondientes para ese servicio y usuario. Si no lo está, la conexión es denegada. Una vez un usuario es autentificado por el controlador de dominio, una ficha especial de autentificación será retornada al cliente, de manera que el usuario no necesitará relogearse a otros recursos en ese dominio. En éste punto, el usuario se considera "logeado" en el dominio. Mira la Figura 12.

1.4.1.2. Controlador de Dominio Primario y de Seguridad.

La redundancia es una idea clave dentro de un dominio Windows. El controlador de dominio que está actualmente activo sobre un dominio es denominado como el Controlador Primario de Dominio (PDC). Además pueden existir uno o más Controladores de Dominio de Seguridad (BDCs) en el dominio, los cuales actuarán en caso de que el controlador primario falle o se vuelva inaccesible. Los BDCs frecuentemente sincronizan sus datos SAM con el controlador primario de dominio, de manera que si llegara el caso, cualquiera de ellos podría realizar servicios DC transparentemente sin provocar ningún tipo de impacto en los clientes. Advierte que los BDCs, sin embargo, sólo tienen copias de sólo lectura del SAM; pueden actualizar sus datos sólo mediante la sincronización con un PDC. Un servidor en un dominio Windows puede usar los SAM de cualquier controlador de dominio primario o de seguridad para autentificar a un usuario que intenta acceder a los recursos y logearse en el dominio.

Ten en cuenta que en muchos aspectos, las características de un grupo de trabajo de Windows y un dominio de Windows se pisan. Esto no es algo accidental, ya que el concepto de los dominios Windows no apareció hasta la aparición de Windows NT 3.5, y los dominios Windows fueron forzados a permanecer compatibles con los grupos de trabajo presentes en Windows for Workgroups 3.1. La cosa clave a recordar aquí es que un dominio es simplemente un grupo de trabajo de Windows con uno o más controladores de dominio añadidos.

Samba puede funcionar como controlador primario de dominio para máquinas Windows 95/98 sin ningún tipo de problemas. Sin embargo, Samba 2.0 puede actuar como controlador primario de dominio sólo para procesos de autentificación; actualmente no puede asumir ninguna otra de las responabilidades de un PDC. (mientras lees este manual, Samba 2.1 puede que ya esté disponible, de forma que podrás usar Samba como PDC para clientes NT). Por otra parte, y "gracias" a la privacidad del protocolo usado por Microsoft para sincronizar datos SAM, Samba actualmente no puede servir como controlador de dominio de seguridad.

1.4.2. Visualización (Browsing).

La visualización, navegación o browsing es una respuesta de alto nivel para la pregunta del usuario: "¿Qué máquinas están ahí en la red Windows?". Recuerda que no hay conexión alguna con un navegador web, aparte de la idea general de "descubrir qué hay por ahí fuera". Y, al igual que en la web, lo que está ahí fuera puede cambiar sin previo aviso.

Antes de visualizar, los usuarios deben connocer el nombre de la máquina específica a la que quieren conectarse desde la red, y entonces manualmente introducir un UNC al como el siguiente en una aplicación o gestor de ficheros para acceder a los recursos:

```
\\HYDRA\network\
```

Con la visualización, sin embargo, puedes examinar los contenidos de una máquina usando un típico interfaz apuntar-y-hacer-click. Por ejemplo, la ventana de Entorno de Red desde un cliente Windows.

1.4.2.1. Niveles de Visualización.

Como indicamos al principio del capítulo, existen actualmente dos tipos de visualización, navegación o browsing con los que te podrás encontrar en una red SMB/CIFS:

- Visualizar una lista de máquinas (con recursos compartidos).
- Visualizar los recursos compartidos de una máquina determinada.

Veamos el primero de ellos. En cada subred de cada grupo de trabajo Windows (o dominio), una computadora tiene la responsabilidad de mantener una lista de las máquinas que están actualmente accesibles a través de la red. Esta computadora es denominada la visualizadora local maestra, y la lista que mantiene es llamada la Lista de Visualización. Las máquinas de una subred usan la lista de visualización para reducir la cantidad de tráfico de la red que se genera durante la visualización. En vez de que cada máquina genere su propia lista de máquinas disponibles, la computadora puede simplemente interrogar al visualizador maestro local para obtener una completa y actualizada lista.

Para visualizar los recursos actuales de una máquina, un usuario debe conectar a la máquina específica; esa información no puede ser obtenida de la lista de visualización. El ver la lista de recursos compartidos de una máquina se puede hacer haciendo click cobre el icono que la representa en la ventana de Entorno de Red de Windows 95/98 o NT. Como ya vimos al principio del capítulo, la máquina responderá con una lista de recursos compartidos que pueden ser accedidos si ese usuario se encuentra debidamente autentificado.

Cada uno de los servidores de un grupo de trabajo de Windows está obligado a anunciar su presencia al visualizador maestro local una vez haya registrado su nombre NetBIOS, y (teóricamente) a anunciar que está dejando el grupo de trabajo cuando se apaga. Es responsabilidad del visualizador maestro local registrar las máquinas. Advierte que el visualizador maestro local no tiene por qué ser necesariamente la misma máquina que el servidor de nombres NetBIOS (NBNS), sobre el cual hablamos anteriormente.

> **ADVERTENCIA**: El Entorno de Red de Windows puede comportarse de forma extraña: hasta que no selecciones una máquina determinada para visualizarla, la ventana del Entrono de Red puede contener datos no actualizados. Esto significa que en la ventana del Entorno de Red pueden aparecer máquinas que están actualmente apagadas, o bien puede obviar a otras máquinas que actualmente están declaradas en el grupo de trabajo. Una vez hayas seleccionado un servidor y hayas conectado a él, entonces puedes tener la seguridad de que los recursos compartidos que muestra realmente existen.

Al contrario de los roles que hemos visto antes, casi cualquier máquina Windows (NT Server, NT Workstation, 98, 95, o Windows 3.1 for Workgroups) puede actuar como visualizador maestro local. Como con los controladores de dominio, el visualizador maestro local puede tener uno o más visualizadores de seguridad en la subred local que pueden actuar en el caso de que el visualizador maestro local falle o se vuelva inaccesible. Para asegurar una operación fluida, los visualizadores de seguridad locales sincronizarán frecuentemente su lista de visualización con el visualizador maestro local. Actualicemos nuestro diagrama de dominio Windows para incluir un visualizador maestro local y otro de seguridad.

Los resultados se muestran en la Figura 13.

Aquí tienes cómo calcular el número mínimo de visulizadores de seguridad que deben existir en un grupo de trabajo:

- Si hay entre 1 y 32 Windows NT workstations en la red, o entre 1 y 16 máquinas Windows 95/98 en la red, el visualizador maestro local coloca un visualizador de seguridad en adición al maestro.
- Si el número de Windows NT workstations está entre 33 y 64, o el número de máquinas Windows 95/98 está entre 17 y 32, el visualizador maestro local ubica dos visualizadores de seguridad.
- Por cada grupo de 32 NT workstations o 16 máquinas Windows 95/98 de más,el visualizador maestro local coloca un visualizador de seguridad más.

Actualmente no existe límite de visualizadores de seguridad que se puedan ubicar por parte del visualizador maestro loca.

1.4.2.2. Elección de Visualizador.

La visualización es un aspecto crítico de cualquier grupo de trabajo Windows. Sin embargo, no todo funciona perfectamente en cualquier red. Por ejemplo, digamos que el servidor Windows NT del despacho del CEO de una pequeña compañía es el visualizador maestro local -esto es, hasta que él lo apague mientras se va a su sesión de masaje-. En éste punto, la estación de trabajo Windows NT Workstation de la sección

Figura 1.11: Un Simple Dominio Windows.

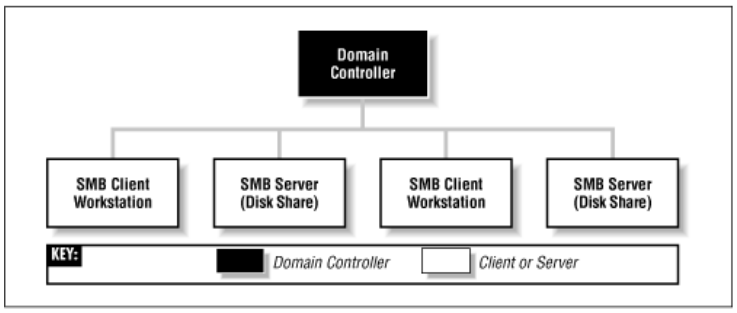


Figura 1.12: Usando un controlador de dominio para autentificación.

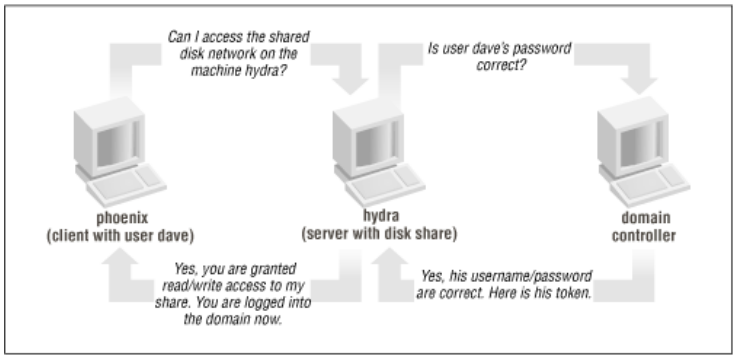
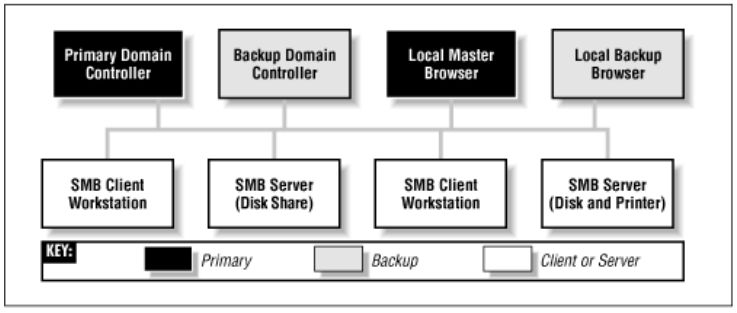


Figura 1.13: Un dominio Windows con un visualizador maestro local y otro de seguridad.



del departamento de repuestos podría estar de acuerdo en tomar el relevo. Sin embargo, esa máquina está actualmente ejecutando un programa realmente enorme, pobremente escrito, y que está comiéndose literalmente la capacidad de proceso del microprocesador. La moraleja: la visualización debe ser muy tolerante con los servidores que vienen y van. Debido a que cada máquina Windows puede llegar a actuar como un visualizador, debe existir una forma de decidir en cada momento quién se va a hacer cargo del trabajo. Este proceso de toma de decisión se denomina elección.

Un algoritmo de elección se construye en casi todos los sistemas operativos Windows, de manera que cada uno de ellos puede ponerse de acuerdo con los demás sobre quién va a ser el visualizador maestro local y quiénes van a actuar como visualizadores de seguridad. Una elección puede forzarse en cualquier momento. Por ejemplo, asumamos que el CEO ha finalizado su masaje y reinicia el servidor. Cuando el servidor se activa de nuevo, éste anuncia su presencia y una elección tendrá lugar para ver si el PC en el departamento de repuestos debería continuar siendo el visualizador maestro.

Cuando se realiza una elección, cada máquina difunde información sobre sí misma en forma de datagramas. Esta indormación incluye:

- La versión del protocolo de elección usado.
- El sistema operativo de la máquina.
- La cantidad de tiempo que el cliente ha estado en la red.
- El nombre de host del cliente.

Estos valores determinan qué sistema operativo tiene más potencia y cumplirá mejor el rol de visualizador maestro local. (El Cap. 6, Usuarios Seguridad y Dominios, describe el proceso de elección con más detalle). La arquitectura desarrollada para conseguir esto no es demasiado elegante que digamos, y ha llevado a muchos problemas de seguridad. Mientras que un dominio de visualización puede estar integrado con un dominio de seguridad, el algoritmo de elección no toma en consideración qué computadoras se vuelven visualizadores. Así, es posible para cualquier máquina ejecutar un servicio de visualización para registrarse a sí misma como participante en la elección de visualizador, y (tras ganar) estar habilitada para cambiar la lista de visualización. No obstante, la visualización es un rasgo importante de las redes Windows, y los requerimientos de mantener la compatibilidad con versiones anteriores de s.o. Windows le asegura estar en uso durante muchos años.

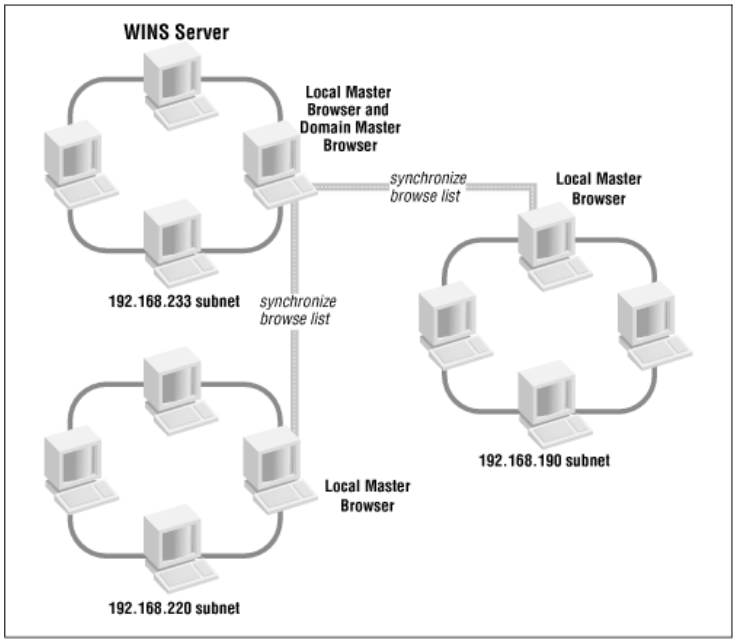
1.4.3. ¿Puede un Grupo de Trabajo Windows abarcar varias SubRedes?

Sí, pero la mayoría de la gente que se ha metido con esto ha tenido muchos quebraderos de cabeza. Abarcar múltiples subredes no era parte del diseño inicial de Windows NT 3.5 o Windows for Workgroups. Como resultado, un dominio Windows que que abarca dos o más subredes es, en realidad, un "encolado" de dos o más grupos de trabajo que comparten un nombre idéntico. La buena noticia es que todavía podrás usar un controlador primario de dominio para control de autentificación en cada una de las subredes. La mala noticia es que las cosas no son tan sencillas en el caso de la visualización.

Como mencionamos antes, cada subred tiene su propio visualizador maestro local. Cuando un dominio Windows abarca múltiples subredes, un administrador del sistema

tendrá que asignar una de las máquinas como el visualizador maestro de dominio. El visualizador maestro de dominio mantendrá una lista de visualización para todo el dominio Windows. Esta lista de visualización es creada por la sincronización periódica de la lista de visualización del visualizador maestro de dominio con las listas de visualización de cada uno de los visualizadores maestros locales. Tras la sincronización, el visualizador maestro y el visualizador maestro de dominio deberían contener entradas idénticas. Mira la Figura 14 como ilustración.

Figura 1.14: Un grupo de trabajo que abarca más de una subred.



¿Te suena bien? Bueno, pues eso no es "el cielo" por las siguientes razones:

- Si existe, un controlador primario de dominio siempre juega el papel de visualizador maestro de dominio. Debido al diseño de Microsoft, los dos siempre comparten el tipo de recurso NetBIOS <1B>, y (desafortunadamente) no pueden ser separados.

- Las máquinas Windows 95/98 no pueden convertirse, ni siquiera contactar con un visualizador maestro de dominio. El equipo de Samba cree que esto es una decisiónd e marketing por parte de Microsoft, que fuerza a los consumidores a tener, al menos una máquina Windows NT workstation (o servidor Samba) en cada subred de un grupo de trabajo multi-red.

Cada visualizador maestro de cada subred local continua manteniendo la lista de visualización para su subred, para la cual se vuelve autoritativo. Así, si una computadora desea ver una lista de los servidores dentro de su propia subred, el visualizador maestro local de esa subred será interrogado. Si una computadora quiere ver una lista de servidores fuera de su subred, sólo podrá llegar hasta donde le lleve el visualizador maestro local. Pero esto funciona porque, a intervalos fijados, la lista de visualización

autoritativa del visualizador maestro local de una subred es sincronizado con el visualizador maestro de dominio, el cual está sincronizado con el visualizador maestro local de las otras subredes en el dominio. Esto se denomina propagación de la lista de visualización.

Samba puede actuar como visualizador maestro de dominio en un dominio Windows si es necesario. En adición, también puede actuar como visualizador maestro local para una subred Windows, sincronizando su lista de visualización con el visualizador maestro de dominio.

1.4.4. El Servicio de Nombres de Internet de Windows (WINS).

El Servicio de Nombres de Internet de Windows, o "Windows Internet Name Service" (WINS) es la implementación de Microsoft de un servidor de nombres NetBIOS (NBNS). Como tal, WINS hereda muchas de las características de NetBIOS. Primero, WINS funciona con nombres simples o llanos; sólo puedes tener máquinas llamadas fred o grupos de trabajo como CANADA o USA. En adición, WINS es dinámico: cuando un cliente se vuelve "aparece" en la red, se le requiere para que reporte su nombre de máquina, su dirección y su grupo de trabajo al servidor WINS local. Este servidor WINS retendrá la información mientras el cliente periódicamente refresque su registro WINS, lo cual indica que todavía está conectado a la red. Advierte que los servidores WINS no son específicos de un grupo de trabajo o dominio; pueden aparecer en cualquier lugar y servir a cualquiera.

Pueden configurarse múltiples servidores WINS para sincronizarse unos con otros tras determinado paso de tiempo. Esto permite entradas de máquinas que aparecen y desaparecen en la red para propagarse de un servidor WINS a otro. Mientras que en teoría esto debería ser eficiente, podría volverse problemático rápidamente si hay varios servidores WINS cubriendo una red. Debido a que los servicios WINS pueden controlar múltiples subredes, frecuentemente es más eficiente tener a cada cliente Windows, no importa cuántos dominios Windows haya, apuntando al mismo servidor WINS. De esa forma, sólo habrá un servidor WINS autoritativo con la información correcta, en lugar de tener varios servidores WINS esforzándose contínuamente en sincronizarse entre ellos con los cambios más recientes.

El actual servidor WINS activo es conocido como el servidor WINS primario. También puedes instalar un servidor WINS secundario, el cual entrará en acción en el caso de que el primario falle o se vuelva inaccesible. Advierte que no hay un proceso de elección para determinar qué máquina se convierte en servidor WINS primario o de seguridad -la elección de servidores WINS es estática y debe ser predeterminada por el administrador del sistema. Tanto el servidor WINS primario como el de seguridad sincronizarán sus bases de datos de direcciones cada ciertos períodos determinados de tiempo.

En la familia de sistemas operativos Windows, sólo un servidor NT Workstation o NT pueden actuar como servidores WINS. Samba también puede funcionar como servidor WINS primario, pero no como secundario.

1.4.5. ¿Qué puede hacer Samba?

¡Vaya! nunca habrías pensado que las redes Microsoft podrían ser tan complejas, verdad? Ahora, centrémonos en ver dónde nos puede ayudar Samba. La Tabla 6 sumariza los roles que Samba puede y no puede asumir en un dominio Windows NT o en un dominio Windows para Trabajo en Grupos. Como podrás ver, debido a que la

mayoría de protocolos del dominio NT son propietarios y no han sido documentados por Microsoft, Samba no puede sincronizar correctamente sus datos con un servidor Microsoft y no puede actuar como servidor de seguridad en la mayoría de los roles. Sin embargo, con las versiones 2.0. x, Samba ofrece soporte limitado para los protocolos de autentificación del controlador primario de dominio y está adquiriendo nuevas funcionalidades cada día.

Cuadro 1.6: Roles de Samba (desde 2.0.4b).

Rol	**¿Puede hacerlo?**
Servidor de Archivos	Sí
Servidor de Impresión	Sí
Controlador Primario de Dominio	Sí (Samba 2.1 o superior recomendado)
Controlador de Dominio de Seguridad	No
Autentificación de clientes Windows 95/98	Sí
Visualizador Maestro Local	Sí
Visualizador de Seguridad	No
Visualizador Maestro de Dominio	Sí
Servidor WINS Primario	Sí
Servidor WINS Secundario	No

1.5. Un Vistazo a la Distribución Samba.

Como mencionamos antes, Samba actualmente contiene varios programas que sirven para diferentes pero determinados propósitos. Vamos a introducirnos en cada uno de ellos brevemente, y veremos cómo funcionan todos ellos. La mayoría de los programas que vienen con la distribución de Samba se centran en sus dos demonios. Echemos un vistazo a las responsabilidades de cada demonio:

smbd El demonio smbd es responsable de manejar los recursos compartidos entre la máquina servidora Samba y sus clientes. Proporciona servicios de archivos, impresión y visualización a los clientes SMB a través de una o más redes. smdb controla todas las notificaciones entre el servidor Samba y los clientes de red. En adición, es responsable de la autentificación de usuarios, bloqueo de recursos y la compartición de datos a través del protocolo SMB.

nmbd El demonio nmbd es un sencillo servidor de nombres que imita la funcionalidad de los servidores WINS y de resolución de nombres NetBIOS. Este demonino está a la escucha de peticiones para el servidor de nombres y proporciona la información apropiada cuando se le llama. También proporciona listas de visualización del Entorno de Red y participa en las elecciones de los visualizadores.

La distribución de Samba también está acompañada por un pequeño grupo de herramientas tipo línea de comandos Unix:

smbclient Un cliente tipo FTP Unix que puede ser usado para conectar a recursos compartidos por Samba.

smbtar Un programa para realizar copias de seguridad de datos sitos en los recursos compartidos, similar al comando Unix "tar".

nmblookup Un programa que proporciona búsquedas de nombres NetBIOS sobre TCP/IP.

smbpasswd Un programa que permite a un administrador cambiar las passwords encriptadas usadas por Samba.

smbstatus Un programa para reportar las conexiones de red actuales hacia los recursos compartidos por el servidor Samba.

testparm Un simple programa para validar el fichero de configuración de Samba.

testprns Un programa que testea si varias impresoras son reconocidas por el demonio smbd.

Cada nueva versión de Samba es sometida a muchos testeos antes de ser anunciada. Además, es rápidamente actualizada si se detectan problemas o bugs. La distribución más estable a la fecha de este libro es la 2.0.5. Este libro se centra en las funcionalidades soportadas por la versión 2.0, en contraposición con las versiones 1.9. x de Samba, las cuales están ya obsoletas.

1.6. ¿Cómo puedo Obtener Samba?

Samba está disponible en formato binario y fuente en una serie de mirrors en Internet. El sitio principal de Samba está localizado en http://www.samba.org.

Sin embargo, si no quieres esperar a que los paquetes te lleguen desde Australia, los mirros para Samba los puedes encontrar en muchas direcciones de Internet. Tienes una lista de ellos en la página principal del sitio oficial de Samba.

Capítulo 2

Instalando Samba en un Sistema Unix.

Ahora que sabes lo que Samba puede hacer por ti y por tus usuarios, es hora de entrar en la configuración de nuestra propia red. Comenzaremos con la instalación de Samba sobre un sistema Unix. Aprenderemos pasito a pasito a instalar Samba. Este capítulo te ayudará a empezar con buen pie.

Para propósitos ilustrativos, instalaremos la versión 2.0.4 del servidor Samba sobre un sistema Linux corriendo la versión 2.0.31 del núcleo. De todas formas, los pasos de instalación son los mismos para todas las plataformas que Samba soporta. Una instalación típica nos llevará una hora, incluyendo la descarga de los fuentes y su compilación, tocar los ficheros de configuración, y testear el servidor.

Aquí tienes una vista rápida de los pasos:

1. Descargar los fuentes o binarios de la distribución.
2. Leer la documentación sobre instalación.
3. Configurar el makefile.
4. Compilar el código del servidor.
5. Instalar los ficheros del servidor.
6. Crear un fichero de configuración de Samba.
7. Testear el fichero de configuración.
8. Iniciar los demonios de Samba.
9. Testear los demonios de Samba.

2.1. Descargando la Distribución.

Si quieres bajarte la última versión, el sitio web principal de Samba es `http://www.samba.org`. Una vez conectes a esta página, verás enlaces a los sitios espejo de Samba distribuidos por el mundo. Escoge el sitio más cercano a ti.

Los sitios web de Samba tienen documentación y tutoriales, listas de correo, las últimas notivias sobre Samba, y las distribuciones de fuentes y binarios de Samba. Los sitios de descarga (FTP) sólo tienen las distribuciones de fuentes y binarios. A menos que específicamente desees una versión en concreto o necesites instalar una distribución binaria, descargate la última distribución de fuentes desde el sitio ftp más cercano. Esta distribución siempre se llama:

```
samba-latest.tar.gz
```

2.1.1. ¿Binarios o Fuentes?

Los paquetes precompilados están disponibles también para un gran número de plataformas Unix. Estos paquetes contienen binarios para cada uno de los ejecutables de Samba, así como la documentación base de Samba. Advierte mientras que la instalación de una distribución de binarios te puede ahorrar problemas y tiempo, hay unas cuantas cosas que deberías tener en mente cuando vayas a decidir uasr los binarios o compilar tú mismo:

- Los paquetes de binarios pueden estar atrasados con respecto a la última versión del en una o dos revisiones (incluso más), especialmente tras una serie de pequeños cambios o para el caso de las plataformas menos populares. Compara las notas de revisión de los paquetes de fuentes y de binarios para asegurarte de que no hay nuevas características que necesites para tu plataforma.
- Si usas binarios precompilados, necesitarás asegurarte de que tienes las librerías correctas requeridas por los ejecutables. En algunas plataformas, los ejecutables están estáticamente enlazados, por lo cual esto no sería problema, pero en los s.o. Unix modernos (p.ej., Linux, SGI Irix, Solaris, HP-UX, etc.), las librerías frecuentemente están enlazadas dinámicamente. Esto significa que los binarios buscan la versión correcta de cada librería en tu sistema, así que podrías tener que instalar una nueva versión de una librería para compilar. El fichero README o makefile que acompañan a la distribución de binarios debería especificar cualesquiera requerimientos[1]. Muchas máquinas con librerías compartidas vienen con una utilidad llamada ldd. Esta herramienta te dirá qué librerías requiere un determinado binario, y qué librerías en el sistema satisfacen dicho requerimiento. Por ejemplo, el testeo del programa smbd en nuestra máquina dio el siguiente resultado:

  ```
  \textbf{$ ldd smbd}
  libreadline.so.3 => /usr/lib/libreadline.so.3
  libdl.so.2 => /lib/libdl.so.2
  libcrypt.so.1 => /lib/libcrypt.so.1
  libc.so.6 => /lib/libc.so.6
  libtermcap.so.2 => /lib/libtermcap.so.2
  /lib/ld-linux.so.2 => /lib/ld-linux.so.2
  ```

- Si hay cualquier incompatibilidad entrer Samba y librerías específicas en tu máquina, la documentación base de la distribución te lo debería aclarar.

[1] Esto es especialmente cierto en programas que usan glibc-2.1 (que viene con Red Hat Linux 6). Esta librería causó consternación en la comunidad de desarrolladores cuando fue revisada, porque se hizo incompatible con versiones anteriores de glibc.

- Recuerda que cada distribución de binarios trae valores preestablecidos sobre la plataforma de instalación, tales como directorios por defecto y valores de algunas opciones de configuración. De nuevo, comprueba la documentación y el fichero makefile incluido en el directorio fuente para ver qué directivas y variables fueron usadas cuando se compilarion los binarios. En algunos casos, podrían no ser apropiados para tu situación.
  Unos cuantos elementos de configuración pueden ser reseteados con opciones de línea de comandos en tiempo de ejecución, en lugar de en tiempo de compilación. Por ejemplo, si tu binario intenta colocar un archivo de registro, bloqueo o de estado en el lugar "inapropiado" (p.ej., en */usr/local*), podrías evitar esto sin necesidad de recompilar.

Un punto a comentar es que los fuentes de Samba requieren un compilador ANSI C. Si estás en una plataforma con un compilador no-ANSI, tal como el compilador *cc* de SunOS version 4, tendrás que instalar un compilador compatible ANSI como *gcc* antes de que puedas hacer nada[2].

2.1.2. Leer la Documentación.

Aunque parezca algo obvio, si bien puedes descomprimir el paquete y acto seguido hacer *configure*, *make*, y *make install*, y a lo mejor hasta te funciona, es una mala idea el no dedicar algo de tiempo a leer la documentación, sobre todo cuando planeamos una red con Samba.

Samba 2.0 se configura automáticamente antes de la compilación. Esto reduce la problemática que pueda existir con problemas específicos de determinadas máquinas, pero puede que haya una opción mencionada en el fichero README que tú no hayas tenido en cuenta, y que te va a resultar imprescindible para tu instalación. Tanto con un tipo de distribución como otra, encontrarás un buen número de documentos en el directorio docs, en variedad de formatos. Los fichero más importantes a leer en toda distribución son:

```
WHATSNEW.txt
docs/textdocs/UNIX_INSTALL.txt
```

Estos ficheros te indican cuáles son las características con las que te vas a encontrar en tu distribución de Samba, y te advertirá sobre problemas típicos de instalación con los que te podrás encontrar. Asegúrate de echarles al menos un vistazo antes de comenzar con la compilación.

2.2. Configurando Samba.

La distribución de fuentes de Samba 2.0 y anteriores no tenían inicialmente un *makefile*. En su lugar, se generaba uno a través de un script GNU *configure*, el cual se localizaba en el directorio *samba-2.0.x /source/*. El script *configure*, que debería ser ejecutado como *root*, se encarga de las configuraciones específicas a la máquina destino donde se iba a instalar Samba. Sin embargo, todavía puedes decidir sobre algunas opciones globales. Las opciones globales pueden ser establecidas pasando opciones a través de la línea de comandos:

[2]Los binarios *gcc* están disponibles para la mayoría de las máquinas modernas. Mira en http://www.gnu.org/ para ver una lista de los sitios con *gcc* y otro software GNU.

```
# ./configure --with-ssl
```

Por ejemplo, esto configurará el fichero *makefile* para Samba con soporte para el protocolo de encriptación *Secure Sockets Layer* (SSL). Si quieres ver la lista completa de opciones, teclea lo siguiente:

```
#./configure --help
```

Cada una de estas opciones activa/desactiva varias características. Normalmente activarás una característica especificando la opción *–con-opcion*, la cual causará que la característica sea compilada e instalada. Del mismo modo, si especificas una opción *–sin-opcion*, la característica será desactivada. Desde Samba 2.0.5, cada una de las siguientes características están desactivadas por defecto:

–with-smbwrapper Incluye soporte SMB wrapper, lo cual permite a los ejecutables de la parte Unix acceder a sistemas de ficheros SMB/CIFS como si fueran sistemas de ficheros de Unix. Recomendamos usar esta opción. Sin embargo, al tiempo de la escritura de este libro, nos topamos con diversas incompatibilidades entre el paquete *smbwrapper* y la librería GNU *libc* versión 2.1, y no se pudo compilar en Red Hat 6.0. Busca más información sobre estas incompatibilidades en la página web de Samba.

–with-afs Incluye soporte para el sistema de ficheros de Andrew de la Universidad Carnegie Mellon. Si vas a servir ficheros AFS vía Samba, te recomendamos compilar Samba una vez primero sin activar esta característica, para asegurarte de que todo funciona bien. Una vez la versión funcione correctamente, recompila Samba con esta característica activada.

–with-dfs Incluye soporte para DFS, una versión posterior de AFS, usado por OSF/1 (Digital Unix). Nota que esto NO ES LO MISMO que el DFS de Microsoft, el cual es un sistema de ficheros completamente distinto. De nuevo, recomendamos compilar Samba primero sin ésta característica, y si todo funciona bien, recompila con ella activada.

–with-krb4=directorio-base

Incluye soporte para Kerberos version 4.0, especificando explícitamente el directorio base de la distribución. Kerberos es un protocolo de seguridad de red de MIT que usa cripografía para proporcionar seguridad entre nodos. Microsoft ha anunciado que Kerberos 5.0 será el mecanismo estandar de autentificación para Microsoft Windows 2000 (NT 5.0). Sin embargo, los mecanismo de autentificación de Kerberos 5.0 son bastante diferentes a los mecanismos de seguridad de Kerberos 4.0. Si tienes Kerberos version 4 en tu sistema, el equipo de Samba recomienda que actualices y uses la opción *–with-krb5* (mira el siguiente elemento). Puedes encontrar más información sobre Kerberos en http://web.mit.edu/kerberos/www.

–with-krb5=directorio-base Incluye soporte para Kerberos version 5.0, especificando explícitamente el directorio base de la distribución. Microsoft ha anunciado que Kerberos 5.0 será el mecanismo estandar de autentificación para Microsoft Windows 2000 (NT 5.0). Sin embargo, no hay garantías de que Microsoft no amplíe Kerberos para sus propias necesidades en el futuro. Actualmente, el soporte de Kerberos en Samba sólo usa un interfaz de claves de texto plano, no encriptadas. Puedes encontrar más información sobre Kerberos en http://web.mit.edu/kerberos/www.

–with-automount Incluye soporte para automounter, una característica frecuentemente usada en sitios que ofrecen NFS.

–with-smbmount Incluye soporte *smbmount*, lo cual es sólo para Linux. Esta característica no estaba mantenida al tiempo de la escritura de este libro, así que el equipo de Samba creó una característica adicional y proporcionó smbwrapper en su lugar. La característica smbwrapper funciona en mayor número de plataformas Unix que *smbmount*, así que normalmente querrás usar *–with-smbwrapper* en lugar de esta opción.

–with-pam Incluye soporte para para *Pluggable Authentication Modules* (PAM), una característica común de autentificación en el s.o. Linux.

–with-ldap Incluye soporte para el *Lightweight Directory Access Protocol* (LDAP). Una versión futura de LDAP será usada en el s.o. Windows 2000 (NT 5.0); este soporte de Samba es experimental. LDAP es un flexible protocolo de directorios cliente-servidor que proporciona información tal como certificados y miembros de grupos[3].

–with-nis Incluye soporte para obtener información password-fichero desde NIS (las páginas amarillas de la red).

–with-nisplus Incluye soporte para obtener información password-fichero desde NIS+, el sucesor de NIS.

–with-ssl Incluye soporte para *Secure Sockets Layer* (SSL), el cual es usado para proporcionar conexiones encriptadas desde cliente al servidor. El Apéndice A, Configurando Samba con SSL, describe la configuración de Samba con soporte SSL.

–with-nisplus-home Incluye soporte para localizar qué servidores contienen un determinado directorio home de usuario y pedirle que conecte a él. Requiere *–with-nis* y, usualmente, *–with-automounter*.

–with-mmap Incluye experimental *memory mapping code*. Esto no es necesario para bloqueo rápido, el cual ya usa *mmap* o *System V shared memory*.

–with-syslog Incluye soporte para usar la utilidad SYSLOG para información de registro generada por el servidor Samba. Hay un par de opciones de configuración de Samba que puedes usar para activar el soporte SYSLOG; El Capítulo 4, Discos Compartidos, las discute.

–with-netatalk Incluye soporte experimental para interoperar con el servidor de ficheros Netatalk (Macintosh).

–with-quotas Incluye soporte de cuotas de disco.

Como cada una de estas opciones están desactivadas por defecto, ninguna de ellas es esencial para Samba.

En adición, la Tabla 2.1 muestra algunos de los otros parámetros que puedes usar con el script configure si quieres ubicar partes de la distribución de Samba en diferentes lugares, quizás para hacer uso de múltiples discos o particiones. Nota que los valores por defecto de la tabla muchas veces se refieren a un prefijo especificado previamente.

[3] Por directorio, no queremos indicar un directorio en un sistema de ficheros, sino un directorio indexado (tal como una guía de teléfonos). La información es almacenada y puede ser fácilmente retornada en un sistema público LDAP.

De nuevo, antes de ejecutar el script configure, es importante que seas el usuario root del sistema. De lo contrario, podrías obtener un warning como este:

```
configure: warning: running as non-root will disable some tests
```

No querrás que ningún test sea desactivado cuando se cree el *makefile* de Samba; esto amplía el potencial de errores al tiempo de compilación o ejecución de Samba en tu sistema.

Aquí tienes una simple ejecución del script *configure*, el cual crea un *makefile* para Samba 2.0.4 para la plataforma Linux. Nota que debes ejecutar el script en el directorio *source*, y que varias líneas de la mitad de la salida han sido omitidas:

```
# cd samba-2.0.4b/source/
# ./configure | tee mylog
loading cache ./config.cache
checking for gcc... (cached) gcc
checking whether the C compiler (gcc -O ) works... yes
checking whether the C compiler (gcc -O ) is a cross-compiler... no
checking whether we are using GNU C... (cached) yes
checking whether gcc accepts -g... (cached) yes
checking for a BSD compatible install... (cached) /usr/bin/install -c
...(contenido omitido)...
checking configure summary
configure OK
creating ./config.status
creating include/stamp-h
creating Makefile
creating include/config.h
```

En general, cualquier mensaje desde configure que no sea antecedido por las palabras *checking* o *creating* es un error; frecuentemente te ayudará redirigir la salida del script a un fichero para localizar errores, como hicimos con el comando *tee* anteriormente. Si se produce un error durante la compilación, información más detallada sobre él se puede encontrar en el fichero *config.log*, el cual se genera en el directorio local mediante el script *configure*.

Si todo va bien, obtendrás un mensaje *checking configure summary* seguido de un mensaje *configure OK* y 4 ó 5 mensajes de creación de ficheros. Así que, si ha ido bien … El Siguiente Paso: compilar.

2.3. Compillando e Instalando Samba.

Llegados a este punto deberías estar preparado para construir los ejecutables de Samba. Compilar es fácil: en el directorio source, teclea make sobre la línea de comandos. La utilidad make producirá una salida de mensajes explicativos y de sucesos, comenzando por:

```
# make
Using FLAGS = -O -Iinclude -I./include -I./ubiqx -I./smbwrapper
-DSMBLOGFILE="/usr/local/samba/var/log.smb"
-DNMBLOGFILE="/usr/local/samba/var/log.nmb"
-DCONFIGFILE="/usr/local/samba/lib/smb.conf"
-DLMHOSTSFILE="/usr/local/samba/lib/lmhosts"
-DSWATDIR="/usr/local/samba/swat"
-DSBINDIR="/usr/local/samba/bin"
-DLOCKDIR="/usr/local/samba/var/locks"
-DSMBRUN="/usr/local/samba/bin/smbrun"
-DCODEPAGEDIR="/usr/local/samba/lib/codepages"
-DDRIVERFILE="/usr/local/samba/lib/printers.def"
-DBINDIR="/usr/local/samba/bin"
-DHAVE_INCLUDES_H
-DPASSWD_PROGRAM="/bin/passwd"
-DSMB_PASSWD_FILE="/usr/local/samba/private/smbpasswd"
Using FLAGS32 = -O -Iinclude -I./include -I./ubiqx -I./smbwrapper
-DSMBLOGFILE="/usr/local/samba/var/log.smb"
-DNMBLOGFILE="/usr/local/samba/var/log.nmb"
-DCONFIGFILE="/usr/local/samba/lib/smb.conf"
-DLMHOSTSFILE="/usr/local/samba/lib/lmhosts"
-DSWATDIR="/usr/local/samba/swat"
-DSBINDIR="/usr/local/samba/bin"
```

```
-DLOCKDIR="/usr/local/samba/var/locks"
-DSMBRUN="/usr/local/samba/bin/smbrun"
-DCODEPAGEDIR="/usr/local/samba/lib/codepages"
-DDRIVERFILE="/usr/local/samba/lib/printers.def"
-DBINDIR="/usr/local/samba/bin"
-DHAVE_INCLUDES_H
-DPASSWD_PROGRAM="/bin/passwd"
-DSMB_PASSWD_FILE="/usr/local/samba/private/smbpasswd"
 Using LIBS = -lreadline -ldl -lcrypt -lpam
Compiling smbd/server.c
Compiling smbd/files.c
Compiling smbd/chgpasswd.c
...(contenido omitido)...
Compiling rpcclient/cmd_samr.c
Compiling rpcclient/cmd_reg.c
Compiling rpcclient/cmd_srvsvc.c
Compiling rpcclient/cmd_netlogon.c
Linking bin/rpcclient Compiling utils/smbpasswd.c
Linking bin/smbpasswd
Compiling utils/make_smbcodepage.c
Linking bin/make_smbcodepage
Compiling utils/nmblookup.c
Linking bin/nmblookup
Compiling utils/make_printerdef.c
Linking bin/make_printerdef
```

Si te encuentras con problemas durante la compilación, comprueba la documentación de Samba para ver si el problema tiene fácil solución. Otra posibilidad es buscar o mandar una pregunta a las listas de distribución de Samba, las cuales vienen al final del appd-34717, o en la página principal del web de Samba. La mayoría de las cuestiones sobre la compilación son específicas del sistema, y casi siempre fáciles de superar.

Ahora que los ficheros han sido compilados, puedes instalarlos en los directorios que tú hayas especificado, con el comando:

```
#
make install
```

Si estás actualizando tu versión de Samba, tus viejos archivos de serán salvados con la extensión .old , y puedes reinstalar la versión antigua con el comando make revert. Tras realizar un make install, deberías copiar los archivos .old (si existen) a una nueva localización o renombrarlos. Si no lo haces, la próxima vez que compiles Samba, los originales .old serán sobreescritos sin previo aviso, y perderás tu versión primaria. Si configuras Samba para usar las ubicaciones por defecto, los nuevos ficheros serán instalados en los directorios listados en la Tabla 2.2. Recuerda que necesitas realizar la instalación desde una cuenta que tenga privilegios de escritura sobre estos directorios; normalmente usarás la cuenta de root.

A lo largo del resto del libro, ocasionalmente nos referiremos a la localización de la raíz de la estructura de directorios como *samba_dir*. En la mayoría de configuraciones, este es el directorio base del paquete Samba: */usr/local/samba*.

> **AVISO**: Cuidado si has hecho */usr* una partición de sólo lectura. Querrás poner ficheros de registro, bloqueo, y de contraseñas en algún sitio.

Aquí está la instalación que hemos hecho en nuestra máquina. Puedes ver que hemos usado */usr/local/samba* como el directorio base de la distribución (p.ej., *samba_dir*):

```
#
make install
Using FLAGS = -O -Iinclude -I./include -I./ubiqx -I./smbwrapper -DSMBLOGFILE="/usr/local/samba/var/log.smb"
-DNMBLOGFILE="/usr/local/samba/var/log.nmb"
-DCONFIGFILE="/usr/local/samba/lib/smb.conf" -
...(contenido omitido)...
The binaries are installed. You may restore the old binaries
(if there were any) using the command "make revert". You may
uninstall the binaries using the command "make uninstallbin"
or "make uninstall" to uninstall binaries, man pages and shell
scripts.
...(contenido omitido)...
==========================================================
The SWAT files have been installed. Remember to read the
README for information on enabling and using SWAT.
==========================================================
```

Cuadro 2.1: Opciones de Configuración Adicionales.

Opción	**Significado**	**V. por Defecto**
–prefix=directorio	Instala los archivos independientes de la arquitectura en el directorio base especificado.	/usr/local/samba
–eprefix=directorio	Instala los archivos dependientes de la arquitectura en el directorio base especificado.	/usr/local/samba
–bindir=directorio	Instala ejecutables de usuario en el directorio especificado.	eprefix /bin
–sbindir=directorio	Instala ejecutables de administrador en el directorio especificado.	eprefix /bin
–libexecdir=directorio	Instala programas ejecutables en el directorio especificado.	eprefix /libexec
–datadir=directorio	Instala los datos de sólo lectura independientes de la arquitectura en el directorio especificado.	prefix /share
–libdir=directorio	Instala librerías de programas en el directorio especificado.	eprefix /lib
–includedir=directorio	Instala los paquetes de ficheros include en el directorio especificado.	prefix /include
–infodir=directorio	Instala los ficheros de información adicional en el directorio especificado.	prefix /info
–mandir=directorio	Instala las páginas de manual en el directorio especificado.	prefix /man

Cuadro 2.2: Directorios de Instalación de Samba.

Directorio	**Descripción**
/usr/local/samba	Raíz
/usr/local/samba/bin	Binarios
/usr/local/samba/lib	smb.conf, lmhosts, ficheros de configuración, etc.
/usr/local/samba/man	Documentación de Samba
/usr/local/samba/private	Fichero de passwords encriptadas de Samba
/usr/local/samba/swat	Archivos de SWAT
/usr/local/samba/var	Ficheros de registro de Samba, de bloqueo, información sobre listas de navegación, ficheros de memoria compartida, ficheros PID.

El último mensaje es sobre SWAT, ya has configurado todos los ficheros. ¡Felicidades! ¡Ahora tienes Samba en tu sistema!

2.3.1. Pasos Finales de la Instalación.

Hay un par de pasos finales a realizar. Específicamente, añadir la Herramient de Administración Web de Samba (SWAT) a los ficheros de configuración */etc/services* y */etc/inetd.conf*. SWAT funciona como demonio bajo *inetd* y proporciona un editor basado en forms en tu navegador web para la creación y/o modificación de ficheros de configuración SMB.

Para añadir SWAT, añade la siguiente línea al final del fichero */etc/services*:

```
swat 901/tcp
```

1. Añade estas líneas a */etc/inetd.conf*. (Verifica la página de manual de tu inetd.conf para ver el formato exacto del fichero inetd.conf, por si difiere con el ejemplo expuesto aquí). No olvides cambiar la ruta al binario SWAT si lo has instalado en una ubicación distinta a la que genera la instalación por defecto en */usr/local/samba*.

```
swat stream tcp nowait.400 root /usr/local/samba/bin/swat swat
```

Y esto es todo en cuanto a la instalación. Pero antes de que puedas empezar a usar Samba, sin embargo, necesitarás crear un fichero de configuración para él.

2.4. Un Fichero de Configuración Básico.

La clave para configurar Samba está en un único fichero de configuración: *smb.conf*. Este fichero de configuración puede ser muy simple o extremadamente complejo, y el resto de este libro se dedica a ayudarte en esta tarea. Por el momento, sin embargo, te mostraremos cómo crear una configuración simple, la cual te permitirá iniciar los demonios Samba y ver que todo está funcionando como debiera. En posteriores capítulos, verás cómo configurar Samba para tareas más complejas e interesantes.

El proceso de instalación no crea automáticamente un fichero de configuración *smb.conf*, aunque se incluyen varios de ejemplo en la distribución de Samba. Para testear el software del servidor, nosotros usaremos el siguiente fichero. Este debería llamarse *smb.conf* y estar ubicado en el directorio */usr/local/samba/lib*[4].

```
[global]
   workgroup = SIMPLE

[test]
   comment = For testing only, please
   path = /export/samba/test
   read only = no
   guest ok = yes
```

[4]Si no compilaste Samba, sino que usaste una distribución de binarios, comprueba la documentación del paquete para ver dóno debes ubicar el fichero *smb.conf*. Si Samba viene preinstalado en tu sistema Unix, probablemente ya existirá un fichero *smb.conf* en algún lugar en tu sistema.

Este breve fichero de configuración le dice al servidor Samba que ofrezca el directorio */export/samba/test* en el servidor como un recurso compartido SMB/CIFS llamado *test*. El servidor también se convierte en parte del grupo de trabajo llamado SIMPLE, donde cada uno de los clientes deben ser también miembros de él. (Usa aquí tu propio grupo de trabajo si ya sabes cuál es). Usaremos el recurso compartido *[*test*]* en el siguiente capítulo para configurar los clientes Windows. Por ahora, puedes completar la configuración ejecutando los siguientes comandos como root en tu servidor Unix:

```
#
mkdir /export/samba/test
#
chmod 777 /export/samba/test
```

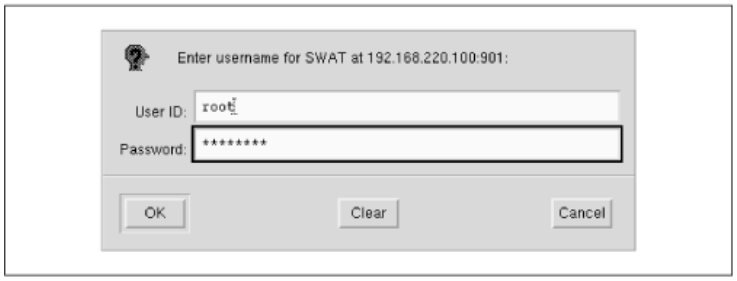
Debemos señalar que, en términos de seguridad, esta es la peor configuración posible. Por el momento, sin embargo, sólo deseamos testear Samba, así que dejaremos la seguridad para otro momento. En adición, hay algunos problemas con la encriptación de contraseñas con las que te encontrarás más tarde en los clientes Windows, así que esta configuración la haremos para que nos dé los menores dolores de cabeza posibles.

Si estás usando Windows 98 o Windows NT Service Pack 3 o anteriores, debes añadir la siguiente entrada a la sección *[global]* del fichero de configuración de Samba: *encrypt passwords = yes*. En adición, debes usar el programa smbpassword (normalmente localizado en */usr/local/samba/bin/*) para reintroducir las combinaciones usuario/contraseña de aquellos usuarios que deberían tener capacidad de acceder a los recursos compartidos. Por ejemplo, si quieres permitir al usuario Unix steve acceder a recursos compartidos desde un cliente SMB, deberías teclear: *smbpassword -a steve*. La primera vez que un usuario es añadido, el programa generará un error indicando que la base de datos de contraseñas encriptadas no existe. No te preocupes, entonces la creará por ti. Asegúrate de que las combinaciones usuario/contraseña que añades a la base de datos de contraseñas encriptadas coinciden con los usuarios y contraseñas de los clientes Windows.

2.4.1. Usando SWAT.

Con Samba 2.0, la creación de un fichero de configuración es más sencilla que escribirlo a mano. Puedes usar tu navegador web para conectar a *http://localhost:901*, y logearte con la cuenta del usuario *root*, como se muestra en la Figura 2.1.

Figura 2.1: SWAT login.



Tras logearte, presiona el botón GLOBALS al principio de la página. Deberías ver la página de Variables Globales que se muestra en la Figura 2.2.

En este ejemplo, establece el campo grupo de trabajo a SIMPLE y el campo *security* a USER. La otra opción que necesitas cambiar es la que determina qué sistema de los que hay en la red es el que resuelve las direcciones NetBIOS; este sistema es denominado servidor WINS. Al principio de la págiona, selecciona la opción *wins support* y ponla a *Yes*, a menos que ya tengas un servidor WINS en tu red. Si lo tienes, pon la dirección IP del servidor WINS en el campo *wins server*. Luego vuelve al principio de la página y pulsa sobre el borón *Commit Changes* para grabar los cambios en el fichero *smb.conf*.

Ahora, presiona el icono *Shares*. Deberías ver una página similar a la de la Figura 2.3. Selecciona *Test* debajo del botón *Choose Share*. Verás la página de Parámetros del Recurso Compartido, como se muestra en la Figura 2.4. Añadiremos un comentario para que nos recuerde que esto es un recurso compartido de pruebas en el fichero *smb.conf*.

Si presionas el botón View, SWAT te muestra el siguiente fichero *smb.conf*:

```
# Samba config file created using SWAT
# from localhost (127.0.0.1)
# Date: 1998/11/27 15:42:40
# Global parameters
workgrp = SIMPLE
[test]
   comment = For testing only, please
   path = /export/samba/test
   read only = no
   guest ok = yes
```

Una vez la configuración está completada, puedes saltarte el paso que viene ahora, ya que la salida de SWAT está a prueba de errores sintácticos.

2.4.2. Testeando el Fichero de Configuración.

Si no usaste SWAT para crear tu fichero de configuración, deberías testearlo para asegurarte de que es sintácticamente correcto. Puede parecere algo tonto ejecutar un programa de testeo contra un archivo de sólo 8 líneas, pero es una buena práctica para futuras comprobaciones de ficheros de configuración más complejos.

El programa testeador, *testparm*, examina un fichero *smb.conf* para la búsqueda de errores sintácticos y reporta cualquier error que encuentre, con una lista de los servicios activos en tu máquina. Un ejemplo; advertirás que hemos “cometido un error” al escribir incorrectamente el nombre del grupo de trabajo, *workgroup* como *workgrp* (la salida es grande, por lo que recomendamos captures las últimas partes con el comando *tee*):

```
Load smb config files from smb.conf
Unknown parameter encountered: "workgrp"
Ignoring unknown parameter "workgrp"
Processing section "[test]"
Loaded services file OK.
Press enter to see a dump of your service definitions
# Global parameters
[global]
```

Figura 2.2: Página de Variables Globales de SWAT.

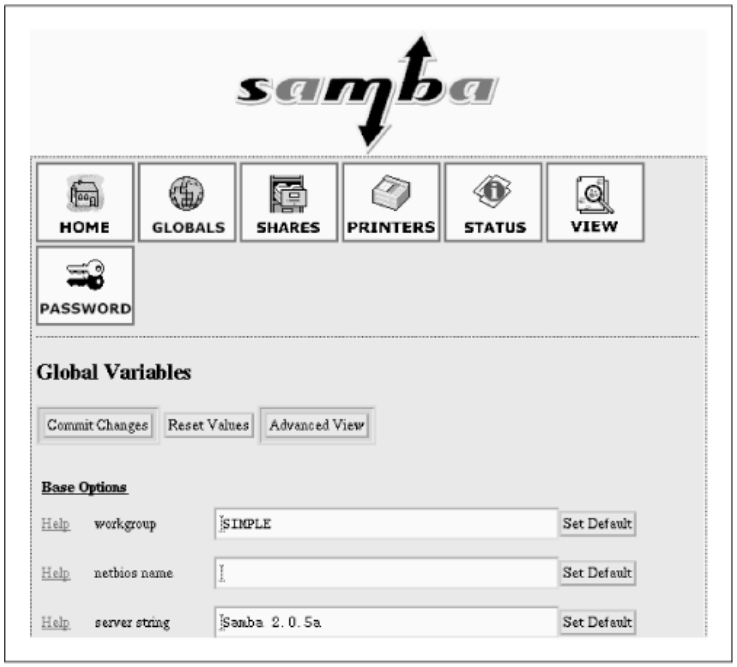
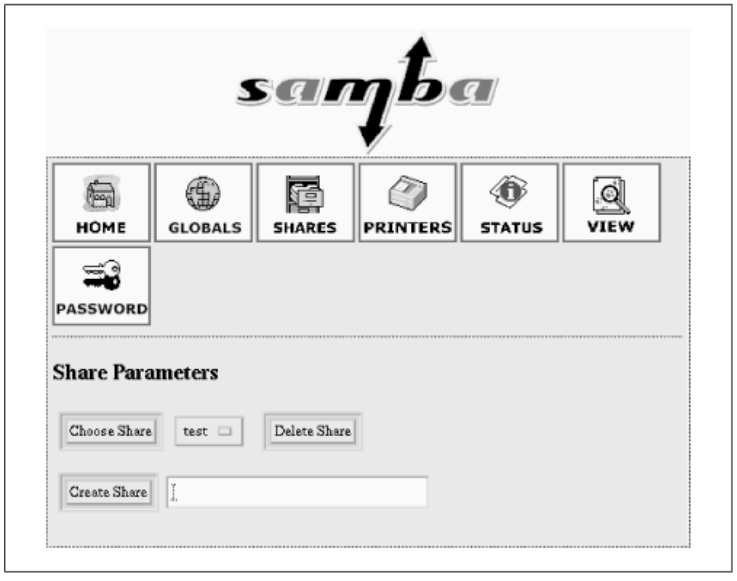


Figura 2.3: Pantalla de Creación de Recursos Compartidos de SWAT.



```
   workgroup = WORKGROUP
   netbios name = netbios
   aliases = server
   string = Samba 2.0.5a
   interfaces = bind
   interfaces only = No


...(contenido omitido)...


[test]
   comment = For testing only, please
   path = /export/samba/test
   read only = No
   guest ok = Yes
```

Las partes interesantes están al principio y al final. El principio de la salida marcará cualesquiera errores sintácticos que hayas cometido, y la del final lista los servicios que el servidor cree debería ofrecer. Una advertencia: asegúrate de que tú y tu servidor tenéis las mismas expectativas.

Si todo parece bien, ¡Entonces ya puedes arrancar los demonios del servidor!

2.5. Iniciando los Demonios de Samba.

Existen dos procesos Samba, smbd y nmbd, que necesitan ser iniciados para que Samba funcione correctamente. Y existen tres maneras de hacer esto:

- A mano. *
- Como demonios que se ejecutan al arrancar el servidor.
- Desde inetd.

2.5.1. Iniciando los Demonios a Mano.

Si te encuentras en la necesidad, puedes iniciar los demonios de Samba a mano. Como *root*, simplemente introduce los siguientes comandos:

```
#
/usr/local/samba/bin/smbd`D
#
/usr/local/samba/bin/nmbd`D
```

En éste punto, Samba estará funcionando en tu sistema y estará preparado para aceptar conexiones.

2.5.2. Demonios Autosuficientes.

Para ejecutar los procesos Samba como demonios autosuficientes, necesitas añadir los comandos listados antes en tus scripts de arranque del sistema. Estos variarán en función de si tienes un sistema Unix basado en BSD o en System V.

2.5.2.1. BSD Unix.

Con un Unix estilo BSD, necesitarás añadir el siguiente código al fichero *rc.local*, el cual está normalmente en los directorios */etc* o */etc/rc.d*:

```
if [ -x /usr/local/samba/bin/smbd]; then
   echo "Starting smbd..."
   /usr/local/samba/bin/smbd -D
   echo "Starting nmbd..."
   /usr/local/samba/bin/nmbd -D
fi
```

Este código es muy simple; chequea para ver si el fichero smbd tiene permisos de ejecución, y si los tiene, arranca cada uno de los demonios Samba al arrancar el sistema.

2.5.2.2. Unix System V.

Con System V, las cosas se pueden complicar un poco más. System V normalmente usa scripts para iniciar y parar los demonios del sistema. Aquí, necesitas instruir a Samba sobre cómo debe operar cuando se arranca y cuando se para. Puedes modificar los contenidos del directorio */etc/rc.local* y añadir algo como esto al siguiente programa, llamado *smb*:

```
#!/bin/sh

# Contains the "killproc" function on Red Hat Linux
./etc/rc.d/init.d/functions

PATH="/usr/local/samba/bin:$PATH"

case $1 in
  'start')
    echo "Starting smbd..."
    smbd -D
    echo "Starting nmbd..."
    nmbd -D ;;
  'stop')
    echo "Stopping smbd and nmbd..."
    killproc smbd
    killproc nmbd
    rm -f /usr/local/samba/var/locks/smbd.pid
    rm -f /usr/local/samba/var/locks/nmbd.pid ;;
*)
```

```
echo "usage: smb {start|stop}" ;;
esac
```

Con este script, puedes arrancar y para el servicio SMB con los siguientes comandos:

```
# /etc/rc.local/smb start
Starting smbd...
Starting nmbd...
# /etc/rc.local/smb stop
Stopping smbd and nmbd...
```

2.5.3. Arrancando desde Inetd.

El demonio *inetd* es un "super demonio". Escucha por los puertos TCP definidos en */etc/services* y ejecuta el programa apropiado para cada puerto, cada uno de los cuales están definidos en */etc/inetd.conf*. La ventaja de esta forma de arranque es que puedes tener un gran número de demonios preparados para atender peticiones, pero no tienen por qué estar arrancados. En su lugar, el demonio inetd escucha su lugar. La pega es un pequeño coste derivado de la creación de un nuevo proceso de demonio, y el hecho de que necesitas editar dos archivos en vez de uno. Esto es útil cuando sólo tienes uno o dos usuarios o tun máquina tiene demasiados demonios. También es más sencilla la actualización sin afectar a una conexión existente.

Si quieres arrancar desde *inetd*, primero abre */etc/services* con tu editor de textos. Si todavía no las tienes definidas, añade las siguientes dos líneas:

```
netbios-ssn 139/tcp
netbios-ns 137/udp
```

A continuación, edita */etc/inetd.conf*. Busca las siguientes dos líneas y añádelas si no existen. Si ya tienes las líneas *smbd* y *nmbd* en el fichero, edítalas para que apunten a los nuevos *smbd* y *nmbd* que tienes instalados. Tu versión de Unix puede usar una sintaxis algo distinta en este fichero; usa las entradas existentes y la página de manual de inetd.conf como guía:

```
netbios-ssn stream tcp nowait root /usr/local/samba/bin/smbd smbd
netbios-ns dgram udp wait root /usr/local/samba/bin/nmbd nmbd
```

Finalmente, mata cualesquiera procesos *smbd* o *nmbd* y envía al proceso inetd una señal de colgar o *"hangup"* (HUP). (El demonio *inetd* relee su fichero de configuración al recibir una señal HUP). Para hacer esto, usa el comando ps para encontrar su ID de proceso, y luego manda la señal con el siguiente comando:

```
#
kill -HUP process_id
```

Tras esto, Samba debería estar arrancado y operativo.

2.6. Testeando los Demonios Samba.

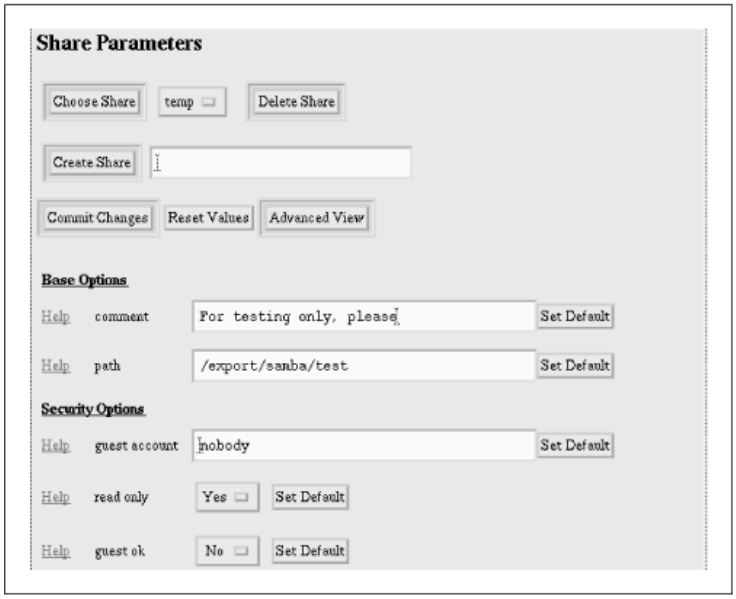
Es difícil de creer, pero ya lo hemos hecho casi todo en cuanto a la configuración del servidor Samba. Y todo lo que queda por hacer es asegurarse de que todo está funcionando como debería. Una forma adecuada de hacer esta comprobación es usar el programa smbclient para examinar qué está ofreciendo el servidor a la red. Si todo se ha configurado correctamente, deberías poder hacer lo siguiente:

```
# smbclient -U% -L localhost
   Added interface ip=192.168.220.100 bcast=192.168.220.255 nmask=255.255.255.0
   Domain=[SIMPLE] OS=[Unix] Server=[Samba 2.0.5a]
   Sharename      ----  Type   Comment
   ---------      ----  ------
   test                        Disk       For testing only, please
   IPC$                     IPC        IPC Service (Samba 2.0.5a)
   Server      --------      Comment
   ---------   --------
   HYDRA               Samba 2.0.5a
   Workgroup         Master
   ---------   -------
   SIMPLE              HYDRA
```

Si hay algún problema, ¡Que no cunda el pánico! Intenta iniciar los demonios manualmente, y chequea la salida del sistema o los ficheros de registro en */usr/local/samba/var/log.smb* para ver si puedes averiguar qué ha pasado. Si piensas que puede ser un problema más serio, pasa al Capítulo 7, Impresión y Resolución de Nombres, para encontrar ayuda en la resolución de problemas con los demonios Samba.

Si todo está funcionando, ¡Felicidades! Ahora tienes configurado el servidor Samba con una compartición de disco. Es una compartición muy simple, pero podemos usarla para configurar y testear los clientes Windows 95/98 y NT (en el siguiente capítulo). Entonces haremos la cosa más interesante añadiendo servicios tales como directorios de usuario (*homes*), impresoras y seguridad, y viendo cómo integrar el servidor en un dominio Windows.

Figura 2.4: Pantalla de Parámetros de Recurso Compartido de SWAT.



Capítulo 3

Configurando los Clientes Windows.

Te gustará saber que la configuración de Windows para usar nuestro nuevo servidor Samba es muy simple. SMB es el lenguaje netivo de Microsoft para la compartición de recursos sobre una red de área local, así que la mayor parte de la configuración de la parte de los clientes Windows ya está hecha. Lo primero que cubriremos en este capítulo tiene que ver con la comunicación y la coordinación entre Windows y Unix, dos sistemas operativos completamente diferentes.

Samba usa TCP/IP para hablar a sus clientes de la red. Si todavía no usas TCP/IP en tus computadoras Windows, este capítulo te mostrará cómo instalarlo. Necesitarás configurar tus máquinas Windows para que puedan operar sobre una red TCP/IP. Una vez estos dos requerimientos hayan sido realizados, podremos mostrarte cómo acceder a los recursos compartidos en el servidor Samba.

Este capítulo se divide en tres secciones. La primera sección cubre la configuración de máquinas Windows 95/98, mientras que el segundo cubre máquinas Windows NT 4.0. La sección final proporciona alguna información sobre cómo son realizadas las conexiones SMB desde los clientes Windows y los servidores, lo cual nos será útil para los siguientes capítulos de este libro.

3.1. Configurando Computadoras Windows 95/98.

Desafortunadamente, Windows 95/98 no fue diseñado para que un PC tuviera más de un usuario; este concepto es más inherente a un sistema opertativo Unix o Windows NT. Sin embargo, Windows 95/98 trae un soporte limitado para múltiples usuarios: si lo configuras, el sistema operativo mantendrá un archivo de parámetros de configuración (profile) y de contraseñas (*.PWL) para cada usuario. Esto está muy lejos de la seguridad de multiusuario. En otras palabras, Windows 95/98 no va a evitar que un usuario pueda destruir el trabajo de otro en el disco duro local (como sí hace Unix), pero los profiles son un comienzo.

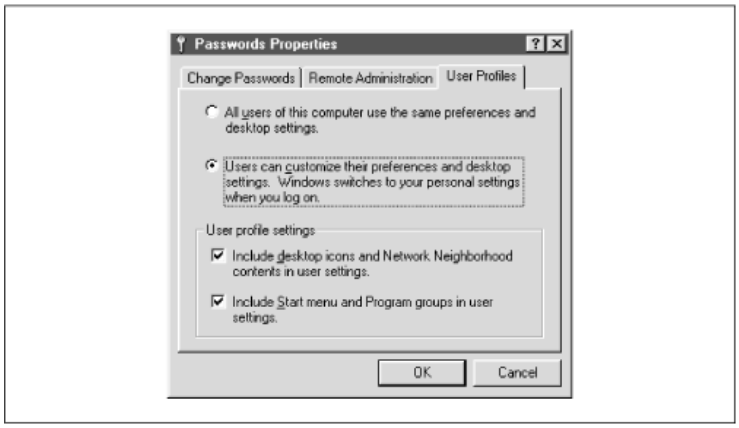
3.1.1. Cuentas y Contraseñas.

Lo primero que necesitamos hacer es decirle a Windows que matenga perfiles (profiles) de usuario separados, y que almacene nombres de usuario y contraseñas para

autentificar a cualquiera que intente un acceso a un recurso compartido por Samba. Lo haremos a través del establecimiento de contraseñas en el Panel de Control. Si no estás familiarizado con el Panel de Control de Windows, puedes acceder a él seleccionando el menú Configuración desde el Botón de Inicio en la esquina inferior derecha del Escritorio de Windows. Alternatvamente, también lo encontrarás como una carpeta bajo el icono que representa a tu computadora (Mi PC).

Una vez seleccionado el icono "Contraseñas" en el Panel de Control, hac click sobre la pestaña "Perfiles de Usuario". Deberías ver la caja de diálogo que se muestra en la Figura 3.1. Entonces pulsa sobre el segundo de los dos botones de radio, el que comienza por "Los usuarios pueden personalizar sus preferencias...". Esto causa que Windows almacene un perfil separado para cada usuario, y almacena el nombre de usuario y la contraseña que proporciones, la cual se usará más tarde cuando se conecte a un servidor SMB/CIFS. Finalmente, selecciona las dos opciones bajo el botón, como se muestra en la figura.

Figura 3.1: El Panel de Propiedades de Contraseñas.



El siguiente paso es seleccionar la pestaña "Cambiar Contraseñas". Para que Samba te pueda permitir acceder a sus recursos compartidos, el nombre de usuario y la contraseña que proporciones en Windows debe coincidir con el que está en el lado del servidor Samba. Si no tienes esta pestaña en tu cuadro de diálogo no te preocupes; será probablemente debido a que todavía no te has asignado un nombre de usuario y contraseña en Windows. Simplemente haz click sobre el botón OK y responde "Sí" cuando Windows solicite reiniciar. Luego pasa a la sección titulada "*Sección 3.1.1.2, Logeándote por primera vez*".

3.1.1.1. Cambiando la Contraseña de Windows.

Tras seleccionar pestaña "Cambiar Contraseñas", aparecerá la caja de diálogo de la Figura 3.2.

Selecciona el botón "Cambiar la Contraseña de Windows". Aparecerá la caja de diálogo "Cambiar la Contraseña de Windows", como muestra la Figura 3.3. Desde aquí, puedes cambiar tu contraseña para que coincida con la contraseña de la cuenta en el servidor Samba a través del cual intentas logearte.

3.1.1.2. Logeándote por Primera Vez.

Si no tienes una pestaña "Cambiar Contraseñas" en la ventana "Propiedades de Contraseñas", entoncces una vez que Windows se haya reiniciado, te solicitará te logees con un nombre de usuario y una contraseña. Usa el nombre de usuario y la contraseña que tengas asignada en el servidor Samba. Una vez confirmes tu nuevo nombre de usuario y contraseña, o si ya tenías una, Windows te preguntará si quieres tener un perfi, usando la caja de diálogo que ves en la Figura 3.4.

Responde "Sí", y Windows creará un perfil separado y un fichero de contraseñas para ti y almacenará una copia de la contraseña en el fichero. Ahora cuando conectes a Samba, Windows enviará su contraseña, la cual será usada para autentificarte contra cada recurso compartido. No nos ocuparemos de los perfiles por el momento; los trataremos en el Capítulo 6, Usuarios, Seguridad y Dominios. Pero apuntaremos, sin embargo, que existe un pequeño riesgo en cuanto a la seguridad: alguien podría robar el fichero de contraseñas y desencriptar las claves, ya que la encriptación es muy simple. Desafortunadamente, no hay solución al problema en Windows 95/98. En Windows 2000 (NT 5.0), la encriptación del fichero de contraseñas se reemplazará con un algoritmo mucho más complejo.

3.1.2. Configurando la Red.

La siguiente cosa que necesitamos hacer es asegurarnos de tener el protocolo TCP/IP configurado correctamente. Para ello, haz doble click sobre el icono de Red en el Panel de Control. Deberías ver la caja de diálogo de configuración de red, como muestra la Figura 3.5.

Las redes Microsoft trabajan con protocolos específicos, como IPX o TCP/IP, para un dispositivo físico específico, tal como una tarjeta Ethernet om una conexión telefónica. Mediante el rutado de un protocolo a través de un dispositivo físico, la máquina puede actuar como cliente o servidor para un particular tipo de red. Para Samba, nosotros estamos interesados en usar el protocolo TCP/IP a través un dispositivo de red, convirtiendo a la máquina en un cliente para redes Microsoft. Así, cuando la caja de diálogo aparezca, deberías ver al menos el componente "Cliente para Redes Microsoft" instalado en la máquina, y con suerte un dispositivo de red (preferiblemente una tarjeta Ethernet) asociado al protocolo TCP/IP. Si sólo hay un dispositivo de red, verás el protocolo TCP/IP listado bajo dicho dispositivo. Aparecerá como en la Figura 3.5., donde el protocolo está debajo del dispositivo.

También puedes ver "Compartir Impresoras y Archivos para Redes Microsoft", el cual es útil. En adición, deberías ver Redes NetBEUI o Novell, los cuales vienen por defecto con las instalaciones Windows pero no son necesarios cuando TCP/IP está funcionando. Elimina NetBEUI si puedes -es innecesario y dificulta la depuración de la visualización de Windows-. Si no tienes servidores Novell en tu red, puedes eliminar Novell (IPX/SPX) también.

3.1.2.1. Añadir TCP/IP.

Si no ves listado TCP/IP, necesitarás instalar el protocolo. Si ya tienes TCP/IP, salta esta sección, y continúa con la sección 3.1.3, "Configurando tu Nombre y Grupo de Trabajo", más adelante en este capítulo.

La instalación de TCP/IP no es difícil, ya que Microsoft distribuye su propia versión de TCP/IP para su instalación gratuita en su CD-ROM de instalación. Puedes añadir el

Figura 3.2: La Pestaña 'Cambiar Contraseñas'.

Figura 3.3: La Caja de Diálogo 'Cambiar la Contraseña de Windows'.

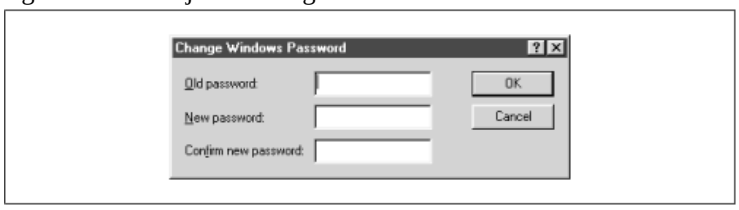
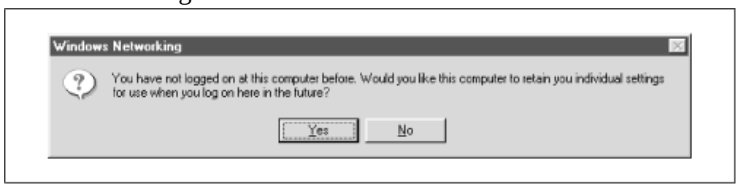


Figura 3.4: Perfiles de Red de Windows.



protocolo haciendo click sobre el botón "Añadir" en la ventana de componentes. Indica que quieres añadir un protocolo específico seleccionando "Protocolo" y pulsando "Añadir..." en la siguiente caja de diálogo, la cual debería ser parecida a la que puedes ver en la Figura 3.6.

Una vez hecho, selecciona el protocolo TCP/IP del fabricante Microsoft, como se muestra en la Figura 3.7, y pulsa OK. Una vez hecho, serás devuelto a la caja de diálogo de Red. Pulsa OK ahí para cerrar la caja de diálogo, así Windows instalará los componentes necesarios desde el disco y reiniciará la máquina.

3.1.2.2. Configurando TCP/IP.

Si tienes más de un dispositivo de red (por ejemplo, una tarjeta Ethernet y un acceso telefónico via modem), cada dispositivo hardware apropiado debería ser "enlazado" al protocolo TCP/IP con una flecha, como se muestra en la Figura 3.8. Selecciona el protocolo TCP/IP enlazado al dispositivo de red que accederá a la red Samba. Cuando esté resaltado, pusla el botón "Propiedades".

Tras hacer esto, el Panel de Propiedades de TCP/IP para ese dispositivo será displayado, como muestra la Figura 3.9.

Hay siete pestañas en este panel, y necesitarás configurar cuatro de ellas:

- Dirección IP.
- Configuración DNS.
- Configuración WINS.
- Enlaces.

3.1.2.3. Pestaña de Dirección IP.

La pestaña de Direcció IP se muestra en la Figura 3.9. Presiona el radio botón "Especificar una Dirección IP" e introduce la dirección del cliente y la máscara de subred en el espacio adecuado. Tú o el administrador de tu red deberías seleccionar una dirección para la máquina. Los valores deberían ubicar a la máquina en la misma subred que el servidor Samba. Por ejemplo, si la dirección del servidor es 192.168.236.86, y su máscara de red 255.255.255.0, deberías usar la dirección 192.168.236.10 (si está disponible) para la computadora Windows 98, con la misma máscara de red que el servidor. Si usas DHCP en tu red para proporcionar direcciones IP a las máquinas Windows, selecciona el botón "Obtener una dirección IP automáticamente".

3.1.2.4. Pestaña de Configuración DNS.

El "Domain Name Service" (DNS) es responsable de trasladar nombres de computadoras de Internet como hobbes.example.com hacia direcciones de máquinas IP tales como 192.168.236.10. Hay dos maneras de usar esto en una máquina Windows 98: puedes especificar un servidor para realizar la traslación por ti, o puedes mantener una lista local de pares nombre/dirección IP.

Las redes que están conectadas a Internet normalmente usan un servidor, ya que los ficheros de máquinas (hosts) requeridos podrían ser enormes. Para una red local sin salida a Internet, la lista de máquinas posibles es pequeña y bien conocida, y podría ser mantenida en una máquina Unix en el archivo /etc/hosts. Si tienes dudas sobre cómo usar un servidor DNS, o qué dirección IP deberías usar, mira el fichero /etc/resolv.conf

Figura 3.5: El Panel de Red de Windows 95/98.

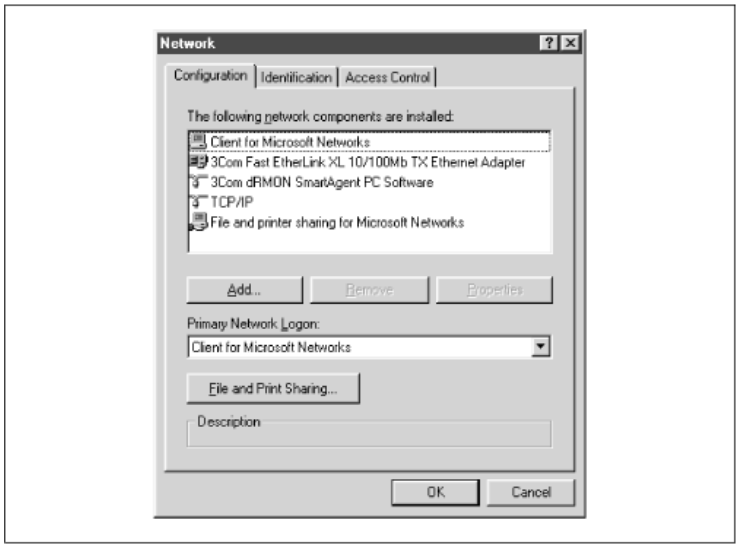


Figura 3.6: Seleccionando un Protocolo para su Instalación.

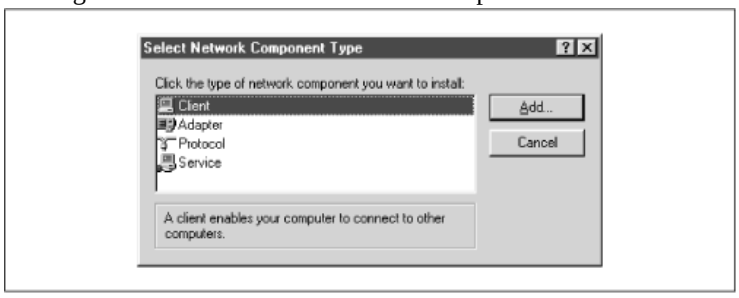


Figura 3.7: Seleccionando el Protocolo a Instalar.

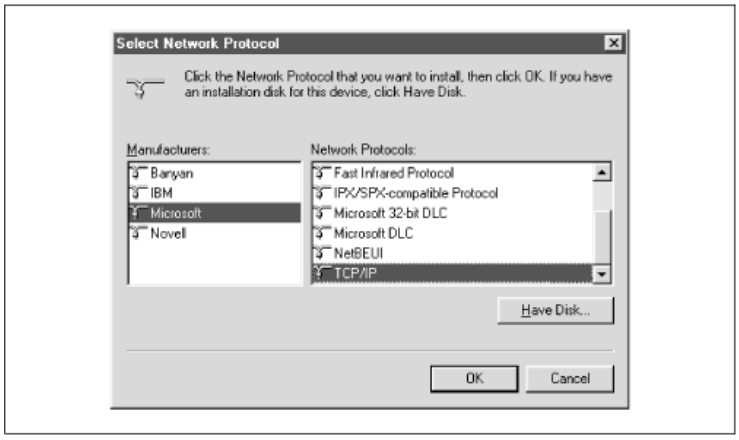


Figura 3.8: Seleccionando el Protocolo TCP/IP Correcto.

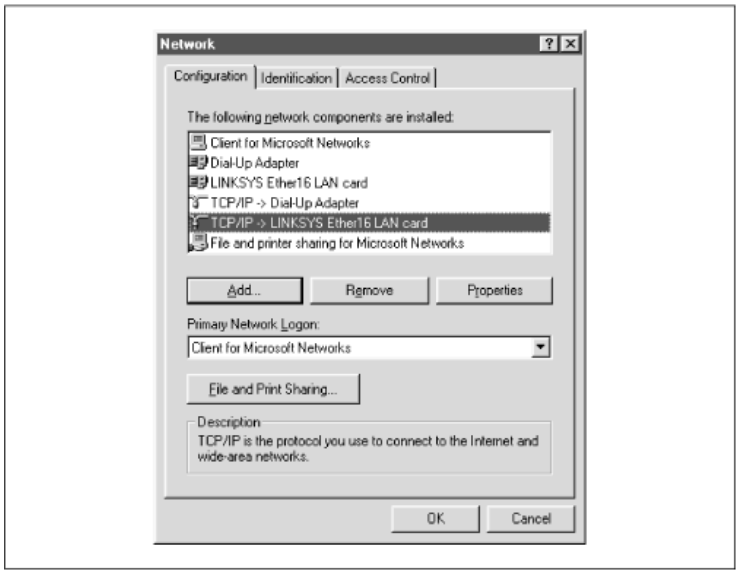
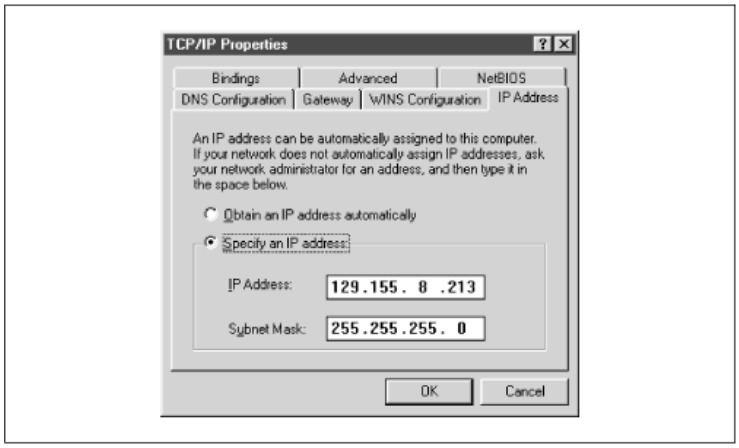


Figura 3.9: Panel de Propiedades de TCP/IP.



de tu servidor Unix. Cualquier máquina que use DNS tendrá este fichero, que sería como este:

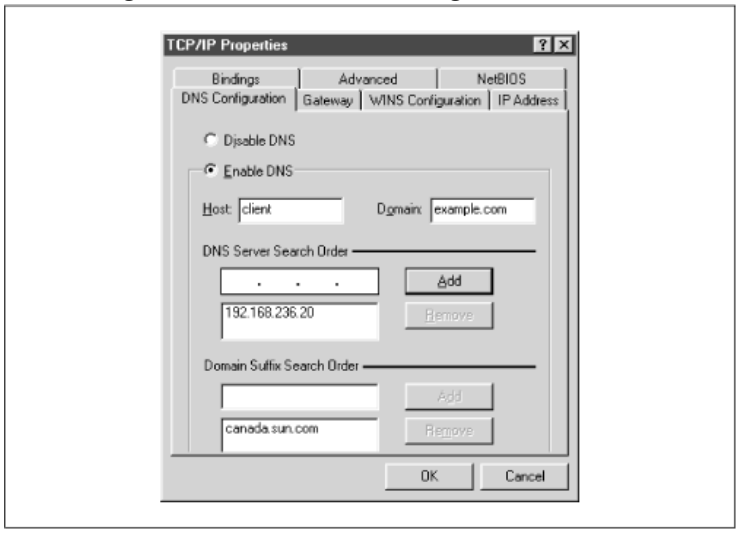
```
#resolv.conf
domain example.com
nameserver 127.0.0.1
nameserver 192.168.236.20
```

En el ejemplo anterior, la segunda línea nameserver en la lista contiene la dirección IP de otra máquina en la red local: 192.168.236.20. Es un buen candidato para convertirse en un servidor DNS[1].

Debes teclear la dirección IP correcta de uno o más servidores DNS (nota que no puedes usar su nombre de Internet, tal como dns.oreilly.com) en el campo apropiado en la Figura 3.10. Asegúrate de no usar 127.0.0.1.

Intenta seleccionar direcciones de tu propia red local. Cualesquiera nombres de servidores listados en /etc/resolv.conf serviría, pero obtendrás mejor rendimiento usando un servidor cercano. (Si no encuentras el fichero /etc/resolv.conf en tus máquinas Unix, desactiva el DNS hasta que encuentres la dirección de al menos un servidor DNS). Asumiremos que sólo tienes un servidor DNS, y que su dirección es 192.168.236.20. Haz click sobre el radio botón "Activar DNS", como se muestra en la Figura 3.10, y añade la dirección del servidor DNS.

Figura 3.10: La Pestaña de Configuración de DNS.



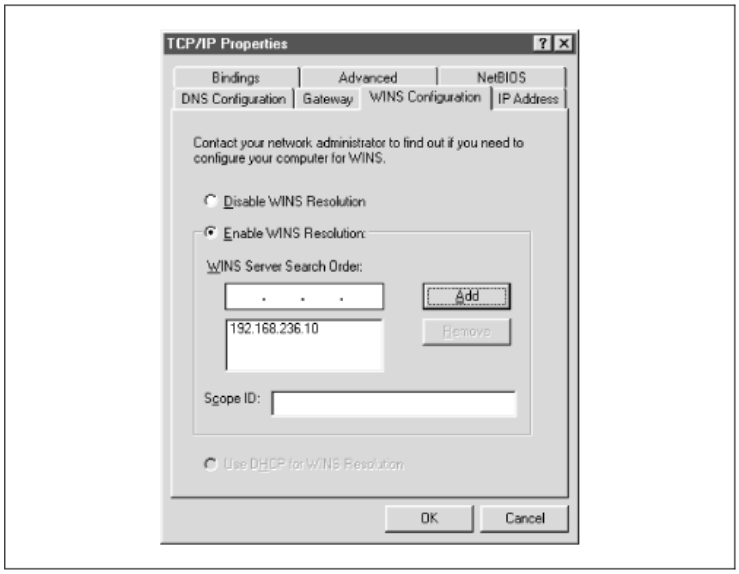
Además, proporciona el nombre de la máquina Windows 95/98 y el dominio de Internet en la que estás. Puedes ignorar el campo "Orden de Búsqueda del Sufijo de Dominio" ("Domain Suffix Search Order") para todo lo que tenga que ver con Samba.

[1] Podemos inhabilitar la otra direcciónya que cada máquina Unix tiene una dirección localhost de 127.0.0.1, ya esté conectada o no a una red. Esta dirección es requerida para el correcto funcionamiento de algunas herramientas del sistema.

3.1.2.5. Pestaña de Configuración WINS.

WINS es el Servicio de Nombres de Internet de Windows, su versión de un servidor de nombres NetBIOS. Si has activado WINS en Samba, debes indicarle a Windows la dirección del servidor Samba. Si estás usando servidores WINS que están en máquinas Windows NT, introduce cada una de ellas. La caja de diálogo que aparece tras seleccionar la pestaña "Configuración WINS" (pestaña "WINS Configuration") se muestra en la Figura 3.11.

Figura 3.11: La Pestaña de Configuración de WINS.



ADVERTENCIA: No mezcles un servidor WINS Samba y un servidor Windows NT server como par primario/seguridad en la caja de diálogo de WINS. Debido a que estos dos no pueden replciar sus bases de datos, esto derivará en que la resolución de nombres se realice incorrectamente.

Aquí, selecciona "Activar Resolución WINS" ("Enable WINS Resolution") e introduce la dirección del servidor WINS en el espacio proporcionado, y luego pulsa "Añadir" ("Add"). No introduzcas nada en el campo "Id de Ámbito" ("Scope ID").

3.1.2.6. Ficheros Hosts.

Si no tienes ni DNS ni WINS, y no quieres usar resolución broadcast, necesitarás proporcionar una tabla de direcciones IP y nombres de máquinas (hosts), en el formato estándar del archivo Unix /etc/hosts. Sobre una máquina Windows, esto se hace en el fichero \WINDOWS\HOSTS del disco donde tengas instalado Windows (normalmente C:\). Un ejemplo de fichero host:

```
# 127.0.0.1    localhost
192.168.236.1  escrime.example.com  escrime
192.168.236.2  riposte.example.com  riposte
192.168.236.3  wizzin.example.com   wizzin
```

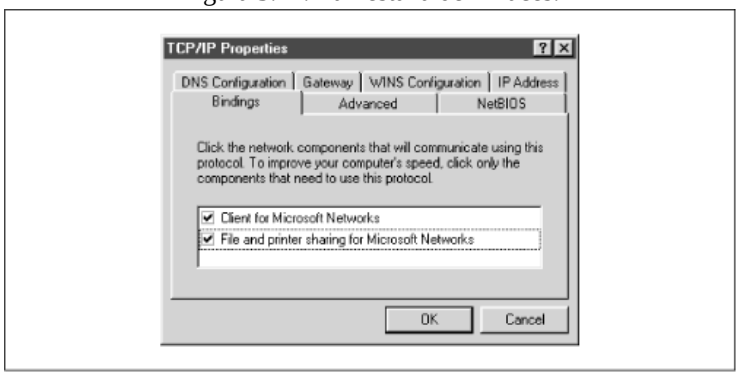
```
192.168.236.4  touche.example.com   touche
192.168.236.10 hobbes.example.com   hobbes
```

Puedes copiar este fichero directamente desde cualquiera de tus máquinas Unix; el formato es idéntico. Sin embargo, sólo deberías usar ficheros hosts en Windows como último recurso para la resolución de nombres.

3.1.2.7. Comprobar los Enlaces.

La última pestaña a comprobar es "Enlaces" ("Bindings"), como se muestra en la Figura 3.12.

Figura 3.12: La Pestaña de Enlaces.



Marca las dos opciones "Cliente para Redes Microsoft" ("Client for Microsoft Networks"). Y si tienes también "Compartir Impresoras y Archivos para Redes Microsoft" ("File and printer sharing for Microsoft Networks") en la caja de diálogo, también deberías marcarla, como se ve en la figura.

3.1.3. Estableciendo tu Nombre y Grupo de Trabajo.

Finalmente, presiona el botón OK en el panel de "Configuración TCP/IP", y regresarás a la pantalla de Configuración de la Red. Entonces selecciona la pestaña "Identificación" ("Identification"), la cual te llevará a la caja de diálogo que ves en la Figura 3.13.

Aquí, por segunda vez, pon el nombre de tu máquina. Esta vez, en vez del nombre de máquina DNS y dominio, estás estableciendo el nombre NetBIOS. Sin embargo, lo mejor es que los dos nombres sean idénticos. Intenta no cometer un típico error: puede resultar muy confuso configurar una máquina si TCP piensa que se llama fred y SMB piensa que se llama ferd.

Establece aquí también el nombre de tu grupo de trabajo. En nuestro caso, ese nombre es SIMPLE, pero si usaste otro en el Capítulo 2, Instalando Samba sobre un Sistema Unix, cuando creamos el fichero de configuración de Samba, usa ese mismo aquí también. Evita llamarlo WORKGROUP o tendrás el mismo nombre de grupo que todas las máquinas no configuradas del mundo.

3.1.4. Accediendo al Servidor Samba.

Pulsa sobre el botón OK para completar la configuración; necesitarás reiniciar para que los cambios tengan efecto.

Ahora llegó el gran momento. Tu servidor Samba está funcionando, y has configurado tus máquinas Windows 95/98 para comunicar con él. Tras el reinicio, pica en Entorno de Red. Deberías ver a tu servidor Samba listado como un miembro más del grupo de trabajo, como se muestra en la Figura 3.14.

Haciendo doble click sobre el servidor se te mostrarán los recursos que este ofrece a la red, como ves en la Figura 3.15 (en este caso, una impresora y el directorio test).

> **ADVERTENCIA**: Si al picar sobre el servidor te ha aparecido inmediatamente una ventana pidiendo la contraseña para el usuario IPC$, significa que Samba no aceptó la contraseña que se envió desde el cliente. En este caso, el nombre de usuario y la contraseña que fueron creadas en el cliente deben coincidir con la combinación usuario/contraseña del servidor Samba. Si estás usando Windows 98 o Windows NT Service Pack 3 o inferior, esto se probablemente debido a que el cliente está enviando claves encriptadas en lugar de contraseñas en formato de texto plano. Puedes remediar la situación realizando dos pasos en el servidor Samba. Primero, añade la siguiente entrada en la sección [global] de tu fichero de configuración de Samba: encrypt password=yes. Segundo, busca el programa smbpasswd en el servidor samba (está en /usr/local/samba/bin por defecto) y úsalo para añadir una entrada a la base de datos de contraseñas encriptadas de Samba. Por ejemplo, para añadir al usuario steve a la base de datos de contraseñas encriptadas de Samba, teclea smbpasswd -a steve. La primera vez que introduzcas la contraseña, el programa lanzará un mensaje de error indicando que la base de datos de contraseñas no existe; creará entonces la base de datos, la cual se ubicará por defecto en /usr/local/samba/private/smbpasswd.

Si no te aparece listado el servidor, inicia el Explorador de Windows y selecciona "Conectar a Unidad de Red" ("Map Network Drive") desde el menú "Herramientas". Esto se mostrará una caja de diálogo en la cual puedes introducir el nombre del servidor Samba y el recurso compartido test en formato Windows *UNC:* \\ *server* *test*, como ya vimos en el primer capítulo. Esto debería intentar contactar con el servidor Samba y su recurso compartido. Si tampoco funciona, vete al Capítulo 9, Resolviendo Problemas con Samba, para buscar ayuda en la resolución del problema.

3.2. Configurando Computadoras Windows NT 4.0.

> **NOTA DEL TRADUCTOR**: Considero más importante avanzar en la traducción de los capítulos relativos a Samba y los clientes Windows 95/98, por lo que la traducción de éste capítulo la dejaré para más adelante.

3.3. Una Introducción a SMB/CIFS.

Vamos a convertir este capítulo en un pequeño tutorial sobre SMB/CIFS. SMB/CIFS es el protocolo que las máquinas Windows 95/98 y NT usan

Figura 3.13: La Pestaña de Identificación.

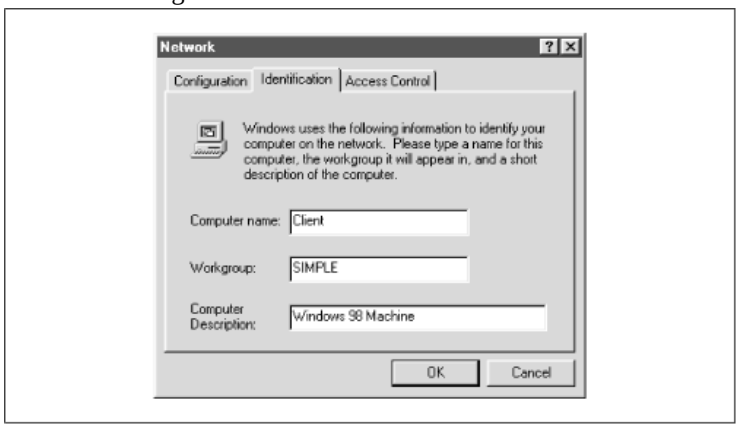


Figura 3.14: Entorno de Red de Windows.

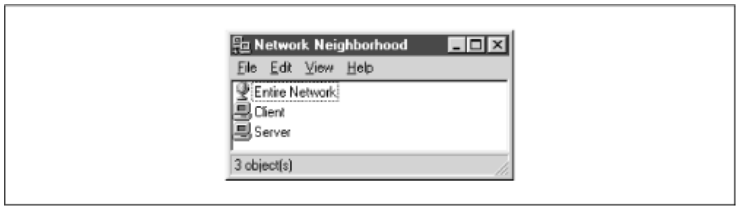
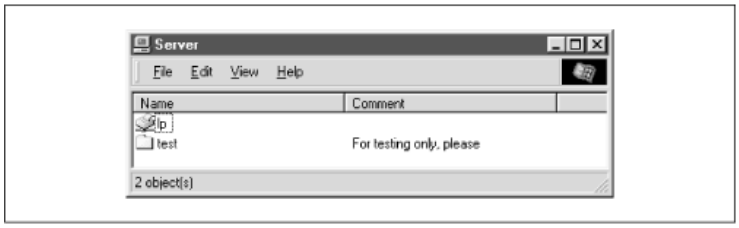


Figura 3.15: Recursos Compartido en el Servidor.



para comunicarse con el servidor Samba y entre ellos. A alto nivel, la suite del protocolo SMB es relativamente simple. Incluye comandos para todas las operaciones de archivos e impresión que puedas necesitar hacer sobre un disco o impresora local, como:

- Abrir y cerrar un fichero.
- Crear y eliminar ficheros y directorios.
- Leer y escribir sobre un fichero.
- Buscar archivos.
- Enviar y eliminar archivos a una cola de impresora.

Cada una de estas operaciones puede ser codificada en un mensaje SMB y transmitida hacia/desde un servidor. El nombre de SMB viene de su formato de datos: son versiones de las estructuras estandar de las llamadas al sistema DOS, o Server Message Blocks, rediseñadas para transmitir a otra máquina a través de una red.

3.3.1. Formato SMB.

Richard Sharpe, del equipo de desarrollo de Samba, define SMB como un protocolo "petición-respuesta"[2]. En efecto, esto significa que un cliente envía una petición SMB a un servidor, y el servidor envía una respuesta SMB de vuelta al cliente. Raramente un servidor envía un mensaje que no es respuesta a la petición de un cliente.

Un mensaje SMB no es tan complejo como puedas pensar. Echemos un vistazo a la estructura interna de uno de estos mensajes. Este puede dividirse en dos partes: la cabecera (header), la cual es de tamaño fijo, y la cadena de comando (command string), cuyo tamaño puede variar en base al contenido del mensaje.

3.3.1.1. Formato de la Cabecera SMB.

La Tabla 3.1. muestra el formato de una cabecera SMB. Los comandos SMB no necesitan usar todos los campos de la cabecera SMB. Por ejemplo, cuando un cliente inicialmente intenta conectar a un servidor, este todavía no tiene un valor de identificador de árbol ("tree identifier") (TID) -se asigna uno cuando se realiza una conexión con éxito-, así que se emplaza un TID nulo (0xFFFF) en el campo de la cabecera. Otros campos pueden ser rellenados con ceros cuando no se usan.

Los campos de la cabecera SMB se listan en la Tabla 3.1.

3.3.1.2. Formato de Comando SMB.

Inmediatamente después de la cabecera va un número variable de bytes que forman un comando o respuesta SMB. Cada comando, tal como "Open File" (identificador de campo COM: SMBopen) o "Get Print Queue" (

[2]Mira en http://anu.samba.org/cifs/docs/what-is-smb.html el excelente sumario de Richard sobre SMB.

SMBsplretq), tiene su propio conjunto de parámetros y datos. Al igual que en el caso de los campos en las cabeceras SMB, no todos los campos de comando necesitan estar rellenados, dependiendo de cada comando específico. Por ejemplo, el comando "Get Server Attributes" (SMBdskattr) establece los campos WCT y BCC a cero. Los campos del segmento de comando se muestran en la Tabla 3.2.

No te preocupes si no comprendes cada uno de estos campos; no es necesario para usar Samba a un nivel de administración. Sin embargo, conocer esto te va a resultar útil cuando analices o depures mensajes del sistema. Te mostraremos alguno de los mensajes de sistema SMB más comunes que clientes y servidores envían usando una versión modificada de tcpdump más adelante en esta sección. (Si quieres un sniffer SMB con una interfaz gráfica, prueba "ethereal", que usa las librerías GTK; mira en la página web de Samba para más información sobre esta herramienta).

Si deseas más información sobre cada uno de los comandos del protocolo SMB, mira la documentación sobre SMB/CIFS en ftp://ftp.microsoft.com/developr/drg/CIFS/.

3.3.1.3. Variaciones sobre SMB.

El protocolo SMB ha sido extendido con nuevos comandos muchas veces desde su concepción. Cada nueva versión es compatible con las anteriores. Esto hace posible tener en una misma LAN varios clientes y servidores corriendo diferentes versiones del protocolo SMB al mismo tiempo.

La Tabla 3.3. muestra las mejores versiones del protocolo SMB. Dentro de cada "versión" de SMB hay muchas subversiones que incluyen comandos soportando versiones determinadas de los mejores sistemas operativos. El ID de cadena es usado por clientes y servidores para determinar qué nivel de protocolo usarán para hablar unos con otros.

Samba implementa la especificación NT LM 0.12 para NT LAN Manager 1.0. Este es compatible con todas las variantes de versiones anteriores. La especificación CIFS es, en realidad, LAN Manager 0.12 con unas cuantas adiciones específicas.

3.3.2. Clientes y Servidores SMB.

Como mencionamos antes, SMB es un protocolo cliente/servidor. Esto significa que un cliente envía una petición a un servidor, el cual actúa en función de la petición y envía una respuesta. Sin embargo, los roles de cliente/servidor pueden invertirse frecuentemente, algunas veces dentro del contexto de una misma sesión SMB. Por ejemplo, considera dos computadoras Windows 95/98 como en la Figura 3.28. La computadora llamada WIZZIN comparte una impresora para la red, y la llamada ESCRIME comparte un directorio de disco. WIZZIN está en el papel de cliente cuando accede a la unidad de red de ESCRIME, y se pone en el papel de servidor cuando imprime un trabajo para ESCRIME.

Esto nos lleva a un punto importante en la terminología Samba:

- Un servidor es una máquina con un recurso compartido.
- Un cliente es una máquina que quiere usar un recurso compartido.

Cuadro 3.1: Campos de la Cabecera SMB.

Campo	**Tamaño (bytes)**	**Descripción**
0xFF 'SMB'	1	Identificador de Protocolo
COM	1	Código Comando, desde 0x00 hasta 0xFF
RCLS	1	Clase de Error
REH	1	Reservado
ERR	2	Código de Error
REB	1	Reservado
RES	14	Reservado
TID	2	TID; un ID único para un recurso en uso por un cliente
PID	2	ID de Proceso
UID	2	ID de Usuario
MID	2	Multiplex identifier; usado para rutar peticiones dentro de procesos

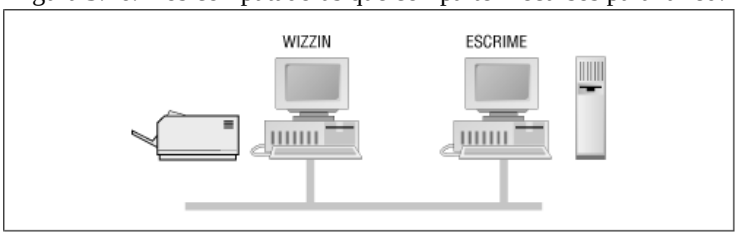
Cuadro 3.2: Contenido de un Comando SMB.

Campo	**Tamaño (bytes)**	**Descripción**
WCT	1	Contador de Palabras
VWV	Variable	Parámetro de palabras (tamaño dado por WCT)
BCC	2	Contador de byte de Parámetro
DATA	Variable	Dato (tamaño dado por BCC)

Cuadro 3.3: Dialectos del Protocolo SMB.

Nombre Protocolo	**ID de Cadena**	**Usado por**
Core	PC NETWORK PROGRAM 1.0	
Core Plus	MICROSOFT NETWORKS 1.03	
LAN Manager 1.0	LANMAN1.0	
LAN Manager 2.0	LM1.2X002	
LAN Manager 2.1	LANMAN2.1	
NT LAN Manager 1.0	NT LM 0.12	Windows NT 4.0
Samba's NT LM 0.12	Samba	Samba
Common Internet File System	CIFS 1.0	Windows 2000

Figura 3.16: Dos computadoras que comparten recursos para la red.



- Un servidor puede ser un cliente en cualquier momento.

Advierte que no hay implicaciones en cuanto a la cantidad de recursos que ofrece un servidor, o si tiene mayor capacidad de disco o velocidad de procesador. Un servidor podría ser un viejo 486 con una impresora conectada a él, o podría ser una estación UltraSparc con un servicio de disco de 20 gigabytes.

Los productos de Microsoft Windows contienen la tecnología cliente/servidor SMB incluídas en el sistema operativo. Wndows NT 4.0 usa un nuevo protocolo SMB distinto al de Windows para Grupos de Trabajo, y ofrece el valor añadido de la seguridad en la red, lo cual discutiremos en el Capítulo 6. En adición, hay un largo número de productos comerciales de servidores SMB disponibles para compañías tales como Sun, Compaq, SCO, Hewlett-Packard, Syntax, e IBM. Desafortunadamente, en la parte del cliente hay muy pocas ofertas,limitándose principalmente a productos de redes de equipos Digital , y por supuesto, Samba.

3.3.3. Una Simple Conexión SMB.

Antes de dar por terminado este capítulo, echemos un vistazo a una simple conexión SMB. Lo que vamos a comentar ahora son algunos interesantes datos técnicos, que realmente no son necesarios para administrar Samba, así que si quieres, esto te lo puedes saltar. Presentamos esta información lo más desmenuzada posible para que te familiarices con la forma en que el protocolo SMB negocia conexiones con otras computadoras en la red.

Hay cuatro pasos que un cliente y un servidor deben completar en orden a establecer una conexión con un recurso:

1. Establecer una conexión virtual.
2. Negociar la variante de protocolo a usar para hablar entre ellos.
3. Establecer los parámetros de la sesión.
4. Realizar una conexión de árbol a un recurso (TID).

Examinaremos cada uno de estos pasos a través de los ojos de una útil herramienta que ya hemos mencionado anteriormente: el modificado tcpdump que está disponible desde el sitio web de Samba.

Puedes descargar este programa desde samba.org en el directorio samba/ftp/tcpdump-smb; la última versión a la hora de escribir este manual era la 3.4-5. Usa este programa tal como usarías la aplicación estándar tcpdump, pero añade el switch -s 1500 para asegurarte de que obtienes todo el paquete y no sólo los primeros bytes del mismo.

3.3.3.1. Estableciendo una Conexión Virtual.

Cuando un usuario realiza una primera petición para acceder a un disco de red o para enviar un trabajo a una impresora remota, NetBIOS se encarga de realizar una conexión (sesión). El resultado es un canal virtual

bidireccional entre el cliente y el servidor. En realidad, sólo hay dos mensajes que el cliente y el servidor necesitan para establecer esa conexión. Esto se muestra en el siguiente ejemplo de petición y respuesta en una sesión, tal como saldría de la catpura a través de tcpdump :

```
>>> NBT Packet
NBT Session Request
Flags=0x81000044
Destination=ESCRIME NameType=0x20 (Server)
Source=WIZZIN       NameType=0x00 (Workstation)

>>> NBT Packet
NBT Session
Granted Flags=0x82000000
```

3.3.4. Negociando la Variante de Protocolo.

En este punto, existe ya un canal abierto entre el cliente y el servidor. A continuación, el cliente envía un mensaje al servidor para negociar un protocolo SMB. Como ya mencionamos antes, el cliente establece su campo identificador de árbol o "tree identifier" (TID) a cero, ya que no sabe todavía qué TID usar. Un identificador de árbol o TID es un número que representa una conexión a un recurso compartido en un servidor.

El comando en el mensaje es SMBnegprot, una petición para negociar una variante de protocolo que será usada durante toda la sesión. Advierte que el cliente envía al servidor una lista con todas las variantes que este puede hablar, y no vicecersa.

El servidor responde a la petición SMBnegprot con la lista de variantes que el cliente ofrece ordenada, comenzando por el índice 0, o el valor 0xFF si ninguno de los protocolos es aceptable. Continuando con nuestro ejemplo, el servidor responde con el valor 5, lo cual indica que el dialecto NT LM 0.12 es el que será usado para el resto de la sesión:

```
>>> NBT Packet
  NBT Session
  Packet Flags=0x0
  Length=154

  SMB PACKET: SMBnegprot (REQUEST)
  SMB Command = 0x72
  Error class = 0x0
  Error code = 0
  Flags1 = 0x0
  Flags2 = 0x0
  Tree ID = 0
  Proc ID = 5371
  UID = 0
  MID = 385
  Word Count = 0
  Dialect=PC
  NETWORK PROGRAM 1.0
  Dialect=MICROSOFT NETWORKS 3.0
```

```
  Dialect=DOS LM1.2X002
  Dialect=DOS LANMAN2.1
  Dialect=Windows for Workgroups 3.1a
  Dialect=NT LM 0.12

>>> NBT Packet
  NBT Session
  Packet Flags=0x0
  Length=69

  SMB PACKET: SMBnegprot (REPLY)
  SMB Command = 0x72
  Error class = 0x0
  Error code = 0
  Flags1 = 0x0
  Flags2 = 0x1
  Tree ID = 0
  Proc ID = 5371
  UID = 0
  MID = 385
  Word Count = 02 [000] 05 00
```

3.3.5. Estableciendo los Parámetros de Sesión y de Logeado.

El siguiente paso es transmitir los parámetros de sesión y de logeado para la sesión. Esto incluye el nombre de cuenta y contraseña (si la hay), el nombre del grupo de trabajo, el tamaño máximo de datos que se pueden transmitir, y el núemro de peticiones pendientes que pueden admitirse en situación de espera al mismo tiempo.

En el siguiente ejemplo, el comando "Session Setup" permite añadirle un comando SMB adicional. La letra X al final del nombre del comando indica esto, y el código hexadecimal del segundo comando es dado en el campo Com2. En este caso el comando es 0x75, el cual es el comando "Tree Connect" y el comando X. El mensaje SMBtconX busca el nombre del recurso en el búfer smb_buf. (Este es el último campo listado en la siguiente petición). En este ejemplo, smb_buf contiene la cadena \\ESCRIME\PUBLIC, la cual es la ruta completa a un directorio compartido en el nodo ESCRIME. Usar los comandos del tipo "y X" acelera cada transacción, ya que el servidor no tiene que esperar a que el cliente haga una segunda petición.

Advierte que el TID sigue valiendo cero todavía. El servidor proporcionará un TID al cliente una vez que la sesión haya sido establecida y una conexión haya sido realizada para el recurso solicitado. En adición, advierte que la contraseña es enviada en la apertura. Podremos cambiar esto más adelante usando contraseñas encriptadas.

```
>>> NBT Packet
  NBT Session
  Packet Flags=0x0
  Length=139
  SMB PACKET: SMBsesssetupX (REQUEST)
  SMB Command = 0x73
  Error class = 0x0
  Error code = 0
  Flags1 = 0x10
  Flags2 = 0x0
  Tree ID = 0
  Proc ID = 5371
  UID = 1
  MID = 385
```

```
  Word Count = 13
  Com2=0x75
  Res1=0x0
  Off2=106
  MaxBuffer=2920
  MaxMpx=2
  VcNumber=0
  SessionKey=0x1FF2
  CaseInsensitivePasswordLength=1
  CaseSensitivePasswordLength=1
  Res=0x0
  Capabilities=0x1
  Pass1&Pass2&Account&Domain&OS&LanMan= KRISTIN PARKSTR Windows 4.0 Windows 4.0
  PassLen=2
  Passwd&Path&Device=
  smb_bcc=22
  smb_buf[]=\\ESCRIME\PUBLIC
```

3.3.6. Relizando Conexiones a un Recurso.

Como paso final, el servidor retorna un TID al cliente, indicando que el usuario tiene acceso autorizado y que el recurso está listo para ser usado. También establece el campo ServiceType al valor "A" para indicar que esto es un servicio de ficheros.Los tipos de servicios disponibles son:

- “A” para una unidad de disco o ficheros.
- “LPT1” para un servicio de impresoras.
- “COMM” para una conexión directa a una impresora o modem.
- “IPC” para un nombre de tubería.

La salida es:

```
>>> NBT Packet
  NBT Session
  Packet Flags=0x0
  Length=78
  SMB PACKET: SMBsesssetupX (REPLY)
  SMB Command = 0x73
  Error class = 0x0
  Error code = 0
  Flags1 = 0x80
  Flags2 = 0x1
  Tree ID = 121
  Proc ID = 5371
  UID = 1
  MID = 385
  Word Count = 3
  Com2=0x75
  Off2=68
  Action=0x1
  [000] Unix Samba 1.9.1
  [010] PARKSTR
  SMB PACKET: SMBtconX (REPLY) (CHAINED)
  smbvwv[]=
  Com2=0xFF
  Off2=78
  smbbuf[]=
  ServiceType=A:
```

Ahora que un TID ha sido asignado, el cliente puede proporcionar cualquier tipo de comando que sea posible usar sobre una unidad de disco. Puede abrir ficheros, leer y escribir en ellos, eliminarlos, crear nuevos, realizar búsquedas por nombre de fichero, etc.

Capítulo 4

Compartición de Unidades de Disco.

En los tres capítulos anteriores, te mostramos cómo instalar Samba sobre un servidor Unix y cómo configurar los clientes Windows para usar una simple compartición de disco. Este capítulo te mostrará cómo Samba puede asumir otros roles más productivos sobre tu red.

Los demonios de Samba, smbd y nmbd, son controlados a través de un simple fichero ASCII, smb.conf, que puede contener unas 200 opciones de configuración. Estas opciones determinan cómo reacciona Samba ante la red, incluyéndolo todo, desde simples permisos hasta conexiones encriptadas y dominios NT. Los siguientes cinco capítulos están diseñados para ayudar a familiarizarte con éste fichero y sus opciones. Algunas de estas opciones las usarás y cambiarás frecuentemente; otras puede que nunca las uses. Todo dependerá de la funcionalidad que desees Samba proporcione a los clientes.

Este capítulo te introduce en la estructura del fichero de configuración de Samba y muestra cómo usar estas opciones para crear y modificar discos compartidos. Los capítulos siguientes discutirán sobre visualización/navegación (browsing), cómo configurar usuarios, seguridad, dominios e impresoras, y sobre un montón de cosas más que puedes implementar con Samba en tu red.

4.1. Aprendiendo a usar el Fichero de Configuración de Samba.

Aquí tienes un ejemplo de fichero de configuración de Samba. Si ya has trabajado con los ficheros .INI de Windows, la estructura del fichero smb.conf te resultará familiar:

```
[global]
  log level = 1
  max log size = 1000
  socket options = TCP_NODELAY IPTOS_LOWDELAY
  guest ok = no
```

```
[homes]
  browseable = no
  map archive = yes

[printers]
  path = /usr/tmp
  guest ok = yes
  printable = yes
  min print space = 2000

[test]
  browseable = yes
  read only = yes
  guest ok = yes
  path = /export/samba/test
```

Aunque no puedas comprender todavía los contenidos, este es un buen fichero de configuración para usar en caso de problemas. Este fichero de configuración establece un nivel de debug (control de errores/avisos) básico que no excederá de 1MB, optimiza el socket de conexiones TCP/IP entre el servidor Samba y cualesquiera clientes SMB, y permite a Samba crear un disco compartido para cada usuario que tenga una cuenta estándar Unix en el servidor. En adición, cada una de las impresoras registradas en el servidor están disponibles públicamente, y además existe un recurso de acceso libre (pero sólo lectura) que mapea el directorio */export/samba/test*. La última parte de éste fichero es similar a la de compartición de disco que ya usamos para testear Samba en el C*apítulo 2, Instalando Samba sobre un Sistema Unix*.

4.1.1. Estructura del Fichero de Configuración.

Echemos otro vistazo al fichero de configuración, esta vez a un nivel más general:

```
[global]
...

[homes]
...

[printers]
...

[test]
...
```

Los nombres encerrados entre corchetes delimitan secciones únicas del fichero smb.conf file; cada nombre de sección denomina al recurso (o servicio) a la que ser refiere la sección. Por ejemplo, las secciones [test] y [homes] son cada una de ellas comparticiones de disco únicas; contienen opciones que mapean a directorios específicos del servidor Samba. La sección [printers] contiene opciones que mapean a varias impresoras del servidor. Todas las secciones definidas en el fichero smb.conf, con la excepción de la sección [global], estarán disponibles como discos o impresoras compartidas para los clientes del servidor Samba.

El resto de líneas son opciones individuales de configuración para cada recurso. Estas opciones continuarán hasta que se encuentre una nueva sección rodeada entre corchetes, o hasta el final del fichero. Cada opción de configuración sigue un formato muy simple:

```
opción = valor
```

Las opciones en el fichero smb.conf son configuradas asignándoles una valor. Debemos advertirte de que algunos de los nombres escogidos para las opciones son muy poco explicatorios de su función.Por ejemplo, read only se explica por sí solo public es una vieja opción, y es de definición vaga; ahora tiene una menos confusa sinónima en guest ok (puede ser accedida por invitados). Describimos algunos de los nombres más comunes en éste capítulo. En adición, el *Apéndide C, Referencia Rápida de Opciones de Configuración de Samba*, contiene un índice alfabético de todas las opciones de configuración y su significado.

4.1.1.1. Espacios en Blanco, Comillas y Comas.

Algo importante a recordar sobre las opciones de configuración es que todos los espacios en blanco en la parte del valor son significativos. Por ejemplo, considera la siguiente opción:

```
volume = The Big Bad Hard Drive Number 3543
```

Samba desprecia los espacios entre la e final de volume y la primera T en The. Son no significativos. El resto de espacios son significativos y serán reconocidos y preservados por Samba cuando lea el fichero. Los espcios no son significativos en nombres de opción (como en guest ok), pero recomendamos que sigas la convención y mantengas espacios entre las palabras que componen los nombres de opciones.

Si te sientes más encerrando el valor de una opción entre comillas, puedes hacerlo. Samba ignorará estas comillas cuando las encuentre. Pero nunca las uses para un nombre de opción; Samba las tratará como un error.

Finalmente, puedes usar espacios en blanco o comas para separar una serie de valores en una lista. Lo siguiente es igualmente válido:

```
netbios aliases = sales, accounting, payroll
netbios aliases = sales accounting payroll
```

En algunos valores, sin embargo, deberás usar una forma específica de separación (o espacios o comas).

4.1.1.2. Capitalización.

La capitalización no es importante en el fichero de configuración de Samba excepto en lugares donde estas podrían llevar a confusión al sistema operativo que lo aloja. Por ejemplo, asumamos que has incluido la siguiente opción en un recurso que apunta a */export/samba/simple*:

```
PATH = /EXPORT/SAMBA/SIMPLE
```

Samba no tendría problemas con la opción de configuración path si esta apareciera íntegramente en mayúsculas. Sin embargo, cuando intente conectar a dicho directorio, podría fallar porque el sistema operativo Unix es sensible a mayúsculas/minúsculas. Consecuentemente, la ruta indicada podría no ser encontrada y los clientes no podrían conectar al recurso.

4.1.1.3. Continuación de Línea.

Puedes continuar una línea en el fichero de configuración de Samba usando la barra o "backslash", como sigue:

```
comment = The first share that has the primary copies \
          of the new Teamworks software product.
```

A consecuencia de la barra o backslash, estas dos líneas serán tratadas como una sola por Samba. La segunda línea comenzará en el primer carácter distinto de espacio en blanco que Samba encuentre; en este caso, o en off.

4.1.1.4. Comentarios.

Puedes insertar comentarios en el fichero smb.conf antecediéndolos por una almohadilla (#) o por un punto y coma (;). Ambos caracteres son *equivalentes*. Por ejemplo, las primeras tres líneas en el siguiente ejemplo se consideran comentarios:

```
# This is the printers section. We have given a minimum print
; space of 2000 to prevent some errors that we've seen when
; the spooler runs out of space.

[printers]
public = yes
min print space = 2000
```

Samba ignorará todas las líneas comentadas en su fichero de configuración; advierte que el símbolo barra (\) no será tenido en cuenta como continuador de una línea comentada. Simplemente también será ignorado, como el resto de la línea. 4.1.1.5

4.1.1.5. Cambios en Tiempo de Ejecución.

Puedes modificar el fichero smb.conf y cualquiera de sus opciones en cualquier momento, incluso cuando el servidor está corriendo. Por defecto, Samba chequea el fichero de configuración cada 60 para saber si debe realizar cambios. Si los encuentra, los cambios son inmediatamente aplicados. Si no quieres que espere tanto, puedes forzar la recarga enviando una señal SIGHUP a los procesos smbd y nmbd, o simplemente reiniciando los demonios.

Por ejemplo, si el proceso smbd fuera el 893, podrías forzar la relectura del fichero de configuración con el siguiente comando:

```
# kill -SIGHUP 893
```

Advierte que NO TODOS los cambios serán inmediatamente reconocidos por los clientes. Por ejemplo, los cambios sobre un recurso que está actualmente en uso no serán registrados hasta que el cliente desconecte y reconecte de nuevo al recurso. En adición, los parámetros específicos del servidor, tales como el nombre de grupo de trabajo o el nombre NetBIOS del servidor no serán regsitrados inmediatamente. Así se evitan problemas con las conexiones actuales de los clientes mientras existen sesiones abiertas.

4.1.2. Variables.

Samba incluye un completo juego de variables para determinar las características del servidor Samba y de los clientes a los cuales conecta. Cada una de estas variables comienza con un símbolo de porcentaje (%), seguido por un único carácter minúscula o mayúscula, y puede ser usada sólo en la parte del valor de una opción de configuración (p.ej., después del signo igual):

```
[pub] path = /home/ftp/pub/%a
```

La variable %a contiene como valor la arquitectura de la máquina cliente (p.ej. WinNT para Windows NT, Win95 para Windows 95 o 98, o WfWg para Windows para Trabajo en Grupos). A consecuencia de esto, Samba asignará una ruta única para el recurso [pub] para las máquinas clientes que corran Windows NT, una diferente para las máquinas clientes que corran Windows 95, y otra para Windows para Trabajo en Grupo. En otras palabras, las rutas que cada cliente podrá ver serán diferentes, en función de la arquitectura de los clientes, como sigue:

```
/home/ftp/pub/WinNT /home/ftp/pub/Win95 /home/ftp/pub/WfWg
```

El uso de variables de esta forma es útil si tienes diferentes clientes corriendo diversas plataformas y deseas crear configuraciones en función de las mismas. Samba tiene 19 variables, que ves en la Tabla 4.1.

Aquí tienes otro ejemplo del uso de variables: digamos que hay 5 clientes en tu red, pero uno de ellos, fred, requirere un configuración de [homes] algo diferente al resto cuando conecta al servidor Samba. Con Samba, esto tiene fácil solución:

```
[homes]
  ...
  include = /usr/local/samba/lib/smb.conf.%m
  ...
```

La opción include aquí provoca un fichero de configuración separado para cada máquina NetBIOS (%m), que será leído en adición al fichero actual. Si el nombre de host de la máquina cliente es fred, y el fichero smb.conf.fred existe en el directorio samba_dir /lib/ directory (o el que hayas especificado en tu configuración), Samba insertará ese fichero de configuración en el genérico smb.conf. Si alguna opción de configuración existe en ambos ficheros, esos valores serán redefinidos por la configuración del fichero particular de cada usuario. Pero OJO. Si cualesquiera opción tras la opción include vuelve a ser redefinida en el fichero de configuración principal, Samba volverá a redefinir su valor. Es decir, siempre prevalece lo último.

Algo importante: si no existe el fichero particular de usuario, Samba NO generará error. De hecho, no hará nada. Esto te va a permitir el poder crear un fichero de configuración extra sólo para fred, en lugar de tener que crear uno para cada máquina NetBIOS de la red.

Los ficheros de configuración específicos de una máquina pueden ser usados tanto para optimizar la configuración de determinados clientes como para hacer el control de errores de Samba más sencillo. Considera lo último; si tenemos un cliente con un problema, podemos usar esta técnica para asignarle un nivel de depuración de errores mayor y dirigir la salida a un fichero de registro particular para esa máquina. Esto nos permitirá ver qué está haciendo Samba sin que afecte al resto de clientes o sobrecargar

Cuadro 4.1: Variables de Samba.

Variable	**Definición**
Relativas a Clientes	
%a	Arquitectura de Cliente (p.ej., Samba, WfWg, WinNT, Win95, o UNKNOWN)
%I	Dirección IP de Cliente (p.ej., 192.168.220.100)
%m	Nombre NetBIOS de Cliente
%M	Nombre DNS de Cliente
Relativas a Usuarios	
%g	Grupo Primario de %u
%G	Grupo Primario de %U
%H	Directorio "home" de %u
%u	Actual nombre usuario Unix
%U	Nombre de usuario (no siempre usado por Samba)
Relativas a Recursos	
%p	Automontador de ruta para el recurso, si difiere de %P
%P	Actual directorio root del recurso
%S	Actual nombre del recurso
Relativas a Servidor	
%d	Actual PID de servidor
%h	nombre DNS de máquina del servidor Samba
%L	Nombre NetBIOS del servidor Samba
%N	Directorio "home" del servidor, desde el mapa automount
%v	Versión de Samba
Varias	
%R	Nivel de protocolo SMB que se ha negociado
%T	Fecha y hora actual

el disco con grabaciones de registros de sucesos de todo el mundo. Recuerda, ¡En grandes redes puede que no tengas siempre la posibilidad de reiniciar el servidor Samba para operaciones de depuración!

Puedes usar las variables de la Tabla 4.1 para asignar valores a una gran variedad de opciones de Samba. Remarcaremos algunas de ellas a medida que avancemos en los próximos capítulos.

4.2. Secciones Especiales.

Tras nuestro paso por las variables, hay unas cuantas secciones especiales en el fichero de configuración de Samba de las que vamos a hablar ahora. De nuevo, no te preocupes si no comprendes todas y cada una de las opciones de configuración listadas a continuación; iremos sobre cada una de ellas más específicamente a medida que avancemos en los capítulos.

4.2.1. La Sección [globals].

La sección [globals] aparece en todos los ficheros de configuración de Samba, aunque no es obligatoria su definición. Cualquier opción de esta sección se aplicará al resto de recursos, como si los contenidos de la sección se copiasen a todas las demás. Sólo una salvedad: otras secciones pueden contener la misma opción pero con distinto valor; lo último prevalece siempre sobre lo antiguo, así que ese último valor prevalecerá sobre el establecido en la sección [globals].

Para ilustrar esto, miremos de nuevo el ejemplo del capítulo:

```
[global]
  log level = 1
  max log size = 1000
  socket options = TCP_NODELAY IPTOS_LOWDELAY
  guest ok = no

[homes]
  browseable = no
  map archive = yes

[printers]
  path = /usr/tmp
  guest ok = yes
  printable = yes
  min print space = 2000

[test]
  browseable = yes
  read only = yes
  guest ok = yes
  path = /export/samba/test
```

En el ejmplo, si vamos a conectar un cliente al recurso [test], Samba leería primero la sección [globals]. En éste punto, establecería la opción guest ok = no como valor global por defecto para cada recurso que se encuentre en el fichero de configuración.

Esto incluye a los recursos [homes] y [printers]. Cuando lea la sección para el recurso [test], sin embargo, se encontrará con la opción de configuración guest ok = yes, y cambiará el valor que le venía de la secció [globals] con el valor yes en el contesto del recurso [pub].

Cualquier opción que aparezca fuera de una sección se asume perteneciente a la sección global.

4.2.2. La Sección [homes].

Si un cliente intenta conectar a un recurso que no aparece en el fichero smb.conf, Samba buscará un recurso [homes] en el fichero de configuración. Si existe alguno, el recurso no identificado es asumido como nombre de usuario Unix, el cual es interrogado en la B.D. de contraseñas del servidor Samba. Si el usuario aparece registrado, Samba asume que el cliente es un usuario Unix intentando conectar a su cuenta "home" en el servidor Unix.

Por ejemplo, supongamos que una máquina cliente se conecta al servidor Samba hydra por primera vez, e intenta conectar a un recurso denominado [alice]. Resulta que no existe ningún recurso [alice] definido en el fichero smb.conf, pero sí existe [homes], así que Samba busca en el fichero de contraseñas y encuentra a una cuenta de usuario llamada alice presente en el sistema. Samba entonces comprueba la contraseña proporcionada por el cliente contra la contraseña de usuario Unix de alice. Si las contraseñas coinciden, entonces Samba lo reconoce como sigue: el usuario alice está intentando conectar con su directorio "home". Samba creará un recurso llamado [alice] para él.

El proceso de uso de la sección [homes] para crear usuarios (y controlar su acceso a través de sus contraseñas) se discutirá con más detalle en el Capítulo 6, Usuarios, Seguridad y Dominios.

4.2.3. La Sección [printers].

La tercera sección especial se denomina [printers] y es similar a [homes]. Si un cliente intenta conectar a un recurso que no existe en el fichero smb.conf, y su nombre no puede ser encontrado en el fichero de contraseñas, Samba comprobará si el recurso solicitado es una impresora. Samba lo hace leyendo el fichero de capacidades de impresora (normalmente /etc/printcap) para ver si el nombre del recurso aparece ahí [1]. Si existe, Samba crea el recurso.

Como [homes], esto significa que no tienes por qué mantener un recurso para cada una de tus impresoras en el fichero smb.conf. En su lugar, Samba comprueba el registro de impresoras de Unix, y proporciona acceso a las impresoras registradas a las máquinas clientes. Sin embargo, existe una limitación obvia: si tienes una cuenta llamada fred y una impresora llamada también fred, Samba siempre encontrará la cuenta de usuario primero, aunque el cliente necesite realmente acceder a la impresora.

El proceso de configurar el recurso [printers] se discute en más detalle en el *Capítulo 7, Impresión y Resolución de Nombres.*

[1] Dependiendo de tu sistema, este fichero puede no estar o ser /etc/printcap. Puedes usar el comando testparm que viene con Samba para determinar el valor de la opción de configuración printcap name; este fue el valor por defecto escogido cuando Samba fue compilado.

4.2.4. Opciones de Configuración.

Las opciones de los ficheros de configuración de Samba entran en dos posibles categorías : globales o recursos (shares). Cada categoría dicta dónde una opción puede aparecer en el fichero de configuración.

Global Las opciones globales deben aparecer en la sección [global] y sólo ahí. Estas son opciones que normalmente se aplican al servidor Samba, y no a los recursos que se dan a compartir.

Share Las opciones de recurso compartido o "share" pueden aparecer en las secciones de recursos específicos, o pueden aparecer en la sección [global]. Si aparecen en la sección [global], definirán un valor por defecto para todos los recursos, a menos que un recurso determinado redefina ese valor.

En adición, los valores que una opción de configuración puede tomar pueden ser divididos en cuatro categorías:

Booleano Simplemente "yes" o "no", pero pueden ser representados por cualquiera de los siguientes valores: yes, no, true, false, 0, 1. Los valores no son sensibles a mayúsculas: YES es lo mismo que yes.

Numéricos Un número entero, hexadecimal u octal. La sintaxis estándar 0x nn es usada para valores hexadecimales, y 0 nnn para octales.

Cadena Una cadena de caracteres sensibles a may/min, tales como un nombre de fichero o de usuario.

Lista Enumerada Una lista finita de valores conocidos. Por ejemplo, un booleano es una lista enumerada definida con sólo dos valores.

4.3. Opciones del Ficheros de Configuración.

Samba pone unas 200 opciones de configuración a tu disposición. Así que empezaremos por las sencillas, introduciéndote en algunas de las opciones que puedes usar para modificar el fichero de configuración.

Como ya comentamos antes, los ficheros de configuración no implican situaciones estáticas. Puedes ordenar a Samba que incluya o reemplace opciones de configuración durante el tiempo de ejecución. Las opciones para hacer esto se sumarizan en la Tabla 4.2.

4.3.1. Fichero de Configuración.

La opción global config file especifica un fichero de configuración de reemplazo que será cargado cuando esta opción sea encontrada. Si el fichero obejto existe, el resto del fichero de configuración actual, así como las opciones que se encuentre a partir de la aparición de esta opción, serán ignorados; Samba se configurará con las opciones del nuevo fichero. La opción config file toma ventaja del uso de las variables que ya vimos antes, lo cual es útil para el caso de que desees cargar un fichero de configuración especial en base al nombre de máquina o de usuario del cliente que se va a conectar.

Por ejemplo, lo siguiente instruye a Samba para que use un fichero de configuración especificado por el nombre NetBIOS del cliente que conecte, si tal fichero existe.

Si existe, las opciones especificadas en el fichero de configuración original son ignoradas. El siguiente ejemplo intenta cargar un nuevo fichero de configuración en base al nombre NetBIOS del cliente:

```
[global]
  config file = /usr/local/samba/lib/smb.conf.%m
```

Si el fichero no existe, la opción es ignorada y Samba continuará su configuración en base al actual fichero de configuración.

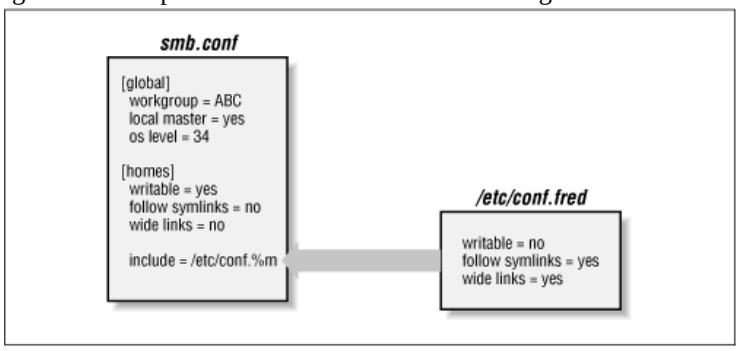
4.3.2. Include.

Esta opción, comentada anteriormente, copia el fichero objetivo en el actual fichero de configuración a partir del punto especificado, como se muestra en la Figura 4.1. Esta opción también toma ventaja gracias al uso de las variables, las cuales son de utilidad para el caso de que quieras opciones de configuración basadas en el nombre de la máquina o del usuario del cliente que está conectando. Puedes usar esta opción como sigue:

```
[global]
  include = /usr/local/samba/lib/smb.conf.%m
```

Si el fichero de configuración especificado no existe, la opción es ignorada. Recuerda que cualquier opción especificada previamente es redefinida. En la Figura 4.1., las tres opciones redefinirán sus valores previos.

Figura 4.1: La opción include en un fichero de configuración de Samba.



La opción include no entiende las variables %u (usuario), %p (actual directorio raíz del recurso), o %s (actual recurso) porque no se no estaban definidas (las variables no tenían valor) al tiempo de la lectura del fichero.

4.3.3. Copy.

La opción de configuración copy te permite clonar las opciones de configuración del nombre de recurso que especifiques en el recurso actual. El recurso fuente debe aparecer en el fichero de configuración antes que el destino. Por ejemplo:

```
[template]
  writable = yes
```

```
  browsable = yes
  valid users = andy, dave, peter

[data]
  path = /usr/local/samba
  copy = template
```

Advierte que cualesquiera opciones en el recurso invocadas por la directiva copy redefinirán los valores en el recurso clonado; no importa que estos aparezcan antes o después de la directiva *copy*.

4.4. Configuración del Servidor.

Ahora es tiempo de configurar tu servidor Samba. Te introduciremos en tres opciones de configuración básicas que pueden aparecer en la sección [global] de tu *smb.conf*:

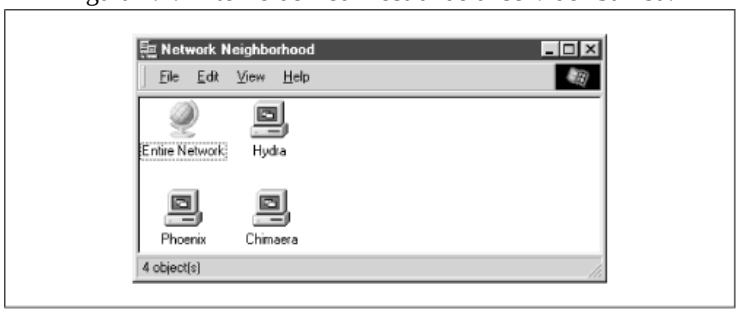
```
[global]
  # Parámetros de configuuración del servidor.
  netbios name = HYDRA
  server string = Samba %v on (%L)
  workgroup = SIMPLE
```

Este fichero de configuración es muy simple; configura el servidor Samba sobre una red NBT bajo el nombre NetBIOS de *hydra*. En adición, la máquina pertenece al grupo de trabajo SIMPLE y displaya un literal descriptivo a los clientes que incluye el número de versión de Samba, así como el nombre NetBIOS del servidor Samba.

Si tenías que introducir la opción *encrypt passwords=yes* en tus anteriores ficheros de configuración, hazlo en este también.

Vamos a seguir con este fichero de configuración. Crea un fichero denominado smb.conf bajo el directorio */usr/local/samba/lib* con el texto que tienes arriba. Luego reinicia el demonio Samba y usa un cliente Windows para verificar los resultados. Asegúrate de que tus clientes Windows están también en el grupo de trabajo SIMPLE. Tras hacer click sobre el icono "Entorno de Red" de un cliente Windows, deberías ver una ventana similar a la de la Figura 4.2. (En esta figura, phoenix y chimaera son tus clientes Windows).

Figura 4.2: Entorno de Red mostrando al servidor Samba.

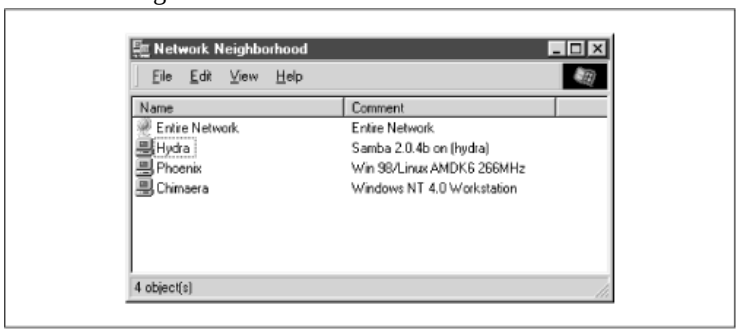


Puedes verificar la opción server string seleccionando "Ver/Detalles" en tu ventana de Entorno de Red (selecciona el elemento de menú "Detalles" bajo el menú "Ver"), y lo verás como en la Figura 4.3.

Cuadro 4.3: Opciones del Fichero de Configuración.

Opción	**Parámetros**	**Función**	**Defecto**	**Ámbito**
config file	string (nombre-completamente-cualificado)	Establece la localización de un fichero de configuración para ser usado en lugar del actual.	Ninguno	Global
include	string (nombre-completamente-cualificado)	Especifica un segmento adicional de opciones de configuración para ser incluidas a partir de ese punto en el fichero de configuración.	Ninguno	Global
copy	string (nombre del recurso)	Te permite clonar las opciones de configuración de un recurso para el recurso actual.	Ninguno	Recurso

Figura 4.3: Vista Detalle del Entorno de Red.



Si picaras sobre el icono de Hydra, debería aparecer una ventana mostrando los servicios que ésta proporciona. En nuestro caso, la ventana debería estar completamente vacía, ya que todavía no se han definido recursos compartidos en el servidor.

4.4.1. Opciones de Configuración del Servidor.

La Tabla 4.3. resume las opciones de configuración introducidas anteriormente. Advierte las tres opciones tienen ámbito global; en otras palabras, deben aparecer en la sección [global] del fichero de configuración.

Cuadro 4.4: Opciones de Configuración del Servidor.

Opción	Parámetros	Función	Defecto	Ambito
netbios name	string	Establece el nombre NetBIOS primario para el servidor Samba.	Nombre de máquina del servidor DNS	Global
server string	string	Establece un literal descriptico para el servidor Samba.	Samba %v	Global
workgroup	string	Establece el nombre NetBIOS de grupo de las máquinas al que pertenece el servidor.	Definido en tiempo de compilación	Global

4.4.1.1. Nombre NetBIOS.

La opción netbios name te permite establecer el nombre NetBIOS del servidor. Por ejemplo:

```
netbios name = YORKVM1
```

El valor por defecto para esta opción de configuración es el nombre de máquina del servidor; esto es, la primera parte del nombre completamente cualificado de la máquina. por ejemplo, una máquina con el nombre DNS ruby.ora.com tomaría como nombre NetBIOS RUBY por defecto. Aunque puedes usar el nombre por defecto, es más común el usar otro nombre NetBIOS distinto al nombre DNS. Recuerda que el nombre que uses debe seguir las reglas de los nombres de máquinas NetBIOS, que ya vimos en el *Capítulo 1, Aprendiendo Samba.*

No es recomendable cambiar el nombre NetBIOS del servidor a menos que tengas una buena razón. Una de esas razones podría ser que el nombre de la máquina no sea único porque la red se encuentra dividida entre dos o más dominios DNS. por ejemplo, YORKVM1 es un buen candidato a nombre para vm1.york.example.com para diferenciarlos de vm1.falkirk.example.com, que es el mismo nombre de máquina pero reside en un dominio DNS diferente.

Otro uso de esta opción es recolocar servicios SMB de una máquina que ha sido retirada de la red por obsoleta. Por ejemplo, si SALES es el servidor SMB para el departamento, y de repente se estropea, puedes resetear inmediatamente netbios name = SALES sobre una máquina de seguridad con Samba que tenías preparado para el caso. los usuarios no tendrán que cambiar sus mapeados de unidades a una máquina diferente; las nuevas conexiones a SALES simplemente irán a la nueva máquina.

4.4.1.2. server string.

El parámetro server string define un literal descriptivo que aparecerá junto al nombre del servidor tanto desde el Entorno de Red (cuando muestra la Vista/Detalles) y la entrada de comentario del gestor de impresión de Microsoft Windows. Puedes usar las variables estandar para proporcionar información en la descripción. por ejemplo:

```
[global]
  server string = Samba %v on (%h)
```

El valor por defecto para ésta opción simplemente presenta la versión actual de Samba y es equivalente a:

```
server string = Samba %v
```

4.4.1.3. workgroup.

El parámetro workgroup establece el actual grupo de trabajo donde el servidor Samba aparecerá. Lo clientes que quieran acceder a sus recursos deberán pertenecer al mismo grupo de trabajo NetBIOS. Recuerda que los nombres de los grupos de trabajo también son nombres NetBIOS, y deben seguir las mismas convenciones para nombres NetBIOS definidas en el Capítulo 1. Por ejemplo:

```
[global]
  workgroup = SIMPLE
```

El valor por defecto para este parámetro es establecido en tiempo de compilación. Si la entrada no se ha cambiado en el makefile, será WORKGROUP. Debido a que este nombre es el mismo que se usa en el caso de redes NetBIOS no configuradas, te recomendamos que siempre establezcas tu nombre de grupo de trabajo en el fichero de configuración de Samba[2].

4.5. Configuración de la Compartición de Disco.

Ya mencionamos en la sección anterior que no teníamos discos compartidos en el servidor hydra. Continuaremos ahora con el fichero de configuración y crearemos una disco compartido vacío llamado [data]. Estas son las adiciones que debemos incorporar:

```
[global]
  netbios name = HYDRA
  server string = Samba %v on (%L)
  workgroup = SIMPLE

[data]
  path = /export/samba/data
  comment = Data Drive
  volume = Sample-Data-Drive
  writeable = yes
  guest ok = yes
```

[2] También deberíamos mencionar que es mala idea tener un grupo de trabajo que tenga el mismo nombre que el que hemos asignado al servidor.

El recurso [data] es el típico en una compartición de disco con Samba. El recurso mapea a un directorio del servidor Samba: */export/samba/data*. También proporcionamos una línea de comentario que describe al recurso como *Data Drive*, así como un nombre de unidad para el recurso en sí.

El recurso es configurado como grabable para que los usuarios puedan escribir datos en él; por defecto, Samba crea recursos de sólo lectura. Como resultado, esta opción necesita ser explícitamente incluída en cada recurso de disco que queramos hacer escribible.

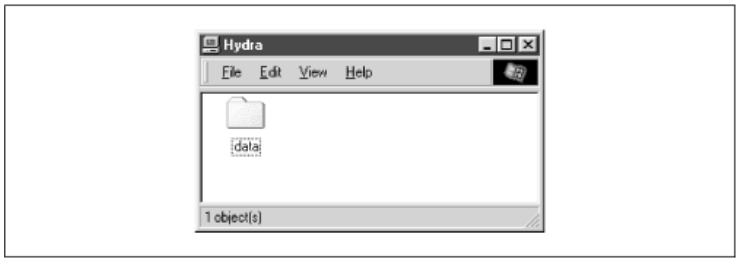
Habrás notado que hemos establecido el parámetro *guest ok* a *yes*. Mientras que no es muy seguro que digamos hacer esto, hay una serie de cosas relativas a las contraseñas que deberemos comprender antes de establecer configuraciones para usuarios individuales y autentificación. Por el momento, vamos a dejar que cualquiera pueda conectar al recurso con esta opción.

Sigamos con las adiciones a nuestro fichero de configuración. En adición, crearemos el directorio /export/samba/data como root en nuestra máquina Samba con los siguientes comandos:

```
# mkdir /export/samba/data
# chmod 777 /export/samba/data
```

Ahora, si conectas de nuevo al servidor hydra (puedes ahcerlo mediante un click sobre su icono en el Entorno de Red de Windows), deberías ver un único recurso listado, llamado data, como se ve en la Figura 4.4. Este recurso debería tener permisos de lectura/escritura. Intenta crear o copiar un fichero sobre el recurso. O, si te sientes aventurero, podrías crear una unidad de red que apunte al recurso.

Figura 4.4: El recurso inicial data en el servidor Samba.



4.5.1. Opciones de Configuración en una Compartición de Disco.

Las opciones básicas de configuración de Samba para recursos de disco se listan en la Tabla 4.4.

4.5.1.1. path.

Esta opción, la cual es sinónima de directory, indica la ruta desde la raíz del recurso de disco o impresión. Puedes seleccionar cualquier ruta en tu servidor Samba, mientras que el propietario del proceso Samba que está corriendo tenga permisos de lectura/escritura sobre el directorio. Si la ruta es para un recurso tipo impresora, debería apuntar a un directorio temporal donde los ficheros puedan ser escritos en el servidor antes de que sean enviados al spooler de la impresora (/tmp y /var/spool son buenas

elecciones). Si la ruta es para un recurso de disco, los contenidos de la carpeta representativa del nombre del recurso en el cliente coincidirán con los contenidos del directorio en el servidor Samba. Por ejemplo, si tenemos el siguiente recurso de disco en nuestro fichero de configuración:

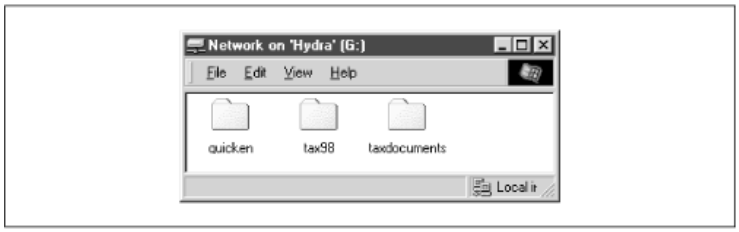
```
[network]
  path = /export/samba/network
  writable = yes
  guest ok = yes
```

Y los contenidos del directorio /usr/local/network en la parte Unix son:

```
$ ls -al /export/samba/network
drwxrwxrwx 9 root nobody 1024 Feb 16 17:17 .
drwxr-xr-x 9 nobody nobody 1024 Feb 16 17:17 ..
drwxr-xr-x 9 nobody nobody 1024 Feb 16 17:17 quicken
drwxr-xr-x 9 nobody nobody 1024 Feb 16 17:17 tax98
drwxr-xr-x 9 nobody nobody 1024 Feb 16 17:17 taxdocuments
```

Entonces deberíamos ver el equivalente (Figura 4.5.) en la parte del cliente:

Figura 4.5: Vista de cliente Windows de un sistema de archivos de red especificado por ruta.



4.5.1.2. guest ok.

Esta opción (la cual tiene una sinónima pero antigua public) permite o prohíbe accesos anónimos a un recurso. El valor por defecto es no. Si se establece a yes, significa que no se necesita nombre de usuario ni contraseña para conectar al recurso. Cuando un usuario conecta, los derechos de acceso son los mismos. La cuenta por defecto a la que Samba ofrece el recurso es nobody. Sin embargo, esto puede ser cambiado con la opción de configuración guest account. Por ejemplo, las siguientes líneas permiten accesos a usuarios anónimos al recurso [accounting] con los permisos de la cuenta ftp:

```
[global]
  guest account = ftp [accounting]
  path = /usr/local/account
  guest ok = yes
```

Advierte que los usuarios aún pueden conectar al recurso usando una combinación nombre de usuario/contraseña válidos. Ellos mantendrán los derechos de acceso garantizados por su propia cuenta y no por la cuenta de anónimo. Si un usuario intenta logearse y falla, sin embargo, obtendrá los permisos establecidos para un usuario anónimo. Puedes obligar a que todo usuario que conecte al recurso lo haga como anónimo (y tendrá los permisos del usuario anónimo) estableciendo la opción *guest only = yes*.

4.5.1.3. comment.

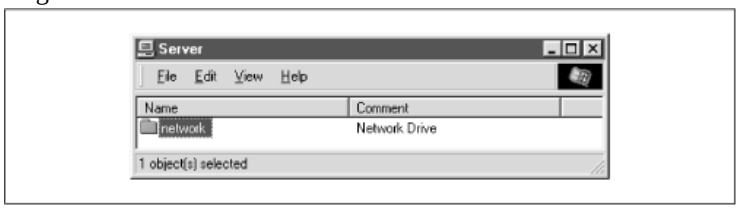
La opción comment te permite introducir un comentario que será enviado al cliente cuando intente navegar por el recurso. El usuario puede ver el comentario seleccionando la opción "Vista/Detalle" en la ventana de Entorno de Red, o tecleando el comando NET VIEW desde una ventana MS-DOS. Por ejemplo, así es como insertarías un comentario para el recurso [network]:

```
[network]
  comment = Network Drive
  path = /export/samba/network
```

Esto generaría una carpeta como la de la Figura 4.6 en la parte del cliente. Advierte que con la actual configuración de Windows, este comentario no será mostrado hasta que un recurso sea mapeado a una unidad de red Windows.

Asegúrate de no confundir la opción comment, la cual documenta los recursos de un servidor Samba, con la opción server string, la cual documenta al servidor mismo.

Figura 4.6: Vista de Cliente Windows de un comentario de recurso.



4.5.1.4. volume.

Esta opción te permite especificar el nombre de volumen para el recurso como lo reportará SMB. Este normalmente suele ser el nombre de recurso dado en el fichero smb.conf. Sin embargo, si quieres llamarle algo más o cambiarlo, puedes hacerlo con ésta opción.

Por ejemplo, un programa de instalación podría chequear el nombre de unidad de un CD-ROM para asegurarse de que el CD-ROM correcto está en la unidad lectora antes de iniciar la instalación. Si copias el contenido del CD-ROM sobre un recurso de red, y quieres instalar desde ahí, puedes usar esta opción para corregir el problema:

```
[network]
  comment = Network Drive
  volume = ASVP-102-RTYUIKA
  path = /home/samba/network
```

4.5.1.5. read only y writeable.

Las opciones *read only* y *writeable* (o *write ok*) son realmente dos formas de decir lo mismo, pero aproximándose desde polos opuestos. Por ejemplo, puedes establecer cualquiera de estas opciones en la sección [global] o en la de un recurso determinado:

```
read only = yes
writeable = no
```

Si las defines como las de arriba, los datos podrán ser leídos desde el recurso, pero no se podrá escribir en él. Podrías pensar que sólo necesitarás esta opción cuando quieras crear un recurso de sólo lectura. Sin embargo, advierte que la opción sólo-lectura es la opción por defecto para los recursos; si quieres habilitar la escritura sobre un recurso, debes indicarlo explícitamente especificando una de las siguientes opciones en ael fichero de configuración para cada recurso:

```
read only = no
writeable = yes
```

Advierte que si especificas más de una ocurrencia de la misma opción, Samba usará el último valor que encuentre para el recurso.

4.6. Opciones de Red con Samba.

Si estás ejecutando Samba sobre una máquina que pertenece a varias subedes, o si quieres implementar una política de seguridad sobre tu propia subred, deberías echar un vistazo a las opciones de configuración de red:

Para los propósitos de este ejemplo, adumiremos que nuestro servidor Samba está conectado a una red con más de una subred. Concretamente, la máquina puede acceder a las subredes 192.168.220.* y 134.213.233.*. Aquí tienes unas adiciones para el fichero de configuración:

```
[global]
  netbios name = HYDRA
  server string = Samba %v on (%L)
  workgroup = SIMPLE

  # Networking configuration options
  hosts allow = 192.168.220. 134.213.233. localhost
  hosts deny = 192.168.220.102
  interfaces = 192.168.220.100/255.255.255.0 \
               134.213.233.110/255.255.255.0

  bind interfaces only = yes

[data]
  path = /home/samba/data
  guest ok = yes
  comment = Data Drive
  volume = Sample-Data-Drive
  writeable = yes
```

Hablaremos primero de las opciones *hosts allow* y *hosts deny*. Si te suenan familiares, estarás pensando probablemente en los ficheros *hosts.allow* y *hosts.deny* que se encuentran en los directorios */et*c de muchos sistemas Unix. El propósito de estas opciones es idéntico al de dichos ficheros; proporcionan una medida de seguridad permitiendo o denegando las conexiones de otras máquinas en base a sus direcciones IP. ¿Y por qué no usamos entonces los archivos *hosts.allow* y *hosts.deny*? Porque pueden existir otros servicios en el servidor que sí quieras ofrecer a esas IPs, pero que no tengan acceso a los recursos que ofrece Samba.

Con la opción *hosts allow*, hemos especificado un rango de IPs: 192.168.220. (fíjate en el punto al final de 220; sólo hemos obviado el cuarto número). Esto equivale a decir: "Todas las máquinas en la red 192.168.220". Sin embargo, también hemos explícitamente denegado el acceso a una IP determinada, 192.168.220.102. Esta no podrá acceder. El resto sí.

Te preguntarás ¿Por qué se denegará el acceso a 192.168.220.102 si está en la subred que autoriza la opción *hosts allow*? Aquí teienes cómo Samba interpreta las reglas especificadas por *hosts allow* y *hosts deny* :

1. Si no hay opciones *allow* o *deny* definidas en *smb.conf*, Samba permitirá conexiones desde cualquier máquina que el propio sistema admita.

2. Si existen opciones *hosts allow* o *hosts deny* definidas en la sección [global] del *smb.conf*, la aplicará a todos los recursos, excepto si los recursos tienen sus propias definiciones establecidas.

3. Si sólo existe definida *hosts allow* para un recurso, sólo las máquinas listadas tendrán acceso al recurso. Las demás serán denegadas.

4. Si sólo hay una opción *hosts deny* definida para un recurso, cualquier máquina que no esté en la lista tendrá acceso al recurso.

5. Si ambas opciones *hosts allow* y *hosts deny* están definidas, una máquina deberá aparecer en la lista de aceptadas y no aparecer en la lista de denegadas. De lo contrario, se le negará el acceso.

   ADVERTENCIA: cuidado con dar acceso a una máquina a un recurso, y luego denegar acceso a toda su subred.

Veamos otro ejemplo. Considera las siguientes opciones:

```
hosts allow = 111.222. hosts deny = 111.222.333.
```

En este caso, sólo las máquinas que pertenezcan a la subred 111.222.*.* tendrán acceso a los recursos de Samba. sin embargo, si un cliente pertenece a la subred 111.222.333.*, le será denegado el acceso, aunque pertenezca al rango de las aceptadas por hosts allow. El cliente debe aparecer en la lista de hosts allow y no debe aparecer en la lista de hosts deny para que pueda tener acceso a un recurso de Samba. Si una máquina intenta acceder a un recurso para el cual no se le permite el acceso, recibirá un mensaje de error.

Las otras dos opciones que podemos especificar son interfaces y bind interface only. Veamos la opción interfaces primero. Samba, por defecto, envía datos sólo desde el interfaz de red primario, el cual en nuestro ejemplo es la subred 192.168.220.100. Si queremos enviar datos a través de más de una interfaz, necesitamos especificar la lista completa con la opción interfaces. En el ejemplo anterior, hemos configurado Samba para que sirva a ambas subredes (192.168.220 y 134.213.233), y para la segunda está actuando a través de la interfaz de red 134.213.233.100. Si tienes más de una interfaz de red en tu computadora, deberías siempre establecer ésta opción, ya que no hay garantías de que que el primer interfaz de red que Samba seleccione sea el correcto.

Finalmente, la opción bind interfaces only instruye al proceso nmbd para que no acepte mensajes de difusión (broadcast) que no sean los de aquellas subredes especificadas con la opción interfaces. Advierte que es diferente a las opciones hosts allow y

hosts deny, las cuales previenen sobre las máquinas que pueden conectar a servicios, pero no controlan la recepción de mensajes de difusión. Usar la opción bind interfaces only es la forma de cortar datagramas de subredes extrañas hacia el servidor Samba. En adición, esta opción instruye al proceso smbd para que enlace sólo con los interfaces listados en la opción interfaces. Esto restringe las redes a las que Samba servirá.

4.6.1. Opciones de Red.

Las opciones de red de las que hemos hablado se resumen en la Tabla 4.5.

4.6.1.1. hosts allow.

La opción hosts allow (a veces escrita como allow hosts) especifica las máquinas que tendrán permiso para acceder a los recursos del servidor Samba, escritas como lista de IPs separadas por comas o espacios en blanco. También puedes emplazar colocando la dirección de tu subred en ésta opción. Por ejemplo, especificamos lo siguiente en nuestro ejemplo:

```
hosts allow = 192.168.220. localhost
```

Advierte que hemos colocado localhost tras la dirección de subred. Uno de los errores más comunes cuando intentamos usar la opción hosts allow es desactivar accidentalmente al servidor Samba para comunicar consigo mismo. El programa smbpasswd ocasionalmente necesitará conectar al servidor Samba como cliente para cambiar una contraseña de usuario encriptada. En adición, la propagación de la visualización local requiere acceso al host local. Si esta opción es activada y la dirección de la máquina localhost no se especifica, los paquetes generados localmente en respuesta a cambios de las contraseñas encriptadas serán descartados por Samba, y la propagación de la lista de visualización no trabajará correctamente. Para evitar esto, permite explícitamente el uso de la dirección de loopback (usa localhost o 127.0.0.1)[3].

Puedes especificar cualquiera de los siguientes formatos para esta opción:

- Nombres de Hosts, tales como ftp.example.com .
- Direcciones IP, como 130.63.9.252.
- Nombres de Dominio, que pueden ser diferenciados de nombres individuales de máquinas porue estos empiezan por un punto. Por ejemplo, , .ora.com representa a todas las máquinas dentro del dominio ora.com.
- Grupos de Red, los cuales comienzan con un símbolo (@), como @printerhosts. Los grupos de red están disponibles en sistemas corriendo páginas amarillas/NIS o NIS+. Si los grupos de red son soportados en tu sistema, debería haber una página de manual sobre netgroups que los describe en más detalle.
- SubRedes, las cuales terminan con un punto. Por ejemplo, 130.63.9. significa todas las máquinas cuyas direcciones IP comenzan por 130.63.9.
- La palabra clava ALL, que permite acceso a cualquier cliente.

[3] Desde Samba 2.0.5, localhost será automáticamente admitido a menos que explícitamente sea denegado.

Cuadro 4.5: Opciones de Configuración Básicas para un Recurso.

Opción	**Parámetros**	**Función**	**Defecto**	**Ambito**
path (directory)	string (nombre completamente cualificado)	Establece el directorio Unix que se proporcionará para un recurso de disco o se usará para el spooler de una impresora compartida	/tmp	Recurso
guest ok (public)	booleano	Si se establece a yes, la autentificación no es necesaria para acceder al recurso	no	Recurso
comment	string	Establece el comentario que aparecerá junto al recurso	Ninguno	Recurso
volume	string	Establece el nombre de la unidad: el nombre DOS para la unidad física	nombre recurso	Recurso
read only	booleano	Si es yes, permite acceso de sólo lectura al recurso	yes	Recurso
writeable (write ok)	booleano	Si es no, permite acceso de sólo lectura al recurso	no	Recurso

Cuadro 4.6: Opciones de Configuración de Red.

Opción	**Parámetros**	**Función**	**Defecto**	**Ambito**
hosts allow (allow hosts)	string (lista de nombres de máquinas)	Especifica las máquinas que pueden conectar a Samba.	ninguno	recurso
hosts deny (deny hosts)	string (lista de nombres de máquinas)	Especifica las máquinas que NO pueden conectar a Samba.	ninguno	recurso
interfaces	string (lista de combinaciones IP/máscara de red)	Establece los interfaces de red a los que Samba atenderá.	dependiente de sistema	Global
bind interfaces only	booleano	Si es yes, Samba sólo enlazará con aquellos interfaces especificados con la opción interfaces.	no	Global
socket address	string (IP)	Establece direcciones IP para la escucha, para usar con múltiples interfaces virtuales en un servidor.	ninguno	Global

- La palabra clave EXCEPT seguida por uno o más nombres, direcciones IP, nombres de dominio, grupos de red o subredes. Por ejemplo, podrías especificar que Samba permita acceso a todas las máquinas excepto a aquellas en la subred 192.168.110 con *hosts allow = ALL EXCEPT 192.168.110*. (no te olvides del punto).

Usar la palabra clave ALL es siempre un mala idea, ya que significa que cualquiera desde cualquier subred puede navegar por tus ficheros si simplemente conocen el nombre de tu servidor.

Advierte que aquí no hay valor por defecto para la opción de configuración hosts allow, aunque el la acción por defecto en el caso de no especificar nada es permitir el acceso desde todos los clientes. Además, si especificas esta opción en la sección [global] del fichero de configuración, esta prevalecerá sobre cualesquiera opciones hosts allow definidas a nivel de recursos.

4.6.1.2. hosts deny.

La opción hosts deny (también deny hosts) especifica máquinas que no tienen permiso para acceder a recursos, escritas en forma de lista de nombres de máquinas o de IPs separadas por una coma o espacio en blanco. Usa el mismo formato que la opción hosts allow. Por ejemplo, para restringir el acceso al servidor para todo el mundo excepto a example.com, escribirías:

```
hosts deny = ALL EXCEPT .example.com
```

Como en el caso de hosts allow, no hay valor por defecto para la opción de configuración hosts deny, aunque si no se especifica nada, se permite el acceso desde todos los clientes. También, si especificas esta opción en la sección [global] del fichero de configuración, esta prevalecerá sobre cualesquiera opciones hosts deny definidas a nivel de los recursos. Si deseas negar el acceso de máquinas a determinados recursos, omite las opciones hosts allow y hosts deny en la sección [global] y defínela a nivel de recurso.

4.6.1.3. interfaces.

La opción interfaces configura las direcciones de red a las cuales quieres que el servidor Samba reconozca y responda. Esta opción es útil si tienes una computadora que resida en más de un subred. Si esta opción no se configura, Samba buscará por el primer interfaz de red del servidor (normalmente la primera tarjeta Ethernet) al arrancar y se configurará para operar sólo con esa subred. Si el servidor lo quieres configurar para atender a más de una subred y no especificas esta opción, Samba sólo trabajará con la primera subred que encuentre. Debes usar esta opción para forzar a Samba a servir a las demás subredes de tu red.

El valor de esta opción es uno o más pares de valores Dirección-IP/Máscara-de-Red, tal como las que siguen:

```
interfaces = 192.168.220.100/255.255.255.0 192.168.210.30/255.255.255.0
```

Opcionalmente puedes especificar un formato CIDR, como sigue:

```
interfaces = 192.168.220.100/24 192.168.210.30/24
```

El número de bit de máscara especifica el primer número de bits que serán incluidos en la máscara de red. Por ejemplo, el número 24 significa que los primeros 24 (de 32) bits serán activados en la máscara de bit, lo cual es lo mismo que decir 255.255.255.0. Así, 16 sería equivalente a 255.255.0.0, y 8 lo sería a 255.0.0.0. ADVERTENCIA: Esta opción puede no funcionar correctamente si estás usando DHCP.

4.6.1.4. bind interfaces only.

La opción bind interfaces only puede usarse para forzar a los procesos smbd y nmbd a servir a peticiones SMB de sólo las direcciones especificadas por la opción interfaces. El proceso nmbd normamente enlaza a todas las direcciones (0.0.0.0.) en los puertos 137 y 138, permitiendo la recepción de broadcasts desde cualquier lugar. Sin embargo, puedes arreglar esto con lo siguiente:

```
bind interfaces only = yes
```

Esto causará que ambos procesos de Samba ignoren cualesquiera paquetes cuya dirección de origen no coincida con la dirección(es) de broadcast especificadas por la opción interfaces, incluyendo a los paquetes de broadcast. Con smbd, esta opción causará que Samba no sirva peticiones de ficheros a subredes que no estén listadas en la opción interfaces. Deberías evitar usar esta opción si quieres permitir conexiones temporales de red, tales como las que se crean a través de SLIP o PPP. Es muy raro necesitar esta opción, y sólo debería ser usada por expertos.

Si estableces bind interfaces only a yes , deberías añadir la dirección de localhost (127.0.01) a la lista de "interfaces". De lo contrario, *smbpasswd* será incapaz de conectar al servidor usando su modo por defecto para cambiar una contraseña.

4.6.1.5. socket address.

La opción socket address indica por cuáles de las direcciones especificadas en el parámetro interfaces debería escuchar Samba a la espera de atender posibles conexiones. Por defecto, Samba acepta conexiones en todas las direcciones especificadas. Cuando se usa en un fichero smb.conf, esta opción forzará a Samba a escuchar sólo por una dirección IP. Por ejemplo:

```
interfaces = 192.168.220.100/24 192.168.210.30/24
socket address = 192.168.210.30
```

Esta opción es más que nada una herramienta para programadores y recomendamos no usarla.

4.7. Servidores Virtuales.

Los servidores virtuales son una técnica para crear múltiples servidores NetBIOS en la red, cuando en realidad sólo existe uno. La técnica es simple de implementar: una máquina simplemente registra más de un nombre NetBIOS en asociación con su dirección IP. Hay beneficios tangibles en hacer esto.

El departamento de cuentas, por ejemplo, podría tener un servidor llamado accounting, y los clientes de este verían las unidades de disco y de impresión de accounting. El departamento de marketing podría tener su propio servidor, marketing, con sus propias unidades, etcétera. Sin embargo, todos los servicios serían proporcionados por un

estación Unix (y un relajado administrador), en lugar de tener un pequeño servidor y un administrador por cada departamento.

Samba permitirá a un servidor Unix a usar más de un nombre NetBIOS con la opción netbios aliases. Mira la Tabla 4.6.

Cuadro 4.7: Opciones de Configuración de un Servidor Virtual.

Opción	Parámetros	Función	Defecto	Ambito
netbios aliases	Lista de nombres NetBIOS	Nombres NetBIOS adicionales por los que responder, para usar con mútiples servidores "virtuales" Samba.	ninguno	Global

4.7.1. netbios aliases.

La opción netbios aliases puede ser usada para dar al servidor Samba más de un nombre NetBIOS. Cada nombre NetBIOS listado como valor será displayado en el Entorno de Red de una máquina. Cuando un conexión sea solicitada por cualquier máquina, sin embargo, esta conectará al mismo servidor Samba.

Esto sería útil si, por ejemplo, estuvieras transfiriendo los datos de tres departamentos a un único servidor Unix con modernos discos de alta capacidad, para retirar o reubicar los viejos servidores NT. Si los tres servidores se llamasen sales, accounting, y admin, podrías dejar a Samba en representación de los tres servidores con las siguientes opciones:

```
[global]
  netbios aliases = sales accounting admin
  include = /usr/local/samba/lib/smb.conf.%L
```

Mira la Figura 4.7. para ver lo que el Entorno de Red displayaría en un cliente. Cuando un cliente intente conectar a Samba, especificará el nombre del servidor al que intenta acceder, al cual puedes acceder a través de la variable %L. Si el servidor solicitado es sales, Samba incluirá el fichero */usr/local/samba/lib/smb.conf.sales*. Este fichero debería contener declaraciones globales y de recursos exclusivamente para el grupo *"sales"*, como ves aquí:

```
[global]
  workgroup = SALES
  hosts allow = 192.168.10.255
[sales1998]
  path = /usr/local/samba/sales/sales1998/
  ...
```

Este ejemplo particular establecería el grupo de trabajo a SALES, y establecería la dirección IP para permitir conexiones sólo para la subred SALES (192.168.10). En adición, ofrecería recursos específicos de ese departamento.

4.8. Opciones de Ficheros de Registro.

Ocasionalmente, necesitaremos averiguar qué está haciendo Samba. Esto es especialmente cierto cuando Samba está realizando una acción no esperada o no está funcionando bien. Para localizar esta información, necesitaremos chequear los ficheros de registro de Samba para ver porqué hizo lo que hizo.

Los ficheros de registro de Samba pueden ser tan breves o completos como tú quieras. Aquí tienes un ejemplo de cómo sería uno de ellos:

```
[1999/07/21 13:23:25, 3] smbd/service.c:close_cnum(514)
phoenix (192.168.220.101) closed connection to service IPC$
[1999/07/21 13:23:25, 3] smbd/connection.c:yield_connection(40)
Yielding connection to IPC$
[1999/07/21 13:23:25, 3] smbd/process.c:process_smb(615)
Transaction 923 of length 49
[1999/07/21 13:23:25, 3] smbd/process.c:switch_message(448)
switch message SMBread (pid 467)
[1999/07/21 13:23:25, 3] lib/doscalls.c:dos_ChDir(336)
dos_ChDir to /home/samba
[1999/07/21 13:23:25, 3] smbd/reply.c:reply_read(2199)
read fnum=4207 num=2820 nread=2820
[1999/07/21 13:23:25, 3] smbd/process.c:process_smb(615)
Transaction 924 of length 55
[1999/07/21 13:23:25, 3] smbd/process.c:switch_message(448)
switch message SMBreadbraw (pid 467)
[1999/07/21 13:23:25, 3] smbd/reply.c:reply_readbraw(2053)
readbraw fnum=4207 start=130820 max=1276 min=0 nread=1276
[1999/07/21 13:23:25, 3] smbd/process.c:process_smb(615)
Transaction 925 of length 55
[1999/07/21 13:23:25, 3] smbd/process.c:switch_message(448)
switch message SMBreadbraw (pid 467)
```

Muchas de estas opciones sólo las usan los programadores de Samba. Sin embargo, veremos el significado de algunas de estas entradas con más detalle en el *Capítulo 9, Solución de Problemas con Samba*.

Samba contiene seis opciones que permiten a los usuarios describir cómo y dónde se debería escribir la información de registro. Cada una de estas opciones son opciones globales y no pueden aparecer en la definición de un recurso. Aquí tienes un fichero de configuración actualizado que cubre cada una de las opciones que veremos:

```
[global]
  netbios name = HYDRA server
  string = Samba %v on (%I)
  workgroup = SIMPLE
  # Networking configuration options
  hosts allow = 192.168.220. 134.213.233. localhost
  hosts deny = 192.168.220.102
  interfaces = 192.168.220.100/255.255.255.0 \ 134.213.233.110/255.255.255.0
  bind interfaces only = yes
  # Debug logging information
  log level = 2
  log file = /var/log/samba.log.%m
  max log size = 50
  debug timestamp = yes
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  comment = Data Drive
  volume = Sample-Data-Drive
  writeable = yes
```

Aquí, hemos añadido un fichero de registro de usuario que reporta información de depuración de nivel 2. Este es un relativamente mínimo nivel de depuración. El rango de nivel va de 1 a 10, donde el nivel proporciona sólo una pequeña cantidad de información y el nivel 10 proporciona una cantidad enorme de información de bajo nivel. El nivel 2 nos proporcionará información útil sin ocupar demasiado espacio en disco en nuestro servidor. En la práctica, deberías evitar el uso de niveles de depuración superiores a 3 a menos que estés programando Samba.

Este fichero está localizado en el directorio */var/log* gracias a la opción log file. Sin embargo, podemos usar las variables para crear ficheros de registro independientes para cada usuario o cliente, como la variable *%m* en la siguiente línea:

```
log file = /usr/local/logs/samba.log.%m
```

La posibilidad de aislar los mensajes de registro puede resultar de un valor inapreciable a la hora de tener que rastrear un error de red si sabes que el problema viene de una máquina o cliente determinado.

Hemos añadido otra precaución a los ficheros de registro: ninguno de ellos puede exceder en tamaño de 50 kilobytes, como indica la opción max log size. Si un fichero

de registro supera este tamaño, el contenido es movido a un fichero con el mismo nombre pero con el sufijo .old añadido. Si el fichero .old ya existe, es sobreescrito y su contenido perdido. El fichero original es limpiado, a la espera de recibir nueva información de registro. Esto previene que el disco se quede sin espacio por culpa de los ficheros de registro de Samba durante el tiempo de vida de los demonios.

Por conveniencia, hemos decidio dejar la fecha y hora de la depuración en los registros con la opción debug timestamp, la cual es el valor por defecto. Esto coloca una cadena de fecha y hora en formato "timestamp" junto a cada mensaje en el fichero de registro. Si no estamos interesados en ésta opción, deberemos especificar el valor no para esta opción.

4.8.1. Usando syslog.

Si quieres usar el registro de sistema o *"system logger"* (*syslog*) en adición o en lugar del fichero de registro estándar de Samba, Samba proporciona opciones para esto también. Sin embargo, para usar *syslog*, lo primero que tienes que hacer es asegurarte de que Samba se compiló con la opción *configure –with-syslog*. Mira el *Capítulo 2* para más información sobre configuración y compilación de Samba.

Una vez está hecho esto, necesitarás configurar tu */etc/syslog.conf* para aceptar información de registro desde Samba. Si no existe todavía una entrada daemon.* en el fichero /etc/syslog.conf, añade lo siguiente:

```
daemon.* /var/log/daemon.log
```

Esto especifica que cualquier información de registro desde demonios de sistema sea almacenada en el fichero /var/log/daemon.log. Aquí es donde será almacenada a partir de ahora la información que genere Samba. Desde aquí, puedes especificar la siguiente opción global en tu fichero de configuración:

```
syslog = 2
```

Esto especifica que cualesquiera mensajes de registro con un nivel de 1 serán enviados tanto al syslog como a los ficheros de registro de Samba. (Los mapas para prioridades de syslog se describen en la sección "syslog."). Asumamos que configuramos la opción log level a 4. Cualquier mensaje de registro con un nivel 2, 3, o 4 será enviado a los ficheros de registro de Samba, pero no al syslog. Sólo los mensajes de registro de nivel 1 serán enviados a ambos. Si el valor de syslog excede del valor de log level, no se mandará nada al syslog.

Si quieres especificar que los mensajes sean enviados sólo al *syslog* -y no a los ficheros de registro propios de Samba- puedes poner esto en el fichero de configuración:

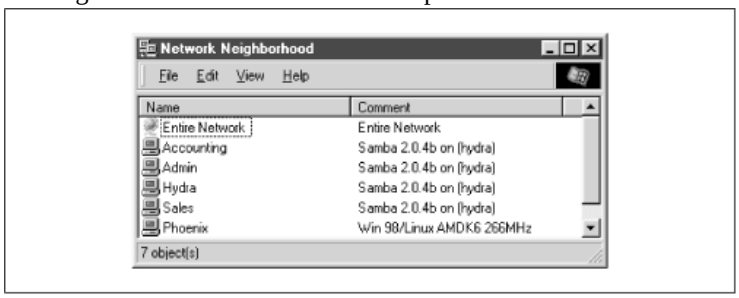
```
syslog only = yes
```

Si este es el caso, cualquier información de registro inferior al número especificado en la opción syslog será descartado, al igual que con la opción log level.

4.8.2. Opciones de Configuración de Registro.

La Tabla 4.7 lista cada una de las opciones de configuración que podemos usar.

Figura 4.7: Usando alias NetBIOS para un Servidor Samba.



Cuadro 4.8: Opciones de Configuración Globales.

Opción	**Parámetros**	**Función**	**Defecto**	**Ambito**
log file	string (nombre completamente cualificado)	Establece el nombre y localización del fichero de registro que usará Samba. Admite uso de variables.	Especificado en el makefile de Samba	Global
log level (debug level)	numérico (0-10)	Establece la cantidad de mensajes de información que serán enviados al fichero de registro. 0 es nada , 3 es considerable	1	Global
max log size	numérico (tamaño en KB)	Establece el tamaño máximo del fichero de registro. Si se excede el tamaño, el fichero será renombrado a .bak y se iniciará un nuevo fichero de registro.	5000	Global
debug timestamp (timestamp logs)	booleano	Si es no, no se incluyen fecha y hora junto a los mensajes.	yes	Global
syslog	numérico (0-10)	Establece el nivel de los mensajes enviados a syslog. Los niveles por debajo de syslog level serán enviados al registro de sistema.	1	Global
syslog only	booleano	Si es yes, usa syslog y no envía salida alguna a los ficheros de registro de Samba.	no	Global

4.8.2.1. 4.8.2.1 log file.

En nuestro servidor, Samba manda la información de registro a ficheros de texto en el sudirectorio var del directorio raíz de Samba, tal como se configuró en el fichero makefile durante la compilación. La opción log file puede usarse para cambiar el nombre y la localización del fichero de registro a otra localización. Por ejemplo, para cambiar el nombre y localización del ficher de registro de Samba a */usr/local/logs/samba.log*, podrías usar lo siguiente:

```
[global]
  log file = /usr/local/logs/samba.log
```

Puedes usar sustitución con variables para crear ficheros de registro específicos para usuarios o clientes individuales.

Puedes sobreescribir la localización del fichero usando el switch de línea de comando -l cuando el servidor es iniciado. Sin embargo, esto no prevalecerá sobre lo estipulado con la opción log file. Si especificas este parámetro, la información inicial de registro será enviada al fichero especificado tras -l (o el especificado por defecto desde el makefile) hasta que los demonios hayan procesado el fichero smb.conf y sepan que deben redirigir la información a un nuevo fichero de registro.

4.8.2.2. log level.

La opción log level establece la cantidad de información a registrar. Normalmente se mantiene a 0 ó 1. Sin embargo, si tienes un determinado problema puedes especificar el nivel 3, el cual proporciona la más útil información de depuración que podrías necesitar para resolver un problema. Los niveles por encima de 3 proporcionan información dirigida más que nada a los programadores para corregir posibles bugs, y ralentizan considerablemente el rendimiento del servidor. Por esto mismo, recomendamos que evites usar niveles por encima de 3.

```
[global]
  log file = /usr/local/logs/samba.log.%m
  log level = 3
```

4.8.2.3. max log size.

La opción max log size establece el tamaño máximo, en kilobytes, del fichero de registro que mantiene Samba. Cuando el fichero de registro excede ese tamaño, el actual fichero es renombrado y se le añade la extensión .old (eliminando ficheros anteriores con ese nomrbe) y un nuevo fichero de registro es iniciado con el nombre original. Por ejemplo:

```
[global]
  log file = /usr/local/logs/samba.log.%m
  max log size = 1000
```

Aquí, si el tamaño de cualquier fichero de configuración supera un megabyte, Samba renombra el fichero de registro samba.log. nombre-de-máquina .old y un nuevo fichero es creado. Si ya existía un fichero con la extensión .old, Samba lo elimina. Recomendamos establecer esta opción en tu fichero de configuración, porque el registro (sobre todo a bajo nivel) puede comprometer la disponibilidad de espacio en disco del servidor. El uso de esta opción protege a los administradores "despistados" de la desagradable sorpresa de descubrir repentinamente que la mayoría del espacio en disco ha sido ocupado por un simple fichero de registro de Samba.

4.8.2.4. timestamp depuración o timestamp registros.

Si estás intentando descubrir un problema de red y piensas que la información de fecha y hora dentro de los ficheros de registro forman parte de él, puedes activarlos o desactivarlos con la opción timestamp logs o la opción debug timestamp (son sinónimas) a no. Por ejemplo, un fichero típico de registro de Samba presenta sus salidas con el siguiente formato:

```
12/31/98 12:03:34 hydra (192.168.220.101) connect to server network as user davecb
```

Con un valor a no para esta opción, la salida aparecerá sin la fecha y hora:

```
hydra (192.168.220.101) connect to server network as user davecb
```

4.8.2.5. syslog.

La opción syslog provoca que los mensajes de registro de Samba sean enviados al registro de sistema de Unix. El tipo de información que va a ser enviada se especifica como parámetro para este argumento. Al igual que con la opción log level, este puede ser un número desde 0 a 10. La información de registro con un nivel inferior al número especificado será enviada al registro de sistema. Sin embargo, registros de nivel igual o inferior al nivel de syslog, pero menores que los de la opción log level, seguirán siendo enviados a los ficheros de registro Samba. Usa la opción syslog only. Por ejemplo:

```
[global]
  log level = 3
  syslog = 1
```

Con esto, toda la información de registro con un nivel de 0 debería ser enviada a los ficheros de registro de Samba y al del registro de sistema, mientras que la información de nivel 1, 2, y 3 se enviaría sólo a los ficheros de registro de Samba. Los niveles por debajo de 3 no son registrados. Advierte que todos los mensajes enviados al registro de sistema son mapeados a un nivel de prioridad que el proceso syslog entiende, como se muestra en la Tabla 4.8. El nivel por defecto es 1.

Cuadro 4.9: Conversión de Prioridad de Syslog.

Log Level	**Prioridad Syslog**
0	LOG_ERR
1	LOG_WARNING
2	LOG_NOTICE
3	LOG_INFO
4 y anteriores	LOG_DEBUG

Si vas a usar syslog, tendrás que ejecutar configure –with-syslog cuando compiles Samba, y necesitarás configurar tu /etc/syslog.conf también. (Mira la sección 4.8.1., "*Usando syslog*", en éste capítulo).

4.8.2.6. syslog only.

La opción syslog only le dice Samba que no use sus ficheros normales de registro -sólo el de registro de sistema-. Para activar esto, especifica la siguiente opción en la sección global del fichero de configuración de Samba:

```
[global]
  syslog only = yes
```

Capítulo 5

Visualización (Browsing) y Compartición Avanzada de Discos.

Este capítulo continúa nuestra discusión sobre la compartición de unidades de disco del capítulo anterior. Aquí, discutiremos varias diferencias entre los sistemas de archivos de Windows y Unix -y cómo Samba realiza la conversión entre ellos-. Existe un sorprendente número de inconsistencias entre un sistema de archivos DOS y un sistema de archivos Unix. En adición, hablaremos brevemente sobre la conversión de nombres, bloqueo de archivos, y una relativamente nueva característica de Samba: bloqueo oportunístico, o 'oplocks'. Sin embargo, antes de movernos por ese territorio, primero deberíamos meternos con la base de la visualización, navegación o 'browsing' con Samba.

5.1. Visualización, Navegación o 'Browsing'.

La visualización es la habilidad de examinar los servidores y recursos que están actualmente disponibles en tu red. Sobre un cliente Windows NT 4.0 o 95/98, un usuario puede visualizar o navegar por los servidores de la red a través del 'Entorno de Red'. Haciendo doble click sobre el icono que representa al servidor, el usuario debería poder ver los recursos de disco y/o de impresión disponibles en esa máquina.

Desde la línea de comandos de Windows, también puedes usar la opción net view para ver qué servidores están actualmente disponibles en la red. Aquí tienes un ejemplo del comando net view en acción:

```
C:\>
net view
Servers available in workgroup SIMPLE
Server name Remark
------------------------------------------------
\\CHIMAERA Windows NT 4.0
\\HYDRA Samba 2.0.4 on (hydra)
\\PHOENIX Windows 98
```

5.1.1. Prevención contra la Visualización.

Puedes restringir la aparición de un recurso en una lista de visualización usando la opción browseable. Esta opción de tipo booleana evita que un recurso sea visible

desde el Entorno de Red. Por ejemplo, para prevenir que el recurso del capítulo anterior nominado [data] sea visible, podríamos escribir esto:

```
[data]
  path = /home/samba/data
  browseable = no
  guest ok = yes
  comment = Data Drive
  volume = Sample-Data-Drive
  writeable = yes
```

Aunque normalmente no querrás hacer esto en un típico recurso de disco, la opción 'browseable' resulta útil en el caso de que necesites crear un recurso con contenidos que no quieras que vean otros, tales como un recurso [netlogin] para almacenar scripts de logeado para el control de un dominio Windows (mira el *Capítulo 6, Usuarios, Seguridad y Dominio*s, para más información sobre los scripts de logeado).

Otro ejemplo es el recurso [homes]. Este recurso es frecuentemente marcado como no browsable para que un recurso llamado [homes] no aparezca cuando los recursos de la máquina son interrogados. Sin embargo, si un usuario alice se logea y mira por los recursos de la máquina, le aparecerá una carpeta denominada [alice]. ¿Y qué pasa si nos queremos asegurar de que el recurso de alice sea visible para todo el mundo antes de que ella se logee? Esto se puede hacer con la opción global auto services. Esta opción precarga los recursos en la lista de visualización para asegurarnos de que estos estén siempre visibles:

```
[global]
  ...
  auto services = alice
  ...
```

5.1.2. Servicios por Defecto.

En el caso de que un usuario no pueda conectar a un recurso, puedes especificar un recurso por defecto al que él podrá conectar. Ya que no sabes quién tendrá que acceder a este recurso por defecto en cualquier momento, probablemente querrás establecer la opción guest ok a yes para este recurso. El especirficar un servicio por defecto puede ser útil cuando envíes a un usuario perdido al directorio de ficheros de ayua. Por ejemplo:

```
[global]
  ...
  default service = helpshare
  ...
[helpshare]
  path = /home/samba/helpshare/%S
  browseable = yes
  guest ok = yes
  comment = Default Share for Unsuccessful Connections
  volume = Sample-Data-Drive
  writeable = no
```

Advierte que hemos usado la variable %S en la opción path. Si usas la variable %S, esta se referrirá al recurso solicitado no existente (el recurso originariamente solicitado por el usuario), no el nombre del recurso por defecto resultante. Esto nos permite crear diferentes rutas con los nombres de cada servidor, los cuales pueden proporcionar más definidos ficheros de ayuda para los usuarios. En adición, cualesquiera signos de subrayado (_) especificados en el recurso solicitado serán convertidos a barras (/) cuando se use la variable %S.

5.1.3. Elecciones de Visualizadores.

Como ya mencionamos en el Capítulo 1, Aprendiendo Samba, una máquina en cada subred siempre mantiene una lista de las actuales máuinas activas en la red. Esta

lista es denominada lista de visualización o lista de navegación, y el servidor que la mantiene es llamado visualizador maestro local. Como las máquinas se encienden y apagan contínuamente en una red, el visualizador maestro local contínuamente actualiza la información en la lista de visualización y se la proporciona a cualquier máquina que la solicite.

Una computadora se convierte en visualizador maestro local manteniendo una elección de visualización en la subred local. Las elecciones de visualizadores pueden ser llamadas en cualquier momento. Samba puede usar una elección de visualizador para una variedad de resultados, incluyendo el ser siempre el visualizador maestro local de la subred o no serlo nunca. Por ejemplo, las siguientes opciones, que hemos añadido al fichero de configuración del *Capítulo 4, Recursos de Disco*, asegurará que Samba siempre gane la elección de visualizador maestro local, sin importar qué otras máquinas estén presentes en la red:

```
[global]
  netbios name = HYDRA server
  string = Samba %v on (%L)
  workgroup = SIMPLE
  # Browsing election options
  os level = 34
  local master = yes
  # Networking configuration options
  hosts allow = 192.168.220. 134.213.233. localhost
  hosts deny = 192.168.220.102
  interfaces = 192.168.220.100/255.255.255.0 \
               134.213.233.110/255.255.255.0
  # Debug logging information
  log level = 2
  log file = /var/log/samba.log.%m
  max log size = 50
  debug timestamp = yes
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  comment = Data Drive
  volume = Sample-Data-Drive
  writable = yes
```

Sin embargo, ¿Qué pasa si quiero que nunca gane esa elección? ¿Qué pasa si quiero que el ganador siempre sea un servidor Windows NT que tengo en esa red? Para hacer esto, necesitamos aprender cómo trabajan las elecciones de visualizador. Como ya sabes, cada máquina que toma lugar en la elección debe mandar información de sí misma por el método de difusión (broadcast). Esta información incluye lo siguiente:

- La versión del protocolo de elección usada.
- El sistema operativo de la máquina.
- La cantidad de tiempo que el cliente ha permanecido en la red.
- El nombre de máquina.

Aquí es donde se decide la elección. Los sistemas operativos tienen asignado un valor binario en función de su versión, como puedes ver en la Tabla 5.1.

A continuación, a cada computadora en la red se le asigna un valor distinto de acuerdo a su papel o rol en la red, como en la Tabla 5.2.

Las elecciones se realizan de la siguiente forma:

1. La máquina con la mayor versión de protocolo de elección ganará (todos los clientes Windows tienen la versión 1 del protocolo de elección).
2. La máquina con el mayor valor de sistema operativo ganará la elección.

3. Si hay conflicto, la máquina con el valor 'Visualizador Maestro Preferido' (rol 8) ganará la elección.

4. Si todavía hay empate o conflicto, el cliente que haya estado más tiempo en línea ganará la elección.

5. Y, finalmente, si todavía existe conflicto, el nombre de cliente que por orden alfabético vaya primero ganará.

6. La máquina que ocupó el segundo lugar en la elección, puede convertirse en visualizador de seguridad.

Como resultado, si quieres que Samba tome el rol de visualizador maestro local, pero sólo en el caso de que no haya un servidor Windows NT Server (4.0 o3.51) en la red, podrías cambiar el parámetro os level en el ejemplo anterior por:

```
os level = 31
```

Esto provocará que Samba inmediatamente pierda la elección contra un servidor Windows NT 4.0 o Windows NT 3.5 Server, los cuales tienen un nivel de s.o. superior. Por otro lado, si quieres decidir al visualizador maestro local en base al rol de red, tal como qué máquina es el controlador primario de dominio, podrías establecer el valor de os level para que coincidiese con el tipo más alto de sistema operativo en la red, y permitir así que la elección de protocolo pasara al siguiente nivel.

¿Cómo puedes saber si una máquina es el visualizador maestro local? Pues usando el comando nbtstat. Coloca el nombre NetBIOS de la máquina que quieras interrogar tras la opción -a:

```
C:\>
nbtstat -a hydra
  NetBIOS Remote Machine Name Table
Name               Type         Status
---------------------------------------------
HYDRA           <00> UNIQUE     Registered
HYDRA           <03> UNIQUE     Registered
HYDRA           <20> UNIQUE     Registered
..__MSBROWSE_ _.    <01> GROUP      Registered
SIMPLE          <00> GROUP      Registered
SIMPLE          <1D> UNIQUE     Registered
SIMPLE          <1E> GROUP      Registered
MAC Address = 00-00-00-00-00-00
```

La línea que estás buscando es *.._ _MSBROWSE_ _.<01>*. Esto indica que el servidor está actualmente actuando como el visualizador maestro local para la subred actual. En adición, si la máquina es un servidor Samba, puedes chequear el fichero de registro de Samba nmbd para buscar una entrada como ésta:

```
nmbd/nmbd_become_lmb.c:become_local_master_stage2(406)
*****
Samba name server HYDRA is now a local master browser for
workgroup SIMPLE on subnet 192.168.220.100
****
```

Finalmente, los servidores Windows NT que están sirviendo como controladores primarios de dominio contienen un chivato que les permite asumir el rol de visualizador maestro local en ciertas condiciones; esto es llamado el bit de visualizador maestro preferido. Antes, mencionamos que Samba podía asignarse este bit para sí. Puedes activarlo con la opción preferred master:

```
# Browsing election options
os level = 33
local master = yes
preferred master = yes
```

Si se configura el bit de maestro preferido, la máquina forzará una elección de visualizador al arrancar. Por supuesto, esto sólo es necesario si estableces la opción os level para que coincida con la máquina Windows NT. Te recomendamos que no uses ésta opción si otra máquina tiene también el rol de maestro preferido, tal como un NT server.

5.1.4. Visualizador Maestro de Dominio.

En el capítulo de inicio, mencionamos que para que un grupo de trabajo o dominio de Windows se extienda en múltiples subredes, una máquina debería asumir el rol de visualizador maestro de dominio. El visualizador maestro de dominio propaga listas de visualización por cada una de las subredes del grupo de trabajo. Esto es posible porque cada visualizador maestro local periódicamente sincroniza su lista de visualización con el visualizador maestro de dominio. Durante la sincronización, el visualizador maestro local pasa sobre cada servidor que el visualizador maestro de dominio no tenga en su lista de visualización, y viceversa. En un mundo perfeceto, cada visualizador maestro local debería tener la lista de visualización para el dominio entero.

A diferencia del visualizador maestro local,no hay proceso de elección para determinar qué máquina asume el rol de visualizador maestro de dominio. en su lugar, el administrador de la red debe establecerlo manualmente. Debido al diseño de Microsoft, sin embargo, tanto el visualizador maestro de dominio como el controlador primario de dominio (PDC) registran un tipo de recurso de <1B>, de forma que los roles -y las máquinas- son inseparables.

Si tienes un servidor Windows NT server en la red actuando como PDC, te recomendamos que no uses Samba como visualizador maestro de dominio. Lo contrario también es cierto: si Samba toma las responsabilidades de actuar como PDC, te recomendamos que lo conviertas también en visualizador maestro de dominio. Aunque es posible separar los roles con Samba, esto no es una buena idea. Usar dos máquinas diferentes para servir, una como PDC y otra como visualizador maestro de dominio, puede causar errores aleatorios en un grupo de trabajo Windows.

Samba puede asumir el rol de visualizador maestro de dominio para todas las subredes en el grupo de trabajo con la siguiente opción:

```
domain master = yes
```

Puedes verificar que la máquina Samba es de hecho el visualizador maestro de dominio chequeando el fichero nmbd:

```
nmbd/nmbd_become_dmb.c:become_domain_master_stage2(118)
*****
Samba name server HYDRA is now a domain master browser for
workgroup SIMPLE on subnet 192.168.220.100
*****
```

O puedes usar el comando nmblookup que viene con Samba para preguntar por un único tipo de recurso <1B>en el grupo de trabajo:

```
#

nmblookup SIMPLE#1B
Sending queries to 192.168.220.255
192.168.220.100 SIMPLE<1b>
```

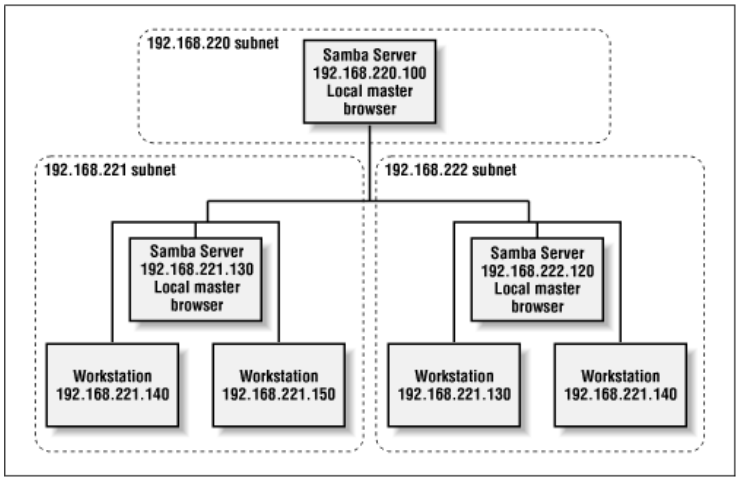
5.1.4.1. Múltiples Subredes.

Hay tres reglas que deber recordar cuando crees un grupo de trabajo o dominio que abarque más de una subred:

- Debes tener o un servidor Windows NT o una máquina Samba actuando como visualizador maestro local en cada subred del grupo de trabajo o dominio (si tienes un visualizador maestro de dominio en una subred, un visualizador maestro local no es necesario).
- Debes tener un servidor Windows NT o una máquina Samba actuando como visualizador maestro de dominio en algún punto del grupo de trabajo.
- Cada visualizador maestro local debe ser instruido para que se sincronice con el visualizador maestro de dominio.

Samba tiene algunas otras características en éste área para el caso de que no tengas o no quieras tener un visualizador maestro de dominio en tu red. Considera las subredes de la Figura 5.1.

Figura 5.1: Múltiples subredes con servidores Samba.



Primero, un servidor Samba que es visualizador maestro local puede usar la opción de configuración remote announce para asegurarse de que las computadoras de diferentes subredes estén enviando mensajes de difusión para el servidor. Esto tiene el efecto de garantizar que el servidor Samba aparezca en las listas de visualización de subredes foráneas. Para asegurarte de esto, sin embargo, las transmisiones dirigidas deben alcanzar al visualizador maestro local de la otra subred. Sé consciente de que muchos routers no permiten mensajes de difusión por defecto; puede que tengas que cambiar

esta característica en el router para que los mensajes de difusión circulen hacia la otra subred.

Con la opción remote announce, listas las subredes y el grupo de trabajo que deberían recibir el broadcast. Por ejemplo, para asegurarte de que las máquinas en las subredes 192.168.221 y 192.168.222 y el grupo de trabajo SIMPLE están enviando información de difusión desde nuestro servidor Samba, podríamos especificar lo siguiente:

```
# Browsing election options
os level = 34
local master = yes
remote announce = 192.168.221.255/SIMPLE \

                  192.168.222.255/SIMPLE
```

Además, puedes especificar la dirección exacta para el envío de los broadcasts si el visualizador maestro local de la red froánea tiene una dirección IP fija.

Un visualizador maestro local Samba puede sincronizar su lista de visualización directamente con otro servidor Samba que esté actuando como visualizador maestro local en una subred diferente. Por ejemplo, asumamos que Samba está configurado como un visualizador maestro local, y que existen visualizadores maestros locales Samba en 192.168.221.130 y 192.168.222.120. Podemos usar la opción remote browse sync para sincronizarse directamente con los servidores Samba, como sigue:

```
# Browsing election options
os level = 34
local master = yes
remote browse sync = 192.168.221.130 192.168.222.120
```

Para que esto funcione, las otras máquinas Samba deben también ser visualizadores maestros locales. También puedes usar broadcasts directos con esta opción si no conoces la dirección IP específica de los visualizadores maestros locales.

5.1.5. Opciones de Visualización.

La Tabla 5.3 muestra 14 opciones que definen cómo Samba realiza las tareas de visualización. Te recomendamos los valores por defecto para un sitio que prefieras sea sencillo de usar por los usuarios a la hora de localizar recursos de disco e impresión.

5.1.5.1. announce as.

Esta opción de configuración global especifica el tipo de sistema operativo con el que Samba será anunciado a las otras máquinas de la red. El valor por defecto para ésta opción es NT, el cual representa a un s.o. Windows NT. Otros posibles valores son Win95, que representa a s.o. Windows 95, y WfW para sistemas del tipo Windows para Trabajo en Grupo. Puedes cambiar el valor por defecto con lo siguiente:

```
[global]
  announce as = Win95
```

De nuevo te recomendamos. No cambies el valor por defecto para esta opción.

5.1.5.2. announce version.

Esta opción global es frecuentemente usada con la opción de configuración announce as; especifica la versión del s.o. con el que Samba se anunciará a las otras máquinas de la red. El valor por defecto es 4.2, el cual la coloca por encima de la actual versión 4.0 de Windows NT. Puedes especificar un nuevo valor con una entrada global como la siguiente:

```
[global]
announce version = 4.3
```

De nuevo, recomendación. No cambies el valor por defecto para esta opción.

5.1.5.3. browseable.

La opción browseable (también conocida como browsable) indica si el recurso referenciado debería aparecer en la lista de recursos disponibles de la máquina en la cual residen. Esta opción siempre está establecida a yes por defecto. Si quieres evitar que un recurso sea visible para un cliente, cambia este valor a no.

Advierte que esto no previene contra el acceso al recurso usando otros métodos, tales como, especifcar una completa ruta UNC (//server/accounting) en el Explorador de Windows. Sólo evita que el recurso sea listado bajo la lista de recursos de la máquina investigada.

5.1.5.4. browse list.

Nunca deberías necesitar cambiar el valor por defecto de este parámetro, que es yes. Si tu servidor Samba está actuando como visualizador maestro local (p.ej., ha ganado la elección de visualizador), puedes usar la opción global browse list para instruir a Samba para proporcionar o negar su lista de visualización a todos los clientes. Por defecto, Samba siempre proporciona una lista de visualización. Puedes denegar esta información especificando lo siguiente:

```
[global]
  browse list = no
```

Si desactivas la lista de visualización, los clientes no pueden visualizar los nombres de las otras máquinas, sus servicios, y otros dominios actualmente disponibles en la red. Advierte que esto no hace a ninguna máquina inaccesible; si alguien conoce una dirección o nomrbe de máquina válida, todavía podrá conectar a ella usando NET USE o mapeando una unidad de red a ella usando el explorador de Windows. Simplemente evita que la información de la lista de visualización sea recibida por cualquier cliente que la solicite.

5.1.5.5. auto services.

La opción global auto services, la cual se llama también como preload , asegura que los recursos especificados están siempre visibles en la lista de visualización. Un uso típico para esta opción es advertir sobre determinados recursos, que están creados por los recursos [homes] o [printers], pero que no son visualizables.

Esta opción trabaja mejor con los recursos de disco. Si deseas forzar a cada una de las impresoras de tu sistema (p.ej., aquellas listadas en el fichero de capacidades de

impresoras) a que aparezcan en la lista de visualización usando esta opción, te recomendamos usar la opción load printers en su lugar. Cualesquiera recursos listados con la opción auto services no será displayada si la opción browse list se establece al valor no.

5.1.5.6. default service.

La opción global default service (a veces llamada default) denomina a un recurso del tipo 'last-ditch'. Si se establece a un nombre de recurso ya existente, y un cliente solicita un recurso de disco o de impresión no existente, Samba intentará conectar al usuario al recurso especificado por ésta opción en su lugar. La opción se especifica como sigue:

```
default service = helpshare
```

Advierte que no hay corchetes rodeando el nombre del recurso helpshare. Además, si usas la variable %S en el recurso especificado por esta opción, ésta representará al solicitado, y no existente, recurso, no al servicio por defecto. Cualesquiera signos de subrayado (_) especificados en el recurso solicitado serán convertidos a barras (/) cuando se use la variable.

5.1.5.7. local master.

Esta opción global especifica si Samba intentará convertirse en el visualizador maestro local para la subred cada vez que arranque Samba. Si esta opción se pone a yes, Samba tomará lugar en las elecciones. Sin embargo, el hecho de establecer esta opción por sí misma no garantiza la victoria (otros parámetros, tales como preferred master y os level ayudan a Samba con las elecciones de visualizador). Si esta opción se establece a no, Samba perderá todas las elecciones de visualizador, sin importar qué valores estén especificados en las demás opciones de configuración. El valor por defecto es yes.

5.1.5.8. lm announce.

La opción global lm announce le dice al demonio nmbd de Samba si enviar o no anuncios de máquinas 'LAN Manager' en nombre del servidor Estos anuncios de máquinas pueden ser requeridos por viejos clientes, tales como sistemas IBM OS/2. Estos anuncions permiten que el servidor sea añadido a la lista de visualización del cliente. Si es activado, Samba se anunciará a sí mismo repetitivamente el número de segundos especificados por la opción lm interval.

Esta opción de configuración toma valoresboleeanos, yes y no, para la activación/desactivación de los anuncions LAN Manager, respectivametne. En adición, hay una tercera opción, auto, la cual causa que nmbd pasivamente escuche anuncios LAN Manager, pero que no envíe ninguno inicialmente. Si se detectan anuncions LAN Manager para otra máquina de la red, nmbd comenzará enviando sus propios anuncions LAN Manager para asegurarse de que es visible. Puedes especificar la opción como sigue:

```
[global]
  lm announce = yes
```

El valor por defecto es auto. Probablemente no necesitarás cambiar este valor por defecto.

5.1.5.9. lm interval.

Esta opción, que es usada en conjunción con lm announce, indica el número de segundos que nmbd esperará antes de emitir en difusión anuncions LAN Manager-style. Recuerda que los anuncios LAN Manager deben ser activados para que esta opción pueda ser usada. El valor por defecto es 60 segundos. Si estableces este valor a 0, Samba no enviará ninguno, sin importar el valor de la opción lm announce. Puedes resetear el valor para ésta opción como sigue:

```
[global]
  lm interval = 90
```

5.1.5.10. preferred master.

La opción preferred master solicita que Samba establezca el bit de maestro preferido cuando participe en una elección. Esto le da al servidor un status más alto en el grupo de trabajo con respecto al resto de máquinas al mismo nivel de s.o. Si estás configurando tu máquina Samba para convertirse en el visualizador maestro, debes establecer el siguiente valor:

```
[global]
  preferred master = yes
```

En caso contrario, deberías dejar su valor por defecto, no. Si Samba está configurado como visualizador maestro preferido, forzará un proceso de elección cada vez que el servidor se active en la red.

5.1.5.11. os level.

La opción global os level indica el nivel de sistema operativo que Samba adoptará en un proceso de elección de visualizador. Si quieres que Samba gane la elección de visualizador maestro, debes establecer el primer nivel superior de s.o. al de cualesquiera otros que puedan existir en tu red. Los valores se muestran en la Tabla 5-1. El valor por defecto es 0, lo que significa que Samba perderá todas las elecciones. Si quieres que las gane todas, debes variar el valor como sigue:

```
os level = 34
```

Esto significa que el servidor votará por sí mismo 34 veces cada vez que se produzcan elecciones, lo cual le garantiza la victoria.

5.1.5.12. domain master.

Si Samba es el controlador primario de dominio para tu grupo de trabajo o tu dominio NT, también debería ser el visualizador maestro de dominio. El visualizador maestro de dominio es una máquina especial que tiene el tipo de recurso NetBIOS <1B>y es usada para propagar listas de visualización a cada uno de los visualizadores maestros locales existentes en las subredes del dominio. Para forzar a Samba a convertirse en el visualizador maestro de dominio, establece lo siguiente en la sección [global] del smb.conf:

```
[global]
  domain master = yes
```

Si tienes un Windows NT server en la red actuando como controlador primario de dominio (PDC), te recomendamos que no uses Samba como visualizador maestro de dominio. Lo contrario también es cierto: Si Samba tiene las responsabilidades de un PDC, te recomendamos que lo hagas también visualizador maestro de dominio. Separar la característica de PDC y de visualizador maestro de dominio provocará errores impredecibles en la red.

5.1.5.13. remote browse sync.

La opción global remote browse sync especifica que Samba debería sincronizar sus listas de visualización con los demás visualizadores maestros en otras subredes. Sin embargo, la sincronización puede ocurrir sólo con otros servidores Samba, y no con computadoras Windows. Por ejemplo, si tu servidor Samba fuese el visualizador maestro de la subred 192.168.235, y existiesen otros visualizadores maestros locales Samba en otras subredes como 192.168.234.92 y 192.168.236.2, pondrías lo siguiente:

```
remote browse sync = 192.168.234.92 192.168.236.2
```

El server Samba podría entonces contactar con las otras máquinas de la lista de direcciones y sincronizar listas de visualización. También puedes poner:

```
remote browse sync = 192.168.234.255 192.168.236.255
```

Esto fuerza a Samba a hacer un broadcast de peticiones para determinar las direcciones IP de los visualizadores maestros locales en cada subred, y entonces poder realizar la sincronización. Esto sólo funciona, sin embargo, si tu router no está configurado para bloquear peticiones de broadcast que terminen en 255.

5.1.5.14. remote announce.

Los servidores Samba son capaces de proporcionar listas de visualización a redes externas con la opción remote announce. Estas son normalmente enviadas al visualizador maestro local de la subred externa en cuestión. Sin embargo, si no conoces la dirección de ese visualizador maestro local, puedes hacer lo siguiente:

```
[global]
  remote announce = 192.168.234.255/ACCOUNTING 192.168.236.255/ACCOUNTING
```

Con esto, Samba enviará anuncios de máquina en difusión (broadcast) a todas las máquinas de las subredes 192.168.234 y 192.168.236, en las cuales responderán los visualizadores maestros locales de cada subred. También puedes especificar las direcciones IP, si las conoces.

5.2. 5.2 Diferencias entre Sistemas de Ficheros.

Uno de los mayores problemas que Samba tiene que superar es la diferencia entre sistemas de ficheros Unix y no-Unix. Esto incluye cosas como manejar enlaces simbólicos, archivos ocultos, y ficheros de configuración (ficheros con 'punto'). Además, los permisos de ficheros también pueden convertirse en un dolor de cabeza si no han sido tenidos en cuenta. Eta sección describe cómo usar Samba para controlar algunas de estas diferencias, e incluso cómo agregar algunas nuevas funcionalidades.

5.2.1. Ficheros Ocultos y Vetados.

Existen algunos casos en los que necesitamos asegurarnos de que un usuario no pueda acceder o ver determinados archivos. Otras veces, no queremos que un usuario pueda acceder al archivo -queremos ocultarlo cuando acceda a un directorio-. En sistemas Windows, un atributo de los ficheros permite ocultarlos en un listado de archivos. Con Unix, la manera tradicional de ocultar archivos en un directorio es precederlos de un punto (.). Esto evita que determinados archivos, como los de configuración, sean visibles ante la ejecución de un típico comando ls. Prohibir a un usuario el acceso a un fichero, sin embargo, implica trabajar con permisos sobre ficheros y/o directorios.

La primera opción de Samba que deberíamos discutir es la booleana hide dot files. Esta opción hace exactamente lo que dice. Cuando se establece a yes, la opción trata a los ficheros antecedidos por un punto (.) como ocultos. Si se establece a no, esos ficheros son siempre visualizados. Lo importante a recordar aquí es que los ficheros sólo están ocultos. Si el usuario ha seleccionado mostrar los ficheros ocultos (p.ej., usando la opción de menú Ver Archivos Ocultos en un cliente Windows 98), todavía serán visibles, independientemente del valor de ésta opción, como se ve en la Figura 5.2.

Figura 5.2: Archivos ocultos en el recurso [data].

En vez de simplemente ocultar los archivos que empiecen por un punto, puedes también especificar especificar un patrón de cadena para que Samba oculte determinados archivos, usando la opción hide files. Por ejemplo, supongamos que hemos especificado lo sigueinte en nuestro recurso de ejemplo [data]:

```
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  writeable = yes
  case sensitive = no
  hide files = /*.java/*README*/
```

Cada entrada para ésta opción debe comenzar, terminar, o estar separada de otra con una barra (/), aunque sólo se liste un patrón. Esta convención permite que los espacios aparezcan en los nombres de archivos. En este ejemplo, el directorio compartido debería aparecer como ves en la Figura 5.3. De nuevo, advierte que hemos seleccionado la opción de ver archivos ocultos en nuestro cliente Windows.

Si queremos evitar que de ningún modo los usuarios puedan ver los archivos, podemos usar en su lugar la opción veto files. Esta opción, que usa la misma sintaxis que

Cuadro 5.1: Valores de los S.O. en una Elección.

S.O.	**Valor**
Windows NT Server 4.0	33
Windows NT Server 3.51	32
Windows NT Workstation 4.0	17
Windows NT Workstation 3.51	16
Windows 98	2
Windows 95	1
Windows 3.1 for Workgroups	1

Cuadro 5.2: Valores de Roles de Computadoras en una Elección.

Rol	**Valor**
Controlador Primario de Dominio	128
Cliente WINS	32
Visualizador Maestro Preferido	8
Visualizador Maestro Activo	4
Visualizador en Espera	2
Visualizador de Seguridad Activo	1

Figura 5.3: Archivos ocultos en base a patrones de nombres de archivos.

Cuadro 5.3: Opciones de Configuración de Visualización.

Opción	**Parámetros**	**Función**	**Defecto**	**Ambito**
announce as	NT o Win95 o Wf W	Establece el s.o. como el que Samba se anunciará a sí mismo.	NT	Global
announce version	numérico	Establece la versión del s.o. como el que Samba se anunciará a sí mismo.	4.2	Global
browseable (browsable)	booleano	Permite a los recursos ser displayados en la lista de recursos de máquinas.	yes	Recurso
browse list	booleano	Si es yes, Samba proporcionará una lista de navegación en este servidor.	yes	Global
auto services (preload)	string (lista recursos)	Establece un lista de recursos que siempre aparecerán en la lista de navegación.	Ninguno	Global
default service (default)	string (nombre recurso)	Da nombre a un recurso (service) que será proporcionado si el cliente solicita acceso a un recurso no listado en smb.conf.	Ninguno	Global
local master	booleano	Si es yes, Samba intentará convertirse en visualizador maestro en la subred local.	yes	Global
lm announce	yes o no o auto	Activa o desactiva anuncios de máquinas del tipo LAN Manager.	auto	Global
lm interval	numérico	Especifica la frecuencia en segundos en que los anuncions "LAN Manager" serán realizados si son activados.	60	Global
preferred master (prefered master)	booleano	Si es yes, Samba usará el bit de visualizador maestro prefereido para intentar convertirse en el visualizador maestro local.	no	Global
domain master	booleano	Si es yes, Samba intentará convertirse en el principal visualizador maestro para el grupo de trabajo.	no	Global
os level	numérico	Establece el nivel de s.o. de Samba en una elección para visualizador maestro local.	0	Global
remote browse sync	string (lista de direcciones IP)	Lista los servidores Samba para sincronizar con sus listas de navegación.	Ninguno	Global
remote announce	string (Dir. IP /grupos de trabajo)	Lista subredes y grupos de trabajo a las que enviar paquetes de difusión, permitiendo a Samba aparecer para visualizar listas.	Ninguno	Global

la opción hide files, especifica una lista de ficheros que nunca deberían ser vistos por el usuario. Por ejemplo, cambiemos el recurso [data] como sigue:

```
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  writeable = yes
  case sensitive = no
  veto files = /*.java/*README*/
```

La sintaxis de esta opción es idéntica a la opción de configuración hide files: cada entrada debe comenzar, terminar o ser separada de otra entrada por una barra (/), aunque sólo halla un patrón. Haciendo esto, los ficheros hello.java y README simplemente desaparecerán del directorio, y el usuario no podrá acceder a ellos a través de SMB.

Hay otra cuestión que necesitamos controlar. ¿Qué ocurre si el usuario intenta borrar un directorio que contiene archivos vetados? Aquí es donde entra en juego la opción delete veto files. Si esta opción booleana se establece a yes, al usuario se le permitirá eliminar tanto los archivos normales como los que se encuentran vetados en el directorio, y el directorio mismo será eliminado. Si la opción se establece a no, el usuario no podrá eliminar los archivos vetados, y consecuentemente no serán eliminados. Desde la perspectiva del usuario, el usuario parecerá estar vacío, pero el directorio no podrá ser eliminado.

La directiva dont descend especifica una lista de directorios cuyos contenidos Samba no debería permitir que fueran visibles. Advierte que hemos dicho contenidos, no directorio. Los usuarios podrán entrar en el directorio, pero no podrán descender por el árbol del directorio -siempre verán una carpeta vacía-. Por ejemplo, usemos esta opción con una forma más básica del recurso definido al principio:

```
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  writeable = yes
  case sensitive = no
  dont descend = config defaults
```

En adición, asumamos que el directorio /home/samba/data tiene los siguientes contenidos:

```
drwxr-xr-x 6 tom users 1024 Jun 13 09:24 .
drwxr-xr-x 8 root root 1024 Jun 10 17:53 ..
-rw-r--r-- 2 tom users 1024 Jun 9 11:43 README
drwxr-xr-x 3 tom users 1024 Jun 13 09:28 config
drwxr-xr-x 3 tom users 1024 Jun 13 09:28 defaults
drwxr-xr-x 3 tom users 1024 Jun 13 09:28 market
```

Si el usuario entonces conecta al recurso, él o ella deberían ver los directorios mostrados en la Figura 5.4. Sin embargo, los contenidos de los directorios /config y /defaults aparecerían vacíos para el usuario, aunque existieran otras carpetas o archivos en ellos. En adición, los usuarios no podrán escribir datos en la carpeta (lo cual evita la posibilidad de crear un archivo con el mismo nombre que uno ya existente, pero

invisible). Si un usuario intentase hacerlo, él o ella recibiría un mensaje de 'Acceso Denegado'. dont descend es una opción administrativa, no de seguridad, y no es sustituta de una buena política de permisos de archivos.

Figura 5.4: Contenidos del recurso [data] con 'dont descend'.

5.2.2. Enlaces.

Los sistemas de ficheros DOS y NT no tienen enlaces simbólicos; los sistemas Windows 95/98/NT se aproximan a ellos con sus 'Accesos Directos'. Así, cuando un cliente intenta abrir un enlace simbólico en un directorio de un recurso compartido por un servidor Samba, Samba intenta seguir el enlace al archivo verdadero y permite al cliente abrirlo, como si estuviera en una máquina Unix. Si no quieres permitir esto, configura la opción follow symlinks:

```
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  writeable = yes
  case sensitive = no
  follow symlinks = no
```

Puedes testear esto creando un directorio en el servidor Unix dentro del recurso como el usuario que va a acceder a él. Introduce los siguientes comandos:

```
%
mkdir hello; cd hello %
cat "This is a test" >hello.txt %
ln -s hello.txt "Link to hello"
```

Esto resulta en los dos archivos mostrados en la Figura 5.5. Normalmente, si haces click sobre cualquiera de ellos, recibirás un archivo que tiene el texto 'This is a test' dentro de él. Sin embargo, con la opción follow symlinks establecida a no, deberías recibir un error similar al de la caja de diálogo de la Figura 5.5 si hicieras click sobre 'Link to hello.'

Finalmente, hablaremos de la opción wide links. Esta opción, si está a yes, permite al usuario cliente seguir enlaces simbólicos que apunten fuera del árbol del recurso, incluyendo archivos o directorios en la otra parte del enlace. Por ejemplo, asumamos que hemos modificado el recurso [data] como sigue:

```
[data]
  path = /home/samba/data
  browseable = yes guest
  ok = yes
  writeable = yes
  case sensitive = no
  follow symlinks = yes
  wide links = yes
```

Con tal de que la opción follow symlinks esté activada, esto causará que Samba siga todos los enlaces simbólicos fuera del actual árbol del recurso. Si creamos un archivo fuera del recurso (por ejemplo, en algún directorio 'home' de usuario) y luego creamos un enlace a él en el recurso como sigue:

```
ln -s ~tom/datafile ./datafile
```

entonces podrás abrir el fichero del directorio de Tom, en función de los permisos que tenga el archivo.

5.2.3. Opciones de Sistemas de Archivos.

La Tabla 5.4 muestra un resumen de las opciones que hemos discutido. Recomendamos los valores por defecto para la mayoría, excepto para aquéllas listadas en las siguientes descripciones.

Cuadro 5.5: Opciones de Configuración de Sistemas de Archivos.

Opción	Parámetros	Función	Defecto	Ambito
unix realname	booleano	Proporciona nombre completo de usuario al cliente.	no	Global
dont descend	string (lista de directorios)	Indica una lista de directorios cuyos contenidos Samba debería hacer invisibles a los clientes.	Ninguno	Recurso
follow symlinks	booleano	Si es no, Samba no seguirá enlaces simbólicos.	yes	Recurso
getwd cache	booleano	Si es yes, Samba usará una caché para llamadas getwd( ).	yes	Global
wide links	booleano	Si es yes, Samba seguirá enlaces simbólicos fuera del recurso.	yes	Recurso
hide dot files	booleano	Si es yes, tratará a los archivos ocultos Unix como archivos ocultos de Windows.	yes	Recurso
hide files	string (lista de archivos)	Lista de patrones de archivos para tratarlos como ocultos.	Ninguno	Recurso
veto files	string (lista de archivos)	Lista de patrones de archivos para no mostrar nunca.	Ninguno	Recurso
delete veto files	booleano	Si es yes, borrará los ficheros marcados por veto files cuando el directorio en el que residen sea eliminado.	no	Recurso

5.2.3.1. unix realname.

Algunos programas requieren un nombre de usuario completo para poder operar. Por ejemplo, un programa de correo de Windows frecuentemente necesita asociar un nombre de usuario con un verdadero nombre. Si tu fichero de contraseñas de sistema contiene los nombres verdaderos de los usuarios en el campo GCOS, la opción unix realname instruye a Samba para proporcionar esta información a los clientes. Sin esto, el nombre del usuario simplemente será su ID de login. Por ejemplo, si tu ficheros de contraseñas de Unix contiene la siguiente línea:

```
rcollins:/KaBfco47Rer5:500:500:Robert Collins:
/home/rcollins:/bin/ksh
```

Y la opción en el fichero de configuración es:

```
[global]
  unix realname = yes
```

entonces el nombre Robert Collins será facilitado a cualquier cliente que solicite el verdadero nombre del usuario rcollins. Normalmente no necesitarás meterte con esta opción.

5.2.3.2. dont descend.

La opción dont descend puede ser usada para especificar varios directorios que deberían aparecer vacíos al cliente. Advierte que el directorio mismo todavía aparecerá. Sin embargo, Samba no mostrará los contenidos del directorio al usuario cliente. No es una buena opción usar esto como una medida de seguridad (un usuario podría encontrar una forma de dar un rodeo); esto sólo es una utilidad para evitar determinadas visualizaciones en los clientes de directorios que tengan archivos delicados. Mira nuestro anterior ejemplo en esta sección.

5.2.3.3. follow symlinks.

Esta opción, controla si Samba seguirá un enlace simbólico en el sistema operativo Unix objetivo, o si debería retornar un error al cliente. Si la opción se establece a yes, el enlace será interpretado como el propio fichero.

5.2.3.4. getwd cache.

Esta opción global especifica si Samba debería usar una caché local para la llamada a sistema Unix getwd() (get current working directory). Puedes cambiar el valor a yes como sigue:

```
[global]
  getwd cache = no
```

Establecer esta opción a yes puede incrementar significativamente el tiempo que se toma para resolver el directorio de trabajo, especialmente si la opción wide links se establece a no. No deberías modificar esta opción en condiciones de configuración normales.

5.2.3.5. wide links.

Esta opción especifica si el usuario cliente puede seguir enlaces simbólicos que apunten fuera del árbol del directorio compartido. Esto incluye cualesquiera archivos o directorios en la otra parte del enlace, con tal de que los permisos de este sean apropiados para el usuario. El valor por defecto para esta opción es yes. Advierte que esta opción no será tenida en cuenta si la opción follow symlinks se establece a no. Establecer esta opción a no ralentiza al demonio smbd considerablemente.

5.2.3.6. hide files.

La opción hide files proporciona uno o más patrones de directorio o archivo para Samba. Cualquier fichero coincidente será tratado como archivo oculto para la perspectiva del cliente. Advierte que esto simplemente significa que el atributo DOS hidden es activado, lo cual puede significar o no que el usuario pueda ver ese archivo.

Cada entrada en la lista debe comenzar, terminar o estar separada de otra entrada por una barra (/), aunque sólo halla un elemento. Esto permite la aparición de espacios en la lista. Los asteriscos pueden ser usados como comodines para representar cero o más caracteres. Las interrogaciones pueden ser usadas para representar exactamente un carácter. Por ejemplo:

```
hide files = /.jav*/README.???/
```

5.2.3.7. hide dot files.

La opción hide dot files oculta cualesquiera archivos en el servidor que comiencen por un punto (.), en orden a imitar la funcionalidad de varios comandos de shell que están presentes en sistemas Unix. Al igual que con hide files, aquellos archivos que comiencen con un punto tendrán el atributo DOS 'hidden' activado, lo cual no garantiza que el cliente no podrá verlo. El valor por defecto para esta opción es yes.

5.2.3.8. veto files.

Más fuerte que el estado de oculto es el estado proporcionado por la opción de configuración veto files. Samba nunca admitirá la existencia de estos archivos. No podrás listarlos o abrirlos desde el cliente. En realidad, esto no es una gran medida de seguridad. Simplemente es un mecanismo para evitar que programas de PC puedan borrar archivos especiales.

La sintaxis de esta opción es idéntica a la de la opción hide files: cada entrada debe comenzar, terminar, o estar separadas de otras entradas por un carácter barra (/), aunque sólo aparezca un elemento. Los asteriscos pueden ser usados como comodines para representar cero o más caracteres. Las interrogaciones pueden ser usadas para representar exactamente un carácter. Por ejemplo:

```
veto files = /*config/*default?/
```

Esta opción es simplemente administrativa -no es sustituta de una buena política de permisos de ficheros-.

5.2.3.9. delete veto files.

Esta opción le dice a Samba que elimine ficheros vetados cuando un usuario intente eliminar el directorio en el que estos residen. El valor por defecto es no. Esto significa que si un usuario intenta eliminar un directorio que contenga un fichero vetado, el fichero (y el directorio) no serán eliminados. En su lugar, el directorio permanecerá y parecerá estar vacío desde la perspectiva del usuario. Si se establece a yes, el directorio y los archivos vetados serán eliminados.

5.3. Permisos de Ficheros y Atributos en MS-DOS y Unix.

DOS nunca fue diseñado para ser un sistema operativo multiusuario ni de red. Unix, por el contrario, fue diseñado con ese propósito desde el principio. Consecuentemente, existen inconsistencias y lagunas o huecos entre los dos sistemas de archivos que Samba no sólo es consciente, sino que también proporciona soluciones para ellos. Uno de los mayores obstáculos es cómo Unix y DOS manejan los permisos de archivos.

Echemos un vistazo a cómo asigna Unix los permisos. Todos los archivos Unix tienen bits de lectura, escritura y ejecución para tres clasificaciones de usuarios: propietario (owner), grupo (group), y resto de usuarios (world). Estos permisos son visibles a la izquierda de los archivos cuando se ejecuta un comando ls -al sobre un directorio Unix. Por ejemplo:

```
-rwxr--r-- 1 tom users 2014 Apr 13 14:11 access.conf
```

Windows, por el contrarioo, tiene cuatro bits principales que usa con cualquier tipo de fichero: sólo lectura (read-only), de sistema (system), oculto (hidden), y archivo (archive). Puedes ver estos bits haciendo doble click con el botón derecho del ratón sobre el fichero y seleccionando el elemento de menú 'Propiedades'. Deberías ver algo similar a lo que muestra la Figura 5.6.[1].

A continuación, la definición de cada uno de estos bits:

Read-only (Sólo Lectura) Los contenidos del archivo pueden ser leídos por un usuario pero no pueden ser sobreescritos.

System (de Sistema) Este archivo cumple un propósito muy específico, requerido por el sistema operativo.

Hidden (Oculto) Este archivo ha sido marcado para hacerlo invisible al usuario, a menos que el sistema esté configurado para que aún así sea visible.

Archive (Archivo) Este archivo ha sido modificado desde la última copia DOS que se hizo de él.

Advierte que no existe un bit específico para indicar si el archivo es o no ejecutable. Los sistemas de ficheros DOS y Windows NT identifican a los archivos ejecutables a través de las extensiones .EXE, .COM, .CMD, o .BAT.

[1] El checkbox de sistema estará probablemente inhabilitado para tu archivo. No te preocupes por ello -todavía deberías poder ver cuándo la caja está marcada y cuándo no-. Figura 5.6: Propiedades de Archivos DOS y Windows.

Figura 5.5: Ventana de error al intentar seguir enlaces simbólicos prohibidos por Samba.

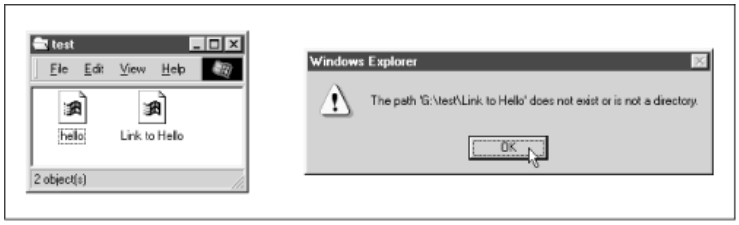
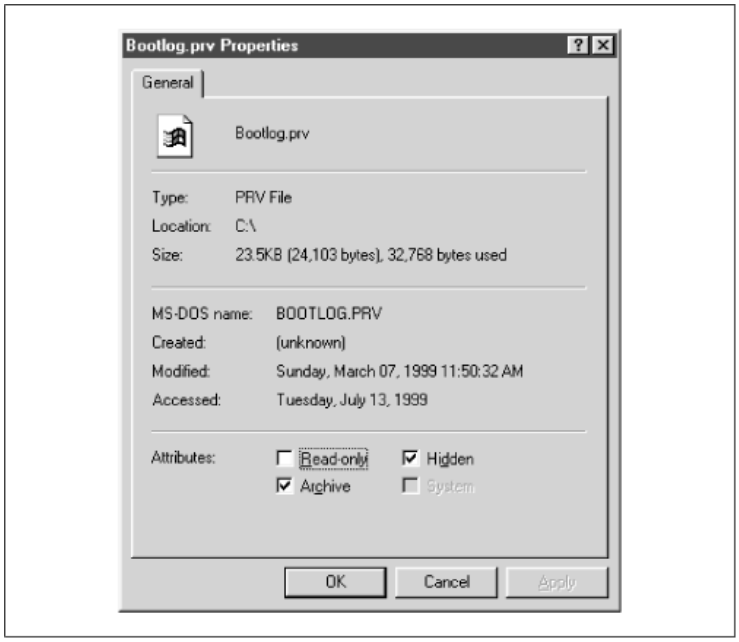


Figura 5.6: Propiedades de Archivos DOS y Windows.



Consecuentemente, no hay uso para ninguno de los tres bits de ejecutable de Unix que están presentes en un archivo que se encuentra en un recurso compartido a través de Samba. Los archivos DOS, sin embargo, tienen sus propios atributos que necesitan ser preservados cuando estos son almacenados en un entorno Unix: los bits de archivo (archive), sistema (system), y oculto (hidden). Samba puede preservar estos bits mediante la reutilización de los bits de permisos de ejecución del archivo en el lado Unix -si lo hemos configurado para que lo haga, claro-. Mapear estos bits, sin embargo, tiene un desafortunado 'efecto secundario': si un usuario Windows almacena un fichero en un recurso compartido por Samba, y tú lo ves desde el lado Unix con el comando ls -al, algunos de los bits de permisos de ejecución no significarán lo que crees que deberían significar.

Tres opciones en Samba deciden si los bits serán o no mapeados: map archive, map system , y map hidden. Estas opciones mapean los atributos archivo (archive), sistema (system), y oculto (hidden) contra los bits de permiso de ejecución del propietario (owner), grupo (group) y resto de usuarios (world) del archivo, respectivamente. Puedes añadir estas opciones al recurso [data], estableciendo cada uno de sus valores como sigue:

```
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  writeable = yes
  map archive = yes
  map system = yes
  map hidden = yes
```

Tras esto, intenta crear un archivo en el recurso desde Unix -por ejemplo, hello.java- y cambia los permisos del fichero a 755. Con estas opciones activadasd, deberías poder chequear los permisos en la parte de Windows y verías que cada uno de los tres valores aparece marcado en la caja de diálogo de Propiedades del archivo. ¿Y qué pasa con el atributo de sólo lectura? Por defecto, Samba 2.0 establece esto cuando un archivo no tiene configurado el bit de permiso de escritura para el propietario en la parte Unix. En otras palabras, puedes configurar este bit cambiando los permisos del fichero a 555.

Deberíamos advertirte que el valor por defecto para la opción map archive es yes, mientras que las otras dos opciones tienen un valor por defecto de no. Esto es así porque muchos programas no trabajarían correctamente si el bit de archivo no se almacena correctamente para archivos DOS y Windows. Los atributos de sistema (system) y oculto (hidden), sin embargo, no son críticos para la operatividad de los programas, así que quedan a la discreción del administrador del sistema.

La figura 5.7. resume los bits de permisos de Unix e ilustra cómo mapea Samba estos bits a atributos DOS. Advierte que los bits de grupo y del resto de usuarios de lectura/escritura no se traducen directamente a atributos DOS, pero ellos todavía retienen sus definiciones originales Unix en el servidor Samba.

5.3.1. Creación de Máscaras.

Samba tiene varias opciones para ayudarnos con la creación de máscaras. La creación (o eliminación) de máscaras de archivos ayudan a definir los permisos que un archivo o directorio recibirá en el momento de ser creado. En Unix, esto significa que

puedes controlar qué permisos no va a tener un archivo o directorio cuando este sea creado. Para archivos accesibles desde Windows, esto significa que puedes desactivar los atributos de sólo lectura, archivo, sistema y oculto de un archivo.

Por ejemplo, la opción create mask forzará que los permisos de un archivo creado por un cliente Windows sean, como mucho,744:

```
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  writeable = yes
  create mask = 744
```

mientras que la opción directory mask que mostramos a continuación forzará los permisos de un recién creado directorio a, como mucho, 755:

```
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  writeable = yes
  directory mask = 755
```

Alternativamente, también puedes forzar varios bits con las opciones force create mode y force directory mode. Estas opciones realizarán un 'OR lógico' contra las máscaras de creación de fichero y directorio, garantizando que estos bits especificados siempre serán establecidos. Podrías establecer estas opciones globalmente para asegurarte de que los permisos de grupo y resto de usuarios de lectura/escritura han sido establecidos apropiadamente para los nuevos archivos o directorios en cada recurso.

Siguiendo la misma filosofía, si quisieras explícitamente establecer los atributos de usuario y grupo de un fichero de Unix que se ha creado desde la parte Windows, puedes usar las opciones force user y force group. Por ejemplo:

```
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  writeable = yes
  create mask = 744
  directory mask = 755
  force user = joe
  force group = accounting
```

Estas opciones actualmente asignan un usuario y grupo Unix estáticos a cada conexión que se realizae a un recurso compartido. Sin embargo, esto ocurre después de que el cliente haya realizado la autentificación; esto no permite acceso libre a un recurso. Estas opciones son frecuentemente usadas por sus capacidades de asignar un determinado usuario y gruop a cada nuevo fichero o directorio que es creado en un recurso. Usa estas opciones con discreción.

Finalmente, una de las capacidades de Unix que DOS no tiene es la habilidad de eliminar un fichero de sólo lectura desde un directorio con permisos de escritura. En

Unix, si un directorio es escribible, un fichero de sólo lectura en ese directorio aún puede ser eliminado. Esto te permitiría eliminar ficheros an cualquiera de tus directoriosThis could permit you to delete files in any of your directories, aún si el archivo fue depositado por algún otro usuario.

Los sistemas de archivos DOS no están diseñados para múltiples usuarios, y por eso sus diseñadores decidieron que 'sólo lectura' significa 'proteger contra cambios accidentales, incluyendo la eliminación', más que 'proteger contra algún otro usuario en una máquina de un usuario'. Así que los diseñadores del DOS prohibieron la eliminación de un fichero de sólo lectura. Aún hoy día, los sistemas de archivos Windows exhiben la esta característica.

Normalmente, esto no es un problema. Los programas Windows no intentarán eliminar ficheros de sólo lectura porque ellos saben que esa es una mala idea. Sin embargo, un número de programas de control de código fuente -que primero fueron escritos para Unix- funcionan en Windows y requieren la habilidad de eliminar ficheros de sólo lectura. Samba permite esta característica con la opción delete readonly. Para activar esta funcionalidad, establece la opción a yes:

```
[data]
  path = /home/samba/data
  browseable = yes
  guest ok = yes
  writeable = yes
  create mask = 744
  directory mask = 755
  force user = joe
  force group = accounting
  delete readonly = yes
```

5.3.2. Opciones de Permisos de Ficheros y Directorios.

Las opciones de permisos de ficheros y directorios están resumidas en la Tabla 5.5; cada opción se decribe en detalle.

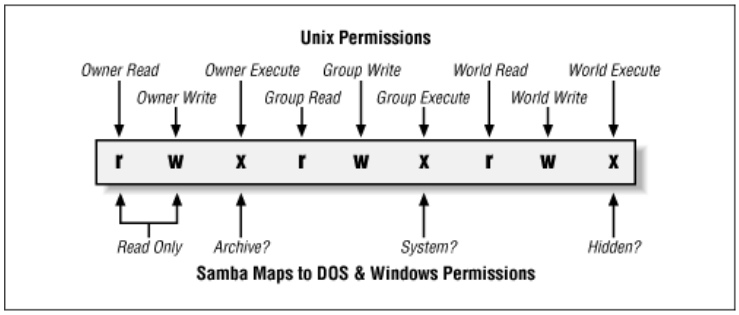
5.3.2.1. create mask.

El argumento para ésta opción es un número octal indicando qué nivel de permisos pueden ser establecidos por un cliente en la creación de archivos en un recurso. El valor por defecto es 0755, lo que significa que el propietario Unix puede leer, escribir y opcionalmente ejecutar esos ficheros, mientras que los miembros del grupo al que pertenece y los otros sólo podrán leer o ejecutar los archivos. Si necesitas cambiarlo a archivos no ejecutables, recomendamos 0644, o rw-r–r–. Recuerda que los bits de permisos de ejecución pueden ser usados por el servidor para mapear determinados atributos de archivos del DOS, como ya vimos antes. Si estás cambiando la máscara de creación, estos bits tienen que ser parte de la máscara también.

5.3.2.2. directory mask.

El argumento para esta opción es un valor octal, indicando qué permisos pueden asignarse a la creación de directorios por parte de un usuario en un recurso. El valor por defecto es 0755, lo cual a cualquiera de la parte Unix leer y recorrer los directorios,

Figura 5.7: Cómo Samba y Unix ven los permisos de un archivo.



Cuadro 5.6: Opciones de Permisos de Ficheros y Directorios.

Opción	Parámetros	Función	Defecto	Ambito
map archive	booleano	Preserva el atributo DOS de "archivo" en bit de permiso de ejecución de usuario (0100).	yes	Recurso
map system	booleano	Preserva el atributo DOS de "sistema" en bit de permiso de ejecución de grupo (0010).	no	Recurso
map hidden	booleano	Preserva el atributo DOS de "oculto" en bit de permiso de ejecución de resto de usuarios(0001).	no	Recurso
create mask (create mode)	numerico	Establece el máximo nivel de permisos para los archivos creados por Samba.	0744	Recurso
directory mask (directory mode)	numerico	Establece el máximo nivel de permisos para los directorio creados por Samba.	0755	Recurso
force create mode	numerico	Fuerza permisos específicos para archivos creados por Samba.	0000	Recurso
force directory mode	numerico	Fuerza permisos específicos para directorios creados por Samba.	0000	Recurso
force group (group)	string (nombre grupo)	Establece el grupo efectivo para un usuario accediendo a ese recurso.	Ninguno	Recurso
force user	string (nombre usuario)	Establece el nombre de usuario efectivo para el usuario que accede al recurso.	Ninguno	Recurso
delete readonly	booleano	Permite a un usuario eliminar un fichero de sólo lectura desde un directorio con permisos de escritura.	no	Recurso

pero sólo te permite modificarlos a ti (al propietario). Recomendamos la máscara 0750, eliminando la posibilidad de acceso al resto de usuarios (que no son tú, y que tampoco pertenecen a tu grupo).

5.3.2.3. force create mode.

Esta opción establece los bits de permisos que Samba forzará a ser establecidos cuando se realice un cambio de permisos en un archivo. Esto se usa frecuentemente para forzar permisos de grupo, ya mencionados antes. También pueden ser usados para preestablecer cualquiera de los atributos DOS que ya comentamos: archivo (0100), sistema (0010), u oculto (0001). Esta opción siempre toma efecto tras las opciones map archive, map system , map hidden, y create mask.

Muchas aplicaciones Windows renombran sus archivos de datos a fichero_de_datos.bak y crean otros nuevos, cambiando al propietario del mismo y sus permisos para que los miembros del mismo grupo Unix no puedan editarlos. Establecer *force create mask=0660* mantendrá el nuevo fichero editable para los miembros del grupo.

5.3.2.4. force directory mode.

Esta opción establece los bits de permisos que Samba forzará cuando un sea realizado un cambio de permisos en un directorio, o cuando un directorio sea creado. Esto es usado frecuentemente para forzar permisos de grupo, como mencionamos antes. Esta opción se pone por defecto a 0000, y puede ser usado igual que con force create mode para añadir permisos de grupo o de otro tipo, si se necesitan. Esta opción siempre toma efecto tras las opciones map archive, map system, map hidden, y directory mask.

5.3.2.5. force group.

Esta opción, a veces llamada group, asigna un ID estático de grupo que será usado en todas las conexiones para un servicio, una vez que el cliente se haya autentificado. Esto asigna un grupo específico a cada nuevo archivo o directorio creado desde un cliente SMB.

5.3.2.6. force user.

La opción force user asigna un ID estático de usuario que será usado en todas las conexiones a un servicio, una vez que el cliente se haya autentificado. Esto asigna un usuario específico a cada nuevo archivo o directorio creado desde un cliente SMB.

5.3.2.7. delete readonly.

Esta opción permite a un usuario eliminar un directorio conteniendo un archivo de sólo lectura. Por defecto , DOS y Windows no permitirán esta operación. Probablemente querrás dejar esta opción desactivada, a menos que alguno de tus programas necesite esta capacidad; muchos usuarios Windows podrían encontrarse con la desagradable sorpresa de que ellos han eliminado accidentalmente un archivo de sólo lectura. De hecho, incluso el comando Unix rm preguntará a los usuarios si realmente desean eliminar un archivo de sólo lectura. Es buena idea dejar a Samba que sea cauto.

5.3.2.8. map archive.

El bit de archivo de DOS es usado para marcar a un fichero que ha sido cambiado desde la última vez que se archivó (p.ej., salvado con el programa DOS ms-backup). Establecer la opción de Samba map archive = yes provoca que el flag de archivo de DOS sea mapeado al bit de 'ejecutable por el propietario' (0100). Es mejor dejar esta opción activada si tus usuarios Windows están haciendo sus propias copias de seguridad, o bien están usando programas que requieren el bit de archivo. Unix ignora la noción de un bit de archivo totalmente. Los programas de backup normalmente mantienen un fichero que lista todos los archivos que han sido salvados y en qué fecha, y así el comparar las fechas de modificación de archivos sirve al mismo propósito.

Establecer esta opción a yes provoca una sorpresa ocasional en Unix cuando un usuario vea que un fichero de datos está marcado como ejecutable, pero raramente causará daño. Si un usuario intentase ejecutarlo, normalmente recibiría una serie de mensajes de error a medida que el shell intentase ejecutar las primeras líneas del archivo como si fueran comandos. El caso contrario también es posible; un programa ejecutable Unix aparecería como que no ha sido salvado recientemente en Windows. Pero de nuevo, esto sería raro, y normalmente inofensivo.

5.3.2.9. map system.

El atributo DOS 'de sistema' es usado para indicar que son ficheros requeridos por el sistema operativo, y no deberían ser eliminados, renombrados o movidos de su ubicación original. Establece esta opción sólo si necesitas almacenar archivos de sistema de Windows en el servidor Unix. Los programas ejecutables de Unix parecerán ser ficheros especiales y no-removibles de Windows cuando sean vistos por los clientes Windows. Esto puede proporcionarte algún inconveniente si quieres mover o eliminar alguno de ellos. Para la mayoría de los sitios, sin embargo, esto es inofensivo.

5.3.2.10. map hidden.

DOS usa el atributo 'oculto' para indicar que un archivo no debería ser normalmente visible en los listados de archivos. Unix no tiene este atributo; Normalmente, no tendrás ficheros DOS que necesiten ser ocultados, así que lo mejor que puedes hacer es dejar esta opción desactivada.

Establecer ésta opción a yes causaría que el servidor mapearía el atributo 'oculto' en el bit de permiso de ejecución de 'el resto de usuarios' (0001). Esta característica puede producir un efecto sorprendente. Cualquier programa Unix que sea ejecutable por todos los usuarios parecería desaparecer cuando lo buscases desde un cliente Windows. Si esta opción no se establece, sin emabrgo, y un usuario Windows intentase marcar un fichero oculto en un recurso Samba, no podría hacerlo, porque ¡¡Samba no tiene sitio para almacenar el atributo 'oculto'!!.

5.4. Planchado de Nombres (Name Mangling) y Tipo.

En los primeros días de DOS y Windows 3.1, los nombres de archivos estaban limitados a 8 caracteres mayúsculas/minúsculas, seguidos por un punto, y 3 caracateres más en mayúsculas. Esto se conocía como el formato 8.3, y era una molestia. Windows 95/98, Windows NT, y Unix han mejorado esta situación permitiendo que muchos caracteres (sensibles a may/minúsculas) conformen un nombre de archivo. La Tabla 5.6

muestra el actual estado del sistema de nombrado de archivos de los sistemas operativos más populares.

Cuadro 5.8: Limitaciones de Nombrado de Archivos en diferentes Sistemas Operativos.

S.O.	**Reglas de Nombrado**
DOS 6.22 o inferior	8 caracteres seguidos por un punto y una extensión de tres letras (formato 8.3). Insensibles a mayúsculas/minúsculas.
Windows 3.1 para Trabajo en Grupos	8 caracteres seguidos por un punto y una extensión de tres letras (formato 8.3). Insensibles a mayúsculas/minúsculas.
Windows 95/98	127 caracteres; sensibles a mayúsculas/minúsculas.
Windows NT	127 caracteres; sensibles a mayúsculas/minúsculas.
Unix	255 caracteres; sensibles a mayúsculas/minúsculas.

Samba debe mantener la compatibilidad con clientes de red que almacenan archivos en el formato 8.3, tales como Windows para Trabajo en Grupos. Si un usuario crea un fichero en un recurso llamado antidisestablishmentarianism.txt, un cliente WFW no podría diferenciarlo de otro archivo en el mismo directorio llamado antidisease.txt. Al igual que Windows 95/98 y Windows NT, Samba tiene que emplear una metodología especial de traslación de nombres largos de archivo a un formato 8.3, de manera que nombres similares de archivos no creen conflictos. Esto es denominado name mangling o “planchado de nombre”, y Samba lo hace de una manera similar, pero no idéntica, a como lo hacen Windows 95 y sus sucesores.

5.4.1. La Operación de “Planchado” de Samba.

Así es como Samba resume un nombre largo de archivo y lo convierte a un formato de archivo 8.3:

- Si el nombre de archivo original no comienzan por un punto, hasta los 5 primeros caracteres alfanumérico que aparezcan hasta el siguiente punto (si lo hay) son convertidos a mayúsculas. Estos caracteres son usados como los 5 primeros caracteres del nombre “planchado” 8.3.
- Si el nombre de archivo original comienza por un punto, el punto es eliminado y hasta los 5 primeros caracteres alfanuméricos que aparezcan antes del próximo punto (si lo hay) son convertidos a mayúsculas. Estos caracteres son usados como los 5 primeros caracteres del nombre pasado a formato 8.3.
- Estos caracteres son inmediatamente seguidos por un símbolo especial de “planchado”: por defecto, una vírgula (~), aunque Samba te permite cambiar este carácter.
- La base del nombre de fichero largo antes del último punto es convertido en un código de dos caracteres; partes del nomrbe tras el último punto pueden ser usadas, si es necesario. Este código de dos caracteres es añadido al nombre del fichero 8.3 tras el planchado de caracteres.
- Los primeros 3 caracteres tras el último punto (si lo hay) del nombre del fichero original son convertidos a mayúsculas y añadidos al nombre planchado o resumido como su extensión. Si el nombre original del fichero comenzase por un punto, tres caracteres de subrayado (_ _ _) son usados como extensión en su lugar.

Aquí tienes algunos ejemplos:

```
virtuosity.dat                  VIRTU~F1.DAT
.htaccess                       HTACC~U0.
hello.java                      HELLO~1F.JAV
team.config.txt                 TEAMC~04.TXT
antidisestablishmentarianism.txt ANTID~E3.TXT
antidiseast.txt                 ANTID~9K.TXT
```

El uso de estas reglas permitirá a WFW diferenciar los dos ficheros desde la parte 'pobre' del cliente que está obligado a ver la red a través de los ojos de ese sistema operativo. Advierte que el mismo nombre largo de fichero debería siempre coincidir con la misma aproximación que genera Samba; esto no siempre ocurre así con Windows. La cara oculta de ésta aproximación es que todavía pueden existir colisiones; sin embargo, las oportunidades están muy reducidas.

Tú generalmente querrás usar las opciones de configuración de planchado de nombres sólo con los clientes más antiguos. Te recomendamos hacerlo así para no 'romper' a los demás clientes añadiendo una directiva include al fichero smb.conf:

```
[global]
include = /ucsr/local/samba/lib/smb.conf.%m
```

Esto se traduce a smb.conf.WfWg cuando un cliente Windows para Trabajo en Grupo (WFW) se conecta. Ahora puedes crear un fichero /usr/local/samba/lib/smb.conf.WfWg, que podría contener las siguientes opciones:

```
[global]
  case sensitive = no
  default case = upper
  preserve case = no
  short preserve case = no
  mangle case = yes
  mangled names= yes
```

Si no estás usando Windows para Trabajo en Grupos 3.1, entonces probablemente no necesitarás cambiar ninguna de estas opciones de sus valores por defecto.

5.4.1.1. Representando y Resolviendo Nombres de Archivo con Samba.

Otra cosa que debemos apuntar es que existe una diferencia entre cómo un sistema operativo representa a un fichero y cómo lo resuelve. Por ejemplo, si estás usando Windows 95/98/NT, probablemente te toparás con un fichero denominado README.TXT. El archivo puede ser representado por el sistema operativo enteramente con caracteres mayúsculas. Sin embargo, si abres una ventana de MS-DOS e introduces el comando edit readme.txt, el archivo en mayúsculas es cargado en el programa de edición, ¡Aunque hayas tecleado el nombre an caracteres minúsculas!

Esto es así porque la familia de sistemas operativos Windows 95/98/NT resuelven los ficheros de una forma insensible a las mayúsculas/minúsculas, aunque los ficheros estén representados de forma sensible a mayúsculas/minúsculas. Los sistemas operativos basados en Unix, por otra parte, siempre resuelven los archivos de forma sensible a mayúsculas/minúsculas; si intentas editar el archivo README.TXT con el comando vi readme.txt, obtendrás la edición de un buffer vacío para un nuevo archivo.

Así es como Samba maneja las mayúsculas/minúsculas: si la opción preserve case se establece a yes, Samba siempre usará las mayúsculas proporcionadas por el sistema operativo para representar (que no resolver) los nombres de archivos. Si se establece a no, este usará el tipo de caracteres especificados por la opción default case. Lo mismo es cierto para la opción short preserve case. Si esta opción se establece a yes, Samba usará el tipo por defecto del sistema operativo para representar nombres de archivo 8.3; de lo contrario, usará el tipo especificado por la opción default case. Finalmente, Samba siempre resolverá nomrbes de archivo en sus recursos basándose en el valor de la opción case sensitive.

5.4.2. Opciones de Planchado.

Samba te permite darle instrucciones más refinadas sobre cómo debe realizar la conversión de nombres, incluyendo el control del tipo de sensibilidad ante caracteres (mayúsculas/minúsculas), el carácter insertado para formar un nombre resumido, y la capacidad de mapear manualmente nombres de archivos de un formato a otro. Estas opciones se muestran en la Tabla 5.7.

Cuadro 5.9: Opciones de Planchado de Nombres.

Opción	Parámetros	Función	Defecto	Ambito
case sensitive (casesignames)	booleano	Si es yes, Samba tratará los nombres como sensibles a mayúsculas/minúsculas (Windows no lo hace).	no	Recurso
default case	(upper o lower)	Tipo a asumir como por defecto (sólo usado cuando preserve case es no).	Lower	Recurso
preserve case	booleano	Si es yes, mantiene el tipo suministrado por el cliente (p.ej., no convierte a default case).	yes	Recurso
short preserve case	booleano	Si es yes, preserva el tipo de los nombres de formato 8.3 que proporciona el cliente.	yes	Recurso
mangle case	booleano	Resume un nombre si éste tiene caracteres de un tipo y de otro.	no	Recurso
mangled names	booleano	Resume nombres largos a formato DOS 8.3.	yes	Recurso
mangling char	string (carácter simple)	Ofrece planchado de nombres.	~	Recurso
mangled stack	numerico	Número de nombres resumidos a mantener en la pila de nombres resumidos local.	50	Global
mangled map	string (lista de patrones)	Permite mapeado de nombres de archivos de un formato a otro.	Ninguno	Recurso

5.4.2.1. case sensitive.

Esta opción de nivel de recurso, la cual tiene el obtuso sinónimo de casesignames, especifica si Samba debería preservar el tipo de caracteres cuando resuelva nombres de archivos en un recurso determinado. El valor por defecto para esta opción es no, el cual es como Windows maneja la resolución de archivos. Si los clientes están usando un sistema operativo que toma ventaja de de los nombres de archivos sensibles a mayúsculas, puedes establecer esta opción de configuración a yes como ves a continuación:

```
[accounting]
  case sensitive = yes
```

Por otra parte, recomendaos que dejes esta opción a su valor por defecto.

```
default case.
```

La opción default case es usada con preserve case. Especifica el tipo por defecto (mayúscula o minúscula) que Samba usará cuando cree un archivo en uno de sus recursos en nombre de un cliente. El valor por defecto es lower, lo que significa que los archivos recién creados usarán nombres mixtos para ellos. Si lo necesitas, puedes sobreescribir esta opción global especificando lo siguiente:

```
[global]
  default case = upper
```

Si especificas este valor, los nombres de los ficheros creados más recientemente serán trasladados a mayúsculas, y no podrán 'atropellarse' en un programa. Recomendamos que uses el valor por defecto a menos que tengas en tu red un cliente Windows for Workgroups u otro cliente de formato 8.3, en cuyo caso debería ser upper.

5.4.2.2. preserve case.

Esta opción especifica si un cliente creado por Samba en la parte del cliente es creado con el tipo de carácter proporcionado por el sistema operativo cliente, o por el tipo especificado por la opción de configuración default case anterior. El valor por defecto es yes, el cual usa el tipo proporcionado por el sistema operativo cliente. Si se establece a no, se usará el valor de la opción default case.

Advierte que esta opción no maneja las peticiones de ficheros 8.3 solicitadas por el cliente -mira la opción short preserve case-. Puedes querer establecer esta opción a yessi las aplicaciones que crean ficheros en el servidor Samba son sensiblesal tipo usado cuando se creó el archivo. Si quieres forzar a Samba, por ejemplo, para imitar la conducta de un sistema de archivos de un Windows NT, puedes dejar esta opción a su valor por defecto, yes.

5.4.2.3. short preserve case.

Esta opción especifica si un nombre de archivo 8.3 creado por Samba en la parte del cliente es creado con el tipo por defecto del sistema operativo cliente, o el tipo especificado por la opción de configuración default case. El valor por defecto es yes, el cual usa el tipo proporcionado por el sistema operativo del cliente. Puedes dejar a Samba escoger el tipo de carácter a través de la opción default case configurándola como sigue:

```
[global]
  short preserve case = no
```

Si quieres forzar a Samba a imitar el funcionamiento de un sistema de archivos de un Windows NT, puedes dejar esta opción configurada a su valor por defecto, yes.

5.4.2.4. mangled names.

Esta opción de nivel de recurso especifica si Samba resumirá archivos para clientes 8. en ese recurso. Si la opción se establece a no, Samba no resumirá los nombres y (dependiendo del cliente), estos serán invisibles o aparecerán truncados para aquellos que usen sistemas operativos 8.3. El valor por defecto es yes. Puedes cambiar esto así:

```
[data]
  mangled names = no
```

5.4.2.5. mangle case.

Esta opción le dice a Samba si debería resumis nombres de archivos que no están compuesto completamente por caracteres del tipo especificado usando la opción default case. El valor por defecto es no. Si lo estableces a yes, deberías asegurarte de que todos los lcientes serán capaces de manejar los nombres resumidos que resulten. Puedes cambiar esto como sigue:

```
[data]
  mangle case = yes
```

Recomendamos que ignores esta opción, a menos que tengas una necesidad justificada de cambiar esto.

5.4.2.6. mangling char.

Esta opción de nivel de recurso especifica el carácter de resumen a usar cuando Samba resuma nombres de archivos al formato 8.3. El valor por defecto es la vírgula (~). Puedes cambiarlo por el que quieras, como por ejemplo:

```
[data]
  mangling char = #
```

5.4.2.7. mangled stack.

Samba mantiene una pila local de nombres de archivos recientemente resumidos al formato 8.3; esta pila puede ser usada para mapeado inverso de nombres resumidos, para devolverlos a su estado original. Esto es frecuentemente necesitado por aplicaciones que crean y almacena archivos, los cierran, y necesitan modificarlos más tarde. El número por defecto de pares 'nombre largo'/'nombre resumido' almacenados en esta pila es 50. Sim embargo, si quieres acortar la cantidad de tiempo de proceso usado para resumir nombres de archivos, puedes incrementar el tamaño de la pila al que quieras, a expensas de la memoria y de un acceso al archivo ligeramente más lento.

```
[global]
  mangled stack = 100
```

5.4.2.8. mangled map.

Si la conducta por defecto de planchado de nombre no es suficiente, puedes darle a Samba más completas instrucciones sobre cómo comportarse usando la opción mangled map. Esta opción te permite especificar patrones de mapeado que pueden ser usados antes o en lugar del tipo de resumen realizado por Samba. Por ejemplo:

```
[data]
  mangled map =(*.database *.db) (*.class *.cls)
```

Aquí, Samba es instruido para buscar cada fichero que encuentre con caracteres que coincidan con el primer patrón especificado en los paréntesis y los convierte al segundo patrón especificado en lo otros paréntesis para displayarlo en un cliente 8.3. Esto es útil en el caso de que el planchado de nombres convierta el nombre del archivo incorrectamente, o a un formato que el cliente no pueda comprender. Los patrones están separados por espacios en blanco.

5.5. Bloqueos y Opciones de Bloqueos.

Los intentos de escritura concurrentes a un mismo archivo no son nada deseables en la mayoría de sistemas operativos. Para prevenir esto, la mayoría de sistemas operativos usan bloqueos para garantizar que sólo un proceso puede escribir sobre un fichero en un determinado instante de tiempo. Tradicionalmente, los sistemas operativos bloquean ficheros completos, aunque algunos permiten bloquear un rango de bytes dentro de un fichero. Si otro proceso intenta escribir sobre un fichero (o sección de fichero) que todavía está bloqueado, este recibirá un error desde el sistema operativo y tendrá que esperar hasta que el bloqueo sea liberado.

Samba soporta las peticiones de bloqueo standard de los sistemas de archivos DOS y NT (deny-mode), los cuales permiten que un sólo proceso pueda escribir sobre un archivo completo en un servidor en un momento determinado de tiempo, así como permiten el bloqueo de rango de bytes. En adición, Samba soporta un nuevo mecanismo de bloqueo conocido en el mundo de Windows NT como bloqueo oportunista (oplock para resumir).

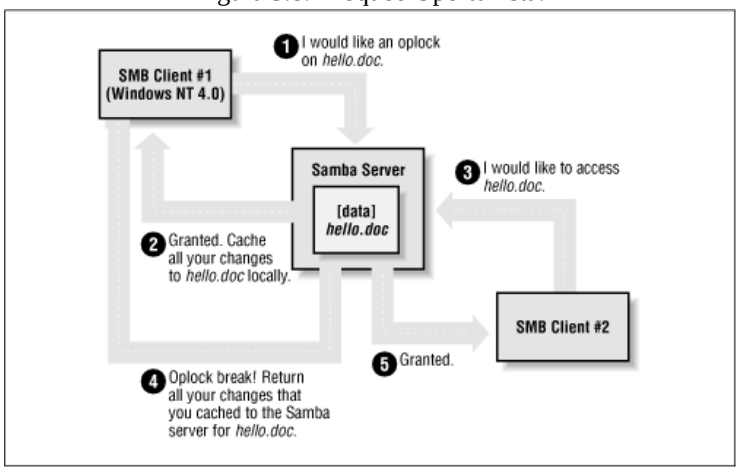
5.5.1. Bloqueo Oportunista.

El bloqueo oportunista permite a un cliente notificar al servidor Samba que éste no será el que pueda escribir en exclusiva sobre un fichero, pero también podrá almacenar (en caché) los cambios que haya realizado sobre el fichero en su propia máquina (y no en el servidor Samba) en orden a acelerar el acceso al fichero para ese cliente. Cuando Samba que ese fichero ha sido bloqueado oportunísticamente por un cliente, este lo marca y espera a que el cliente complete su trabajo sobre el fichero, al punto que espera a que el cliente envíe los cambios finales de vuelta al servidor Samba para sincronización.

Si una segunda petición de cliente accede a ese fichero antes de que el primero haya terminado de trabajar con él, Samba puede enviar una petición oplock break (romper bloqueo oportunista) al primer cliente. Este le dice al cliente que pare el almacenamiento en caché de sus cambios y retorne al estado actual del fichero desde el servidor, para que el cliente que interrumpe pueda usarlo sin peligro a cambios posteriores por la parte del otro cliente. Un bloqueo oportunista, sin embargo, no elimina un bloqueo

estandar del tipo modo-denegación. Es posible que el proceso que interrumpe consiga una rotura del bloqueo oportunista sólo para encontrarse con que el proceso original también tenía un bloqueo modo-denegación sobre el fichero. La Figura 5.8 ilustra el proceso del bloqueo oportunista.

Figura 5.8: Bloqueo Oportunista.



En términos de bloqueos, recomendamos efusivamente el uso de los valores por defecto proporcionados por Samba: bloqueos standard del tipo modo-denegación DOS/Windows para compatibilidad y bloqueos oportunistas para el rendimiento extra que permita la caché local. Si tu sistema operativo puede tomar ventaja de los bloqueos oportunistas, esto te podría proporcionar un aumento del rendimiento. A menos que tengas una razón específica para cambiar cualquiera de estas opciones, es mejor que las dejes como están.

5.5.2. Unix y los Bloqueos.

Los sistemas Windows cooperan bien para impedir la pérdida de los cambios realizados. Pero si un fichero almacenado en un sistema Samba es accedido por un proceso Unix, este proceso no entiende nada de bloqueos de Windows, y podría fácilmente saltarse ese bloqueo. Algunos sistemas Unix han sido diseñados para comprender los bloqueos de Windows mantendos por Samba. Actualmente existe el soporte sólo en SGI Irix 6.5.2f y posteriores; Linux y FreeBSD deberían implementarlo pronto.

Si tienes un sistema que entiende estos bloqueos, establece kernel oplocks = yes en el fichero de configuración de Samba. Esto debería eliminar los conflictos entre los procesos Unix y los usuarios Windows.

Si tu sistema no soporta los bloqueos oportunistas a nivel de núcleo, podrías terminar con datos corruptos cuando alguien ejecute un proceso Unix que lea o escriba un fichero al que los usuarios Windows también hayan accedido. Sin embargo, Samba proporciona un mecanismo de protección en ausencia de bloqueos oportunistas a nivel de núcleo: la opción veto oplock files. Si puedes anticipar qué ficheros Samba serán usados tanto por usuarios Windows y Unix, establece sus nombres en una opción veto oplock files. Esto suprimirá el uso de bloqueos oportunistas sobre los ficheros coincidentes, lo cual suprime el cacheado del cliente, y permite a los programas Windows y

Unix usar bloqueo de sistema o tiempos de actualización para detectar una pugna por el mismo archivo. Un ejemplo:

```
veto oplock files = /*.dbm/}
```

Esta opción permite tanto a los procesos Unix como a los usuarios Windows editar archivos que terminen con el sufijo .dbm. Advierte que la sintaxis de esta opción es similar a veto files.

Las opciones de Samba para bloqueos y bloqueos oportunistas los tienes en la Tabla 5.8.

Cuadro 5.11: Opciones de Bloqueos y Bloqueos Oportunistas.

Opción	Parámetros	Función	Defecto	Ambito
share modes	booleano	Si es yes, activa el soporte para bloqueos globales de fichero tipo DOS.	yes	Recurso
locking	booleano	Si es yes, activa bloqueos de rango de bytes.	yes	Recurso
strict locking	booleano	Si es yes, deniega el acceso total a un fichero completo si existe un bloqueo de rango de bytes.	no	Recurso
oplocks	booleano	Si es yes, activa una caché local de los ficheros en el cliente para este recurso.	yes	Recurso
kernel oplocks	booleano	Si es yes, indica que el kernel soporta bloqueos oportunistas.	yes	Global
fake oplocks	booleano	Si es yes, le dice al cliente que se obtuvo el bloqueo, pero que actualmente no está bloqueado.	no	Recurso
blocking locks	booleano	Permite al peticionario del bloqueo esperar a obtener el bloqueo.	yes	Recurso
veto oplock files	string (lista de nombres de archivos)	No hace bloqueos oportunistas a los ficheros especificados.	Ninguno	Recurso
lock directory	string (Ruta completamente cualificada)	Establece la localización donde varios archivos Samba, incluyendo bloqueos, son almacenados.	Como se especificase en el makefile de Samba	Global

5.5.2.1. share modes.

Los tipos de bloqueo más antiguos disponibles para Samba eran los de modo-denegación, conocidos como share modes, los cuales eran empleados por programas como editores de texto para evitar escrituras accidentales sobre los archivos abiertos. Los bloqueos de modo-denegación se listan en la Tabla 5.9.

El parámetro share modes, el cual refuerza el uso de estos bloqueos, está activado por defecto. Para desactivarlo, usa el siguiente comando:

[accounting] share modes = no

Te recomendamos efusivamente no desactives el mecanismo de bloqueo por defecto a menos que tengas una razón justificada para hacerlo. La mayoría de las aplicaciones Windows y DOS cuentan con estos mecanismos de bloqueo en orden a funcionar correctamente, y echarán de menos esta funcionalidad si la desactivas.

5.5.2.2. locking.

La opción locking puede ser usado para decirle a Samba que establezca o no bloqueos de rango de bytes en la parte del cliente. Samba implementa bloqueos de rango de bytes en el servidor con bloqueos de aviso típicos de Unix y consecuentemente prevendrán a otros procesos Unix contra la escritura de un rango de bytes bloqueado.

Esta opción puede ser especificada por cada recurso como sigue:

```
[accounting]
  locking = yes
```

Si la opción locking se establece a yes, el peticionario será puesto en espera hasta que el tenedor actual u otro tipo de bloqueo lo libere (o caiga). Si, por el contrario, la opción se establece a no, no se mantendrán ningún tipo de bloqueos de rango de bytes sobre los ficheros, aunque se intenten peticiones de bloqueo y desbloqueo. La opción se establece a yes por defecto; sin embargo, puedes desactivar esta opción si se trata de un medio de sólo lectura.

5.5.2.3. strict locking.

Esta opción chequea cada acceso a fichero buscando bloqueos de rango de bytes en el rango de bytes al que se está accediendo. Esto normalmente no es necesario si un cliente se adhiere a todos los mecanismos de bloqueo. Esta opción se establece a no por defecto; sin embargo, puedes resetear esto por cada recurso como sigue:

```
[accounting]
  strict locking = yes
```

Si esta opción se establece a yes, los bloqueos mandatorios son reforzados sobre cualquier archivo con bloqueos de rango de bytes.

5.5.2.4. blocking locks.

Samba también soporta bloqueos de bloque, una variante menor de bloqueos de rango. Aquí, si el rango de bytes no está disponible, el cliente especifica una cantidad de tiempo que está dispuesto a esperar. El servidor entonces cachea la petición de bloqueo, chequeando periódicamente para ver si el fichero está disponible. Si lo está, lo notifica al cliente; sin embargo, si el tiempo expira, Samba le dirá al cliente que la petición no pudo ser atendida. Esta estrategia previene contra que el cliente esté contínuamente conectando para ver si el bloqueo está disponible.

Puedes desactivar esta opción por cada recurso como sigue:

```
[accounting]
  blocking locks = no
```

Cuando se establece a yes, los bloqueos de bloques se reforzarán sobre el archivo. Si esta opción se establece a no, Samba se comporta como si existieran unos mecanismos de bloqueo normales sobre el fichero. El valor por defecto es yes.

5.5.2.5. oplocks.

Esta opción activa o desactiva el soporte para bloqueos oportunistas en el cliente. La opción es activada por defecto. Sin embargo, puedes desactivarla con el siguiente comando:

```
[data]
  oplocks = no
```

Si estás en un entorno de red altamente inestable o tienes muchos clientes que no puedan tomar ventaja de los bloqueos oportunistas, puede que sea mejor desactivar esta característica en Samba. Lo bloqueos oportunistas deberían ser desactivados si estás accediendo a los mismos ficheros tanto desde aplicaciones Unix (tales como vi) y clientes SMB (a menos que tengas la suerte de contar con un s.o. que soporte bloqueos oportunistas a nivel de núcleo, como ya comentamos antes).

5.5.2.6. fake oplocks.

Antes de que los bloqueos oportunistas estuvieran disponibles en Samba, los demonios Samba pretendian permitir bloqueos oportunistas a través de la opción fake oplocks. Si esta opción fue activada, todos los clientes fueron avisados de que el fichero está disponible para hacerle bloqueos oportunistas, y nunca se avisará sobre accesos simultáneos. Esta opción ha caido en deshuso ahora que los verdaderos bloqueos oportunistas están disponibles en Samba.

5.5.2.7. kernel oplocks.

Si una aplicación Unix separada de Samba intenta actualizar un fichero que Samba tiene con bloqueo oportunista para un cliente Windows, este tendrá éxito (dependiendo del s.o.) y tanto Samba como el cliente nunca lo advertirán. Sin embargo, si el s.o. Unix local lo soporta, Samba puede advertir sobre los ficheros con bloqueos oportunistas, los cuales pueden suspender el proceso Unix, notificar al cliente vía Samba que guarde su copia del archivo, y sólo entonces permitir la apertura del ficihero. Esencialmente, esto significa que el kernel del sistema operativo del sistema Samba tiene la habilidad de manejar bloqueos oportunistas tan bien como Samba.

Puedes activar esta característica con la opción kernel oplocks, como sigue:

```
[global]
  kernel oplocks = yes
```

Samba puede detectar automáticamente bloqueos oportunistas a nivel de kernel y usarlos, si están presentes. Al tiempo de la escritura de este capítulo, esta característica está soportada sólo por SGI Irix 6.5.2f y posteriores. Sin embargo, el soporte para Linux y FreeBSD se espera esté en un futuro próximo. Un sistema sin bloqueos oportunistas a nivel del kernel permitirán a los procesos Unix actualizar el fichero, pero los programas cliente serán notarán el cambio sólo más tarde, en todo caso.

5.5.2.8. veto oplock files.

Puedes proporcionar una lista de nombres de archivos sobre los que nunca se permitirán bloqueos oportunistas con la opción veto oplock files. Esta opción puede ser establecida globalmente o a nivel de recurso. Por ejemplo:

```
veto oplock files = /*.bat/*.htm/
```

El valor para esta opción es una serie de patrones. Cada entrada del patrón debe comenzar, termianr, o estar separada de otra por una barra (/), aunque sólo haya uno en la lista. Los asteriscos se pueden usar como comodín para representar cero o más caracteres. Las interrogaciones pueden ser usados para representar exactamente un carácter.

Recomendamos que desactives los bloqueos oportunistas sobre cualesquiera ficheros que vayan a ser actualizados por Unix o que vayan a ser compartidos por varios procesos simultáneamente.

5.5.2.9. lock directory.

Esta opción (a veces llamada lock dir) especifica la localización de un directorio donde Samba almacenará ficheros de bloqueo de modo-denegación. Samba almacena otros archivos también en este directorio, tales como listas de navegación y su fichero de memoria compartida. Si WINS está activado, la base de datos WINS se escribe también en éste directorio. El valor por defecto para esta opción es especificado en el makefile de Samba; y suele ser /usr/local/samba/var/locks. Puedes cambiar esta localización como sigue:

```
[global]
  lock directory = /usr/local/samba/locks
```

Normalmente no necesitarás modificar esta opción, a menos que quieras mover los ficheros de bloqueo a otra localización, tal como */var/spool/locks*.

Cuadro 5.13: Bloqueos SMB de modo-denegación.

Bloqueo	**Descripción**
DENY_NONE	No deniega otras peticiones sobre el fichero.
DENY_ALL	Deniega todas las peticiones abiertas sobre el fichero actual.
DENY_READ	Deniega cualesquiera peticiones de sólo lectura abiertas sobre el fichero actual.
DENY_WRITE	Deniega cualesquiera peticiones de sólo escritura sobre el fichero actual.
DENY_DOS	Si se abrió en modo lectura, otros podrán leer pero no escribir sobre el fichero. Si se abrió para escribir, los demás no podrán abrirlo de ningún modo.
DENY_FCB	Obsoleto.

Capítulo 6

Usuarios, Seguridad y Dominios

Este capítulo trata la configuración de usuarios con el servidor Samba. En un principio parece bastante sencillo, pero pronto descubrirás que hay varios problemas accesorios que irán apareciendo. Un asunto donde los administradores Samba tienen dificultades es la autenticación del usuario; las contraseñas y los problemas de seguridad son, con mucho, las cuestiones más comunes tratadas en las listas de correo de soporte de Samba. Aprendiendo por qué distintos mecanismos de autenticación trabajan en ciertas arquitecturas (y no lo hacen en otras) puede ahorrarte una enorme cantidad de tiempo comprobado y depurando a los usuarios Samba en un futuro.

6.1. Usuarios y Grupos

Antes de empezar, necesitamos advertirte de que si estás conectando Samba con un Windows 98 o NT Workstation SP3, necesitas configurar tu servidor para utilizar contraseñas encriptadas antes de que puedas hacer la conexión, de otra forma, el cliente rechazará conectarse al servidor Samba. Esto se debe a que estos clientes Windows envían contraseñas encriptadas y Samba necesita ser configurado para esperarlas y desencriptarlas. Te enseñaremos como hacerlo mas adelante, en este mismo capítulo, suponiendo que no lo hayas abordado ya en el *Capítulo 2, Instalando Samba en un Sistema Unix*.

Vamos a empezar con un solo usuario. La forma más fácil de configurar un cliente es crear una cuenta Unix (y su directorio personal correspondiente) para es cliente en el servidor, y notificar a Samba la existencia de ese usuario. Posteriormente en el fichero de configuración de Samba se creara un recurso de disco compartido que apunte al directorio personal del usuario y restringiremos su acceso a traves de la opcion valid users. Por ejemplo:

```
[dave]
  path = /home/dave
  comment = Dave's Home Directory
  writeable = yes
  valid users = dave
```

La opción valid users especifica que usuarios pueden tener acceso al recurso compartido. En este caso sólo el usuario dave es autorizado a acceder a él. En los capitulos anteriores, especificamos que cualquier usuario podía acceder a un recurso compartido

de disco usando el parametro guest ok. Como nosotros no queremos conceder acceso de invitado, esta opción está ausente. Podemos dar ambos derechos de acceso, el de usuarios autenticados e invitados a un recurso específico si asi lo deseamos. La diferencia entre ambos radica en los derechos de acceso para cada uno de los ficheros.

Recuerda que puedes referirte al directorio personal del usuario usando la variable %H. Además, también puedes utilizar las variables de nombre de usuario Unix %u y de nombre de usuario cliente %U en las opciones. Por ejemplo:

```
[dave]
  comment= %U home directory
  writeable = yes
  valid users = Dave path = %H
```

Estos dos ejemplos funcionan mientras el usuario Unix que Samba utiliza para referirse al cliente tiene derechos de lectura/escritura sobre el directorio referenciado por la opcion path. En otras palabras, un cliente debe, primero, pasar los mecanismos de seguridad de Samba (por ejemplo, las contraseñas encriptadas, la opción valid users, etc...) así como los permisos sobre ficheros y directorios normales de Unix en el lado Unix de su usuario, antes de que pueda tener acceso de lectura/escritura a un recurso compartido.

Con un solo usuario accediendo a su directorio personal, nos ocupamos de los permisos de acceso cuando el sistema operativo crea la cuenta de usuario. En cualquier caso, si estás creando un directorio compartido para acceso de grupo, hay varios pasos adicionales que has de realizar. Vamos a echar un vistazo a un recurso compartido para el departamento de Contabilidad en el fichero smb.conf:

```
[accounting]
  comment = Accounting Department Directory
  writeable = yes
  valid users = @account
  path = /home/samba/accounting
  create mode = 0660
  directory mode = 0770
```

La primera cosa que notarás que hemos hecho de distinta forma es especificar @account como el usuario válido en lugar de uno o más nombres de usuario individuales. Este es un atajo para decir que los usuarios válidos están representados por el grupo Unix account. Estos usuarios necesitan ser añadidos al grupo account en el fichero de grupos del sistema (/etc/group o equivalente) para ser reconocidos como parte del grupo. Una vez que lo son, Samba los reconocerá como usuarios válidos para el recurso compartido.

A mayores, necesitarás crear un directorio compartido para que los miembros del grupo tengan acceso, que será mapeado a través de la opción de configuración path. Estos son los comandos Unix que crearán el directorio compartido para el departamento de Contabilidad (suponemos que /home/samba ya existen).

```
# mkdir /home/samba/accounting
# chgrp account /home/samba/accounting
# chmod 770 /home/samba/accounting
```

Hay otras dos opciones en el fichero smb.conf del ejemplo, que ya hemos visto en el capítulo anterior. Estas opciones son: create mode y directory mode. Señalan los

máximos permisos de ficheros y directorio que un nuevo fichero o directorio pueden tener. En este caso, hemos denegado el acceso global a los contenidos de ese recurso (esto es reforzado por el comando chmod, que ya hemos visto antes).

6.1.1. El recurso compartido [homes]

Vamos a volver a los recursos compartidos de usuario por un momento. Si tenemos varios usuarios a los que dar acceso compartido a sus directorios personales, probablemente queramos utilizar el recurso compartido especial [homes] del que hablamos en el *Capitulo 5, Navegación y Comparticiones Avanzadas de Unidades de Disco*. Con este recurso todo lo que necesitas decir es:

```
[homes]
  browseable = no
  writable = yes
```

El recurso compartido especial [homes] es una sección especial del fichero de configuración de Samba. Si un usuario intenta conectar a un recurso compartido ordinario que no aparece en el fichero smb.conf (como especificándolo con un UNC en Windows Explorer), Samba va a buscar el recurso compartido [homes]. Si no existe, el nombre del recurso compartido enviado a Samba se asume como un nombre de usuario y se busca como tal su contraseña en la base de datos (/etc/passwd o equivalente) del servidor Samba. Si aparece, Samba asume que el cliente es un usuario Unix intentando conectar a su directorio personal.

A modo de ejemplo, supongamos que sofia esta intentando conectar a un recurso compartido llamado [sofia] en el servidor Samba. No hay un recurso compartido con ese nombre en el fichero de configuración, pero si existe un recurso compartido [homes] y el usuario sofia está presente en la base de datos de contraseñas, por lo que Samba sigue los pasos siguientes:

1. Samba crea un nuevo recurso compartido llamado [sofia] con el path especificado en la seccion [homes]. Si no hay una opción path especificada en la sección [homes], Samba la inicializa a su directorio personal.

2. Samba inicializa las opciones del nuevo recurso a partir de las opciones por defecto de la sección [globals], y cualquier opción superior en [homes] con la excepción de browseable.

3. Samba conecta al cliente sofia a este nuevo recurso.

El recurso [homes] es una forma rápida y limpia de crear recursos compartidos para tu comunidad de usuarios sin tener que duplicar la información de la base de datos de contraseñas en el fichero smb.conf. Pero hay varios detalles que hemos de tener en cuenta:

- La sección [homes] puede representar cualquier cuenta en la máquina, lo que no es siempre deseable. Por ejemplo, podría crear un recurso compartido para root, bin, sys, uucp y similares. (Puedes establecer una opción invalid users para protegerte frente a esto).

- El significado de la opción browseable es diferente de los otros recursos; indica sólo que la sección [homes] no aparecerá en la lista local de exploración, pero

no que el recurso [alice] no lo hará. Cuando se crea la sección [alice] (despues de la conexión inicial), usara el valor browseable de la sección [globals] para ese recurso y no el de la sección [homes].

Como mencionamos, no hay necesidad de una opción path en [homes] si los usuarios tienen directorios personales Unix en el fichero /etc/passwd del servidor. Has de asegurarte de que exista un directorio personal válido, de todas formas, Samba no creará automáticamente un directorio personal para un usuario, y rechazará una conexión si el directorio personal del usuario no existe o no es accesible.

6.2. Controlando el acceso a los recursos compartidos.

Con frecuencia, por razones de seguridad, necesitarás restringir los usuarios que pueden acceder a un determinado recurso compartido. Esto es muy fácil de hacer con Samba, dado que contiene gran cantidad de opciones para crear prácticamente cualquier configuración de seguridad. Vamos a estudiar una serie de configuraciones que te pueden interesar para crear la tuya propia.

Como ya dijimos, si estás conectando con Windows 98 o Windows NT 4.0 con el Service Pack 3.0 (o superior), tienes que tener en cuenta que estos clientes enviarán contraseñas encriptadas al servidor Samba. Si Samba no está configurado para recibirlas, rechazará la conexión continuamente. Este capítulo describe como configurar Samba para utilizar contraseñas encriptadas. Ve a la sección de 'Contraseñas'.

Hemos visto que ocurre cuando especificas usuarios válidos. Sin embargo, también puedes establecer una lista de usuarios no válidos -usuarios a los que nunca les será permitido acceder a Samba o a sus recursos. Esto se hace con la opción invalid users. Anteriormente hemos apuntado a un uso frecuente que se le da a esta opción: Un valor por defecto relacionado con la sección [homes] para asegurar que determinados usuarios y superusuarios del sistema no pueden ser manipulados o alterados para conseguir acceso a éste. Por ejemplo:

```
[global]
  invalid users = root bin daemon adm sync shutdown \
                  halt mail news uucp operator gopher
  auto services = dave peter bob

[homes]
  browsable = no
  writeable = yes
```

La opción invalid users, como la valid users, puede utilizar tanto nombres de grupo como nombres de usuario. En el caso de que un usuario o grupo apareciese en ambas listas, la opción invalid users tendría preferencia por lo que a ese usuario o grupo se le denegaría el acceso al recurso.

En el otro extremo, puedes especificar, de forma explícita, a que usuarios se les va a conceder acceso de superusuario (root) a un recurso, a través de la opción admin users. Por ejemplo:

```
[sales]
  path = /home/sales
  comment = Fiction Corp Sales Data
```

```
  writeable = yes
  valid users = tom dick harry
  admin users = mike
```

Esta opción admite también nombres de grupo y de usuario. Además puedes especificar grupos NIS precediéndolos con un símbolo @;si este grupo no se encuentra, Samba asumirá que te refieres a un grupo estándar de Unix.

Se cuidadoso al asignar a un grupo privilegios administrativos sobre un recurso. El equipo desarrollador de Samba recomienda encarecidamente evitar el uso de esta opción, porque en esencia lo que hace es dar derechos de superusuario sobre ese recurso a los usuarios o grupos especificados.

Si deseas forzar derechos de solo-lectura o solo-escritura a los usuarios que acceden a un recurso, puedes hacerlo con las opciones read list y write list, respectivamente. Estas opciones pueden ser usadas para restringir el acceso a un recurso que tenga derechos de escritura, o para dar derechos de escritura a determinados usuarios sobre un recurso creado como de solo-lectura, respectivamente. Por ejemplo:

```
[sales]
  path = /home/sales
  comment = Fiction Corp Sales Data
  read only = yes
  write list = tom dick
```

La opción write list no prevalece sobre los permisos de Unix. Es decir, si tú creaste el recurso sin dar a los usuarios de la lista write list permisos de escritura sobre el sistema Unix, les será denegado ese derecho independientemente del valor de la opción write list.

6.2.1. Acceso de Invitado.

Como mencionamos antes, tú puedes especificar que usuarios van a tener acceso de invitado a un recurso. Las opciones que controlan este tipo de acceso son fáciles de utilizar. La primera, guest account, señala la cuenta Unix que se asignará a los usuarios invitados cuando conecten con el servidor Samba. El valor por defecto para esta opción se establece durante la compilación, y normalmente es nobody. De todas formas, puedes establecer el valor de usuario invitado a ftp si tienes problemas accediendo a varios servicios del sistema.

Si quieres restringir el acceso, en un recurso determinado, sólo a los usuarios invitados - dicho de otra forma, todos los usuarios se conectarán como usuarios invitados cuando accedan a ese recurso - puedes utilizar la opción guest only unida a la opción guest ok, como muestra el siguiente ejemplo:

```
[sales]
  path = /home/sales
  comment = Fiction Corp Sales Data
  writeable = yes
  guest ok = yes
  guest account = ftp
  guest only = yes
```

Asegúrate de que especificas yes tanto para guest only como para guest ok; de otra forma, Samba no usará la cuenta de invitado que has indicado.

6.2.2. Opciones de control de acceso.

La Tabla 6-1 resume las opciones que puedes utilizar para controlar el acceso a los recursos.

Cuadro 6.1: Control de Acceso a los Recursos.

Opcion	Parámetros	Función	Defecto	Ámbito
admin users	Cadena (lista de nombres de usuario).	Indica la lista de usuarios que pueden efectuar operaciones como "root".	Ninguno	Recurso
valid users	Cadena (lista de nombres de usuario).	Indica la lista de usuarios que pueden conectarse a un recurso.	Ninguno	Recurso
invalid users	Cadena (lista de nombres de usuario).	Indica la lista de usuarios que NO tendrán acceso a un recurso.	Ninguno	Recurso
read list	Cadena (lista de nombres de usuario).	Indica la lista de usuarios que tendrán únicamente derecho de solo-lectura sobre un recurso modificable.	Ninguno	Recurso
write list	Cadena (lista de nombres de usuario).	Indica la lista de usuarios que tendrán derecho de lectura-escritura sobre un recurso de solo-lectura.	Ninguno	Recurso
max connections	Valor numérico.	Indica el máximo número de conexiones permitidas para un recurso en un momento dado.	0	Recurso
guest only (only guest)	Valor lógico (Si/No).	Indica si ese recurso permite solo acceso de invitado.	no	Recurso
guest account	Cadena (nombre de una cuenta).	Indica la cuenta que se utilizará para el acceso de invitado.	nobody	Recurso

6.2.2.1. La opción Admin users.

Esta opción establece una lista que usuarios que realizaran sus operaciones sobre los ficheros como si fueran root. Esto significa que pueden modificar o destruir cualquier trabajo de otro usuario, no importa cuales sean los permisos. Cualquier fichero que creen pertenecerá a root y usará el grupo por defecto del usuario administrador. La opción admin users se usa para conceder a los usuarios PC actuar como administradores para determinados recursos. Te recomendamos que evites el utilizar esta opción.

6.2.2.2. Valid users e invalid users.

Estas 2 opciones te permiten indicar los usuarios y grupos que tienen o no tienen acceso a un determinado recurso. Puedes utilizar una lista de usuarios separados por comas, o indicar un grupo NIS o Unix que ira precedido con el símbolo de la arroba @.

La regla más importante que has de recordar sobre estas dos opciones es que, a cualquier usuario o grupo que aparezca en la lista invalid users le será denegado el acceso, aunque esté incluido (sea cual sea la forma) en la lista valid users. Ninguna

de estas opciones tiene un valor por defecto. Si las dos opciones no tienen valor, se le permitirá el acceso al recurso a cualquier usuario.

6.2.2.3. Read list y write list.

Como las opciones valid users e invalid users, este par de opciones indican qué usuarios tienen derechos de solo-lectura sobre un recurso modificable y derechos de lectura-escritura sobre un recurso de solo-lectura, respectivamente. El valor de estas opciones es una lista de usuarios. Read list tiene preferencia sobre cualquier otro permiso Samba concedido - también sobre los permisos sobre ficheros en el servidor - para denegar a los usuarios el derecho de escritura. Write list tiene preferencia sobre otros permisos Samba para conceder derechos de escritura, pero no puede darlos si el usuario carece de ellos en el sistema Unix. Puedes especificar nombres de grupo Unix o NIS utilizando como prefijo la arroba @ (como por ejemplo @users). Ninguna de las dos opciones tiene asociado un valor por defecto.

6.2.2.4. Max connections.

Esta opción establece el máximo numero de conexiones clientes que puede tener un recurso en un determinado momento. Cualquier conexión que se intente establecer una vez alcanzado el máximo será rechazada. El valor por defecto es 0, que indica que se permite un numero ilimitado de conexiones. Puedes anular este valor para un recurso de la forma siguiente: [accounting] max connections = 30

6.2.2.5. Esta opción es útil en el caso de que necesites limitar el numero de usuarios que están accediendo a la vez a un programa con licencia o a un dato determinado. Guest only.

Esta opción a nivel de recurso (algunas veces llamada only guest) obliga a que una conexión con un recurso se haga a través del usuario que establece la opción guest account. El recurso al que se aplique esta opción debe especificar de forma explícita guest ok=yes para que la opción sea reconocida por Samba. El valor por defecto es no.

6.2.2.6. Guest account.

Esta opción establece el nombre de la cuenta que se utilizará para el acceso de invitado a los recursos de Samba. Su valor por defecto varia de un sistema a otro, pero normalmente se establece a nobody. Algunos cuentas de usuario por defecto tienen problemas conectando como usuarios invitados. Si esto ocurre en tu sistema, el equipo Samba recomienda usar la cuenta ftp como cuenta de invitado.

6.2.3. Opciones de Usuario.

La tabla 6-2 muestra dos opciones adicionales que Samba puede utilizar para corregir las incompatibilidades entre nombres de usuario Unix y Windows.

6.2.3.1. Username map.

Los nombres de usuario cliente en una red Samba puede ser bastante largos (hasta 255 caracteres), mientras que los nombres de usuario en redes Unix no pueden tener más de 8 caracteres. Esto significa que un usuario individual puede tener un nombre de

usuario en un cliente y otro (más corto) en el servidor Samba. Puedes solucionar esto mapeando un nombre de usuario de formato libre a un nombre de usuario Unix de 8 o menos caracteres. Este mapeado se graba en un fichero de texto estándar, usando un formato que explicaremos pronto. Hecho esto, indicarás la ruta a Samba mediante la opción global username map. Asegúrate de restringir acceso a este fichero; establece su propietario a root y deniega el acceso de escritura a los demás. De no hacerlo así, un intruso que pueda acceder a ese fichero podría mapear su nombre de usuario al usuario root del servidor Samba.

Puedes especificar esta opción como sigue:

```
[global]
  username map = /etc/samba/usermap.txt
```

Cada una de las entradas en el fichero de mapeado de usuarios debe seguir el formato siguiente: El nombre de usuario Unix, seguido por un signo igual (=), seguido por uno o más nombres de usuario cliente separados por espacios. Ten en cuenta que a menos que se indique lo contrario (por ejemplo, en una conexión de invitado), Samba esperará que los dos usuarios tengan la misma contraseña. También puedes mapear grupos NT a uno o más grupos Unix usando el símbolo de arroba @. Algunos ejemplos serían:

```
jarwin = JosephArwin
manderso = MarkAnderson
users = @account
```

Además, puedes usar el asterisco para especificar un comodín que indique cualquier nombre de usuario cliente como una entrada en el fichero de mapa de usuarios:

```
Nobody = *
```

Se pueden incluir comentarios en el fichero precediendo las líneas con los símbolos (#) y (;).

Ten en cuenta que también puedes utilizar este fichero para redirigir un usuario Unix a otro usuario. Ten cuidado al hacerlo porque Samba y tu cliente no avisarán de esto al usuario y Samba puede estar esperando otra contraseña distinta.

6.2.3.2. Username level

Los clientes SMB (como Windows) normalmente envían los nombres de usuario en la conexión Samba en mayúsculas, es decir, los nombres de usuario no son necesariamente sensibles a la diferencia entre minúsculas y mayúsculas. En un servidor Unix, si lo son: el usuario ANDY es distinto del usuario andy. Por defecto, Samba se enfrenta a esto haciendo lo siguiente:

1. Buscando una cuenta de cliente con el nombre exacto enviado por el cliente.
2. Comprobando el nombre de usuario en minúsculas.
3. Comprobando el nombre de usuario en minúsculas con la primera letra en mayúsculas.

Si deseas que Samba intente más combinaciones de minúsculas y mayúsculas, puedes usar la opción de configuración global username level. Esta opción toma un valor entero que especifica cuantas letras en el nombre de usuario serán pasadas a mayúsculas al intentar conectar con un recurso. Puedes establecer estas opciones como sigue:

```
[global]
  username level = 3
```

En este caso, Samba intentará todas las permutaciones de nombres de usuario que pueda teniendo tres letras mayúsculas. Cuanto mayor sea el número, más cálculos deberá hacer Samba para comprobar el nombre y tanto más durará la autenticación.

6.3. Seguridad y autenticación.

En este capítulo trataremos como Samba autentifica a los usuarios. Cada usuario que intente conectar a un recurso que no permita acceso de invitado deberá suministrar una contraseña para efectuar la conexión. Lo que Samba hace con esa contraseña - y en consecuencia la estrategia que utiliza para autentificar al usuario -es trabajo de la opción de configuración security-. Actualmente hay cuatro niveles de seguridad utilizados por Samba en sus redes: share, user, server y domain.

Seguridad añivel de recurso (share). Cada recurso en el grupo de trabajo tiene una o más contraseñas asociadas con él. Cualquiera que conozca una contraseña valida puede acceder al recurso.

Seguridad añivel de usuario (user). Cada recurso en el grupo de trabajo se configura para permitir acceso a determinados usuarios. Con cada conexión inicial, el servidor Samba verifica los usuarios y sus contraseñas para permitirles el acceso al recurso.

Seguridad añivel de servidor (server). Es la misma que a nivel de usuario (user), pero en esta el servidor Samba utiliza otro servidor SMB para validar los usuarios y sus contraseñas antes de conceder el acceso.

Seguridad añivel de dominio (domain). Samba se convierte en un miembro de un dominio Windows y utiliza al Controlador Primario del Dominio (PDC) para llevar a cabo la autenticación. Una vez autenticado, al usuario se le da un atributo especial que le permite acceso a todos los recursos a los que tenga derechos de acceso. Con este atributo, el PDC no tendrá que volver a validar al usuario cada vez que intente conectarse a otro recurso dentro del dominio.

Cada una de estas políticas de seguridad puede ser implementada a través de la opción global security, como se muestra en la tabla 6.3.

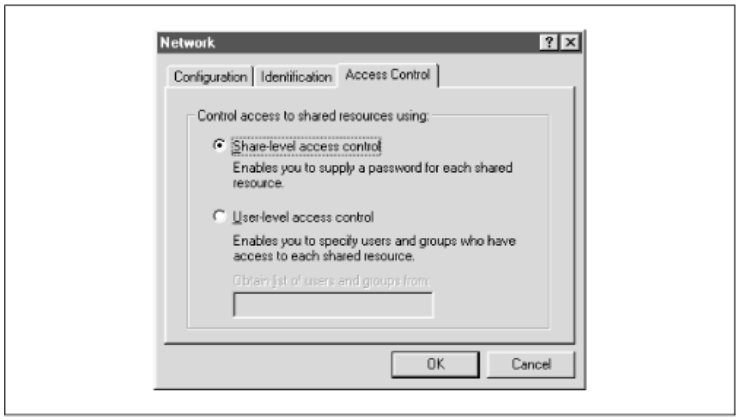
6.3.1. Seguridad a nivel de recurso (share).

En este caso cada recurso tiene una o más contraseñas asociadas con él. Se diferencia de los demás modos de seguridad en que aquí no hay restricciones como quién puede acceder al recurso, siempre que los usuarios conozcan la contraseña correcta. Los recursos pueden tener distintas contraseñas. Por ejemplo, una contraseña puede conceder sólo derechos de lectura, otra puede conceder derechos de lectura-escritura,

etc. La seguridad se mantiene mientras usuarios no autorizados no descubran las contraseñas para un recurso al que no deberían poder acceder.

OS/2 y Windows 95/98 soportan seguridad a nivel de recurso. En Windows 95/98 puedes establecerla usando la pestaña de Control de Acceso en la opción Red del Panel de Control. Una vez aquí marcas Control de acceso a los recursos (que desmarca la opción de Control de acceso de los usuarios) como se muestra en la figura 6-1 y pulsas el botón de Aceptar.

Figura 6.1: Control de Acceso.



Ahora, haces clic en un recurso -como un disco duro o un CD-ROM- y eliges el menú Propiedades. Se abrirá el cuadro de diálogo de propiedades del recurso. Elige la pestaña Compartir y activa el recurso como Compartido Como. Desde aquí, puedes configurar como se presentará el recurso a los usuarios individuales, así como asignar si el recurso será de solo-lectura, de lectura-escritura o una mezcla, dependiendo de la contraseña que se utilice.

Puedes estar pensando que este modelo de seguridad no es el más correcto para Samba - y tendrás razón. De hecho, si activas la opción security=share en el fichero de configuración de Samba, seguirá utilizando las combinaciones de nombre de usuario/contraseña de los ficheros de contraseñas del sistema para autentificar al usuario. Mas concretamente, Samba seguirá los siguientes pasos cuando un cliente solicite una conexión usando una política de seguridad a nivel de recurso:

1. Cuando se solicita la conexión, Samba aceptará la contraseña y (si se envía) el nombre de usuario del cliente.

2. Si el recurso es guest only, al usuario se le concede acceso inmediatamente al recurso, pero con los derechos del usuario especificado por el parámetro guest account; no se realiza ninguna comprobación sobre la contraseña.

3. Para otros recursos, Samba añade el nombre de usuario a una lista de usuarios que tienen permitido el acceso a ese recurso. Entonces intenta validar la contraseña recibida como complemento a ese nombre de usuario. Si tiene éxito, Samba concede el acceso los derechos asignados a ese usuario y este no necesitara autenticarse de nuevo a no ser que se establezca una opción revalidate=yes dentro de la configuración del recurso.

4. Si la autenticación no tiene éxito, Samba intentará validar la contraseña frente a una lista de usuarios que ha sido creada previamente utilizando los intentos de conexión, así como cualquiera especificada en la configuración del recurso. Si la contraseña no coincide con ningún nombre de usuario (como se establece en el fichero de contraseñas del sistema, normalmente /etc/password), no se le concederá acceso bajo ese nombre.

5. De cualquier forma, si el recurso tiene una opción guest ok o public, el usuario por defecto accederá con los derechos del usuario que se especifique en la opción guest account.

Con la opción de configuración username puedes establecer inicialmente, los usuarios de seguridad a nivel de recurso, tal como se muestra:

```
[global]
  security = share

[accounting1]
  path = /home/samba/accounting1
  guest ok = no
  writable = yes
  username = davecb, pkelly, andyo
```

Así, cuando un usuario intenta conectar con el recurso, Samba comprobará la contraseña enviada por este contra cada uno de los usuarios de su lista, además de las contraseñas de los usuarios davecb, pkelly y andyo. Si cualquiera de ellas coincide, se realizará la conexión, en otro caso, ésta fallará.

6.3.1.1. Opciones de seguridad a nivel de recurso

La Tabla 6.4 muestra las opciones más comunes asociadas con este nivel de seguridad:

6.3.1.2. Only user.

Esta opción lógica indica cuando Samba permitirá conexiones usando este nivel de seguridad basándose solamente en los usuarios especificados en la opción username, en lugar de aquellos establecidos en la lista interna de Samba. El valor por defecto para esta opción es no. Pero puedes saltártela para un recurso determinado de la forma siguiente:

```
[global]
  security = share

[data]
  username = andy, peter, valerie
  only user = yes
```

6.3.1.3. Username.

Esta opción establece una lista de usuarios contra los que Samba testeará una contraseña para permitir la conexión. Se usa normalmente con clientes que tienen seguridad a nivel de recurso basada exclusivamente en una contraseña -en este caso, una que coincide con la contraseña de determinado usuario:

```
[global]
security = share

[data]
username = andy, peter, terry
```

Te recomendamos que no la utilices a menos que estés instalando un servidor con seguridad a nivel de recurso.

6.3.2. Seguridad a nivel de usuario.

Es el modo más aconsejable de seguridad para trabajar con Samba. Con él, cada recurso tiene asignados determinados usuarios que pueden acceder a él. Cuando un usuario solicita una conexión a un recurso, Samba lo autentifica validando el nombre de usuario y la contraseña frente a los usuarios autorizados en el fichero de configuración y las contraseñas almacenadas en la base de datos del servidor Samba. Como mencionamos antes en este mismo capitulo, una manera de establecer que usuarios tienen acceso a un recurso especifico es usando la opción valid users para ese recurso:

```
[global]
  security = user

[accounting1]
  writable = yes
  valid users = bob, joe, sandy
```

A cada uno de los usuarios en la lista se le permitirá el acceso al recurso si la contraseña proporcionada coincide con la establecida en la base de datos de sistema en el servidor. Si esta autenticación inicial tiene éxito, el usuario no necesita volver a escribir la contraseña para acceder a este recurso a no ser que se haya activado la opción revalidate=yes.

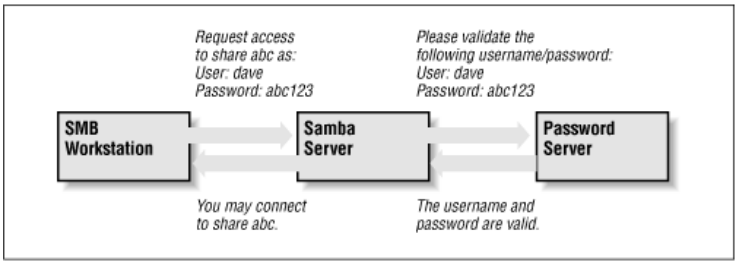
Las contraseñas pueden ser enviadas al servidor Samba de forma encriptada o no. Si tienes ambos tipos de sistemas en tu red, debes asegurarte que las contraseñas de cada usuario están grabadas tanto en la base de datos de cuentas tradicional como en la base de datos de contraseñas encriptadas. De esta forma, los usuarios autorizados tendrán acceso al recurso desde cualquier tipo de cliente. De todas formas, te recomendamos que si la seguridad es una prioridad utilices contraseñas encriptadas y abandones las no encriptadas. La sección contraseñas de este capitulo explica como utilizar contraseñas tanto encriptadas como no.

6.3.3. Seguridad a nivel de servidor.

La seguridad a nivel de servidor es parecida a la de usuario. Sin embargo, con la seguridad a nivel de servidor, Samba delega la autenticación de la contraseña a otro

servidor de contraseñas SMB, normalmente otro servidor Samba o un Windows NT Server actuando como PDC en la red. Fíjate en que Samba todavía mantiene una lista de recursos y configuraciones en su fichero smb.conf. Cuando un cliente intenta hacer una conexión a un recurso determinado, Samba comprueba que el usuario está autorizado a conectar a ese recurso, entonces intenta validar la contraseña conectando con el servidor SMB a través de un protocolo conocido y presentándole el nombre y la contraseña. Si la contraseña es aceptada, se establecerá la conexión con el cliente. Este proceso queda reflejado en la Figura 6.2.

Figura 6.2: Una configuración típica usando seguridad a nivel de servidor



Puedes configurar Samba para usar un servidor de contraseñas distinto con el uso de la opción global password server:

```
[global]
  security = server
  password server = PHOENIX120 HYDRA134
```

Fíjate en que puedes especificar mas de un servidor como valor de la opción password server; Samba ira recorriendo la lista en el caso de que no sea posible contactar con el primer servidor. Ten en cuenta que los nombres de estos servidores son nombres NetBIOS, no nombres DNS o sus IP equivalentes. Asimismo, si uno de los servidores rechaza la contraseña, la conexión fallara automáticamente, Samba no intentara probar en otro servidor.

Un aviso: Cuando uses esta opción seguirás necesitando una cuenta que represente al usuario en el servidor Samba general. Esto se debe a que el sistema operativo Unix necesita un nombre de usuario para realizar determinadas operaciones de entrada/salida (I/O). El método aconsejable es dar al usuario una cuenta en el servidor Samba pero desactivar su contraseña reemplazándola en el fichero de contraseñas del sistema (por ejemplo /etc/passwd) por un asterisco (*).

6.3.4. Seguridad a nivel de dominio.

La seguridad a nivel de dominio es parecida a la de nivel de servidor. Sin embargo, con la seguridad a nivel de dominio, el servidor Samba está actuando como miembro de un dominio Windows. Recuerda del capítulo 1 que cada dominio tiene un Controlador de Dominio, que es normalmente un Windows NT Server que ofrece la autenticación de las contraseñas. Incluir estos controladores proporciona al grupo un servidor de contraseñas definitivo. Los controladores de dominio mantienen los usuarios y contraseñas en sus propios módulos de autenticación de seguridad (SAM), y autentifican a cada usuario cuando este conecta por primera vez y cuando desea acceder a un recurso en otra maquina.

Como mencionamos antes en este capitulo, Samba tiene una prestación similar al ofrecer seguridad a nivel de usuario, pero esta opción está centralizada en Unix y asume que la autenticación se produce a través de los ficheros de contraseñas de Unix. Si la maquina Unix es parte de un dominio NIS o NIS+ Samba autenticará a los usuarios de forma transparente contra un fichero de contraseñas compartido, como es típico en Unix. Hecho esto, Samba ofrecerá acceso al dominio NIS o NIS+ desde Windows. Desde luego, no existe relación alguna entre el concepto de dominio NIS o NIS+ y el concepto de dominio de Windows.

Con la seguridad a nivel de dominio, tenemos la opción de usar el mecanismo nativo de NT. Esto tiene una serie de ventajas:

- Proporciona una mejor integración con NT: hay menos 'arreglos' en las opciones del smb.conf referidas a los dominios que con la mayoría de las posibilidades de Windows. Esto va a permitir utilizar de forma extensiva las opciones de administración de NT, como el Administrador de Usuarios para Dominios que permitirá a los usuarios individuales de los PC tratar los servidores Samba como si fueran grandes máquinas NT.

- Con la mejor integración vienen depuraciones del protocolo y del código, lo que permite al equipo Samba seguir con la evolución de la implementación NT. NT Service Pack 4 soluciona determinados problemas en el protocolo, y la mejor integración de Samba hace más fácil identificar y adaptarse a estos cambios.

- Hay una menor carga sobre el PDC porque hay una conexión permanente menos entre el y el servidor Samba. Independientemente del protocolo usado por la opción security=server, el servidor Samba puede hacer un Procedimiento de Llamada Remota (RPC) solo cuando necesita información de autenticación. No necesita mantener una conexión permanente para esto.

- Finalmente, el procedimiento de autenticación de dominio en NT devuelve todos los atributos del usuario, no solo si ha tenido o no éxito. Los atributos incluyen una lista larga y orientada a la red de los identificadores Unix, grupos NT, y mucha más información. Esto incluye:

  - Nombre de usuario
  - Nombre completo
  - Descripción
  - Identificador de seguridad (una extensión del identificador Unix orientada al dominio)
  - Pertenencia a grupos NT
  - Horas de entrada, y en su caso si hay que forzar al usuario a salir inmediatamente.
  - Puestos de red que el usuario esta autorizado a utilizar
  - Fecha de expiración de la cuenta
  - Directorio personal
  - Secuencia de entrada
  - Perfil
  - Tipo de cuenta

Los desarrolladores de Samba utilizaron seguridad a nivel de dominio en Samba versión 2.0.4 para permitirle añadir y eliminar usuarios del dominio de forma semiautomática. Además, añadió soporte para otras prestaciones al estilo NT, como soportar listas de control de acceso y cambiar los permisos de los ficheros desde el cliente.

La ventaja de esta aproximación es una menor administración; hay una sola base de datos de autenticación que mantener sincronizada. La única administración local en el servidor Samba será la creación de directorios para los usuarios y el mantenimiento del fichero /etc/passwd para tener sus identificadores y grupos al día.

6.3.5. Añadiendo un servidor Samba a un dominio Windows.

Si ya tienes un dominio NT, puedes fácilmente añadir un servidor Samba a este. Primero, necesitaras parar los demonios Samba. Entonces, añades el servidor Samba al dominio NT en el PDC usando el 'Administrador Windows NT Server para Dominios'. Cuando te pregunte por el tipo de servidor, eliges 'Windows NT Workstation o Server', y le das el nombre NetBIOS del servidor Samba. Esto crea la cuenta de la maquina en el servidor NT. Después, generas una contraseña tipo Microsoft usando la utilidad smbpasswd, esto se explica en la siguiente sección. Por ejemplo, si nuestro dominio es SIMPLE y el PDC de Windows NT es beowulf, podemos usar el siguiente comando en el servidor Samba:

```
smbpasswd -j SIMPLE -r beowulf
```

Finalmente, añade las siguientes opciones a la sección [global] del fichero smb.conf y reinicia los demonios Samba:

```
[global]
  security = domain
  domain logons = yes
  workgroup = SIMPLE
  password server = beowulf
```

Samba ahora estará configurado para usar la seguridad a nivel de dominio. La opción *domain logins* se explica mas adelante en este capitulo.

6.4. Contraseñas

Las contraseñas son un asunto problemático con Samba. Tanto es así, que la mayoría de las veces son el principal problema que los usuarios se encuentran cuando instalan Samba, y generan la mayoría de las cuestiones enviadas a los grupos de soporte de Samba. En capítulos anteriores, evitamos la necesidad del uso de contraseñas a través de la opción guest ok en cada uno de nuestros ficheros de configuración, lo que nos permitía realizar conexiones sin tener que autentificar contraseñas. No obstante, ahora hemos de profundizar mas en Samba para saber que esta pasando en la red.

Las contraseñas enviadas desde clientes individuales pueden ser encriptadas o no encriptadas. Las contraseñas encriptadas son, desde luego, más seguras. Una contraseña no encriptada puede ser leída fácilmente con un programa de lectura de paquetes ('sniffer'), como el modificado tcpdump que hemos usado en el Capitulo 3, Configurando los clientes Windows. Que las contraseñas sean encriptadas depende del sistema

operativo que el cliente este usando para conectar con Samba. La Tabla 6-5 muestra que sistemas operativos Windows encriptan sus contraseñas antes de enviarlas al Controlador Primario de Dominio para su autenticación. Si tu cliente no es Windows, comprueba la documentación del sistema para saber si las contraseñas SMB son encriptadas.

Actualmente se usan dos sistemas de encriptación: Uno para los clientes Windows 95 y 98 que sigue utilizando el estilo de Microsoft LAN Manager, y otro distinto para Windows NT clientes y servidores. Windows 95 y 98 utilizan un viejo sistema de encriptación derivado del programa de red LAN Manager, mientras que los sistemas Windows NT tanto clientes como servidores utilizan un sistema nuevo.

Si las contraseñas encriptadas están soportadas, Samba las almacena en un fichero llamado smbpasswd. Por defecto, este fichero esta situado en el directorio private de la distribución Samba (/usr/local/samba/private). Al mismo tiempo, los clientes almacenan una versión encriptada de la contraseña del usuario en su propio sistema. La contraseña sin encriptar (en texto plano) nunca se almacena en ningún sistema. Cada uno de ellos almacena la contraseña encriptada según un algoritmo conocido cuando esta se establece o cambia.

Cuando un cliente solicita una conexión a un servidor SMB que soporte contraseñas encriptadas (como Samba o Windows NT) los dos ordenadores llevan a cabo las siguientes negociaciones:

1. El cliente intenta negociar un protocolo con el servidor.
2. El servidor responde con un protocolo e indica que soporta contraseñas encriptadas. En este momento, devuelve una cadena de 8 bytes generada aleatoriamente.
3. El cliente utiliza esta cadena como una llave para encriptar la ya encriptada contraseña usando un algoritmo predefinido por el protocolo negociado. Entonces envía el resultado al servidor.
4. El servidor realiza el mismo proceso con la contraseña almacenada en su propia base de datos. Si los resultados coinciden, las contraseñas son equivalentes y el usuario es autenticado.

Ten en cuenta que, aunque las contraseñas originales no están involucradas en el proceso de autenticación, has de ser muy cuidadoso conque las contraseñas encriptadas almacenadas en el fichero smbpasswd estén a salvo de usuarios no autorizados. Si estas contraseñas están desprotegidas, un usuario no autorizado puede penetrar en el sistema reproduciendo los pasos del algoritmo del que hablamos antes. Las contraseñas encriptadas son tan delicadas como las contraseñas de texto plano -esto se conoce como equivalencia-texto plano en el mundo de la criptografía-. Desde luego, debes asegurarte que los clientes también protejan sus contraseñas.

Puedes configurar Samba para aceptar contraseñas encriptadas añadiendo lo siguiente en la sección [global] del fichero smb.conf. Ten en cuenta que especificamos completa la ruta del fichero de contraseñas de Samba:

```
[global]
  security = user
  encrypt passwords = yes
  smb passwd file = /usr/local/samba/private/smbpasswd
```

Samba, de todas formas, no aceptara ningún usuario hasta que el fichero *smbpasswd* haya sido inicializado.

Cuadro 6.3: Opciones Adicionales.

Opcion	Parámetros	Función	Defecto	Ámbito
username map	Cadena (Ruta completa al fichero)	Indica el nombre del fichero de mapas de usuario	Ninguno	Global
username level	Valor numérico	Indica el número de letras mayúsculas que se utilizarán al intentar localizar un nombre de usuario	0	Global

Cuadro 6.4: Opciones de seguridad

Opción	Parámetros	Función	Defecto	Ámbito
security	domain, server, share o user	Indica el tipo de seguridad que usará el servidor Samba	user (Samba 2.0) o share (Samba 1.9)	Global

Cuadro 6.5: Opciones de seguridad a nivel de recurso.

Opción	Parámetros	Función	Valor por defecto	Ámbito
only user	Booleano	Indica cuando los nombres de usuario especificados por username serán los únicos permitidos.	no	Recurso
username	Cadena (lista de usuarios)	Especifica una lista de usuarios contra los que se comprobará la validez de la contraseña.	ninguno	Recurso

Cuadro 6.6: Sistemas Operativos Windows con Contraseñas Encriptadas.

Sistema Operativo	Encriptado o No-Encriptado
Windows 95	No-Encriptado
Windows 95 con la Actualización SMB	Encriptado
Windows 98	Encriptado
Windows NT 3.x	No-Encriptado
Windows NT 4 anterior al SP3	No-Encriptado
Windows NT 4 despues del SP3	Encriptado

6.4.1. Deshabilitando contraseñas encriptadas en el cliente.

Aunque la autenticación Unix ha estado en uso durante décadas, incluyendo el uso de telnet y rlogin para tener acceso a través de Internet, contienen riesgos de seguridad bien conocidos. Las contraseñas en texto plano son enviadas a través de Internet y pueden ser leídas desde los paquetes TCP por cualquier fisgón que probablemente no tenga buenas intenciones. Aun así, si crees que tu red es segura y quieres usar el estándar Unix de autenticación /etc/passwd para todos los clientes, puedes hacerlo, pero deberás desactivar las contraseñas encriptadas en aquellos sistemas que las usan por defecto.

Para conseguirlo, debes modificar el registro de Windows instalando dos ficheros en cada sistema. Dependiendo del que sea los ficheros son, respectivamente, NT4_PlainPassword.reg o Win95_PlainPassword.reg. Puedes realizar esta instalación copiando el fichero.reg adecuado desde el directorio /docs de la distribución de Samba a un disquete DOS, y ejecutandolo desde el comando Ejecutar en el menu Inicio de Windows. Además, el fichero.reg de Windows 95 también funciona para Windows 98. Una vez que reinicies el ordenador cliente, este no encriptará sus contraseñas antes de enviarlas al servidor. Esto significa que la contraseña con equivalencia-texto plano puede ser vista en los paquetes TCP que se están enviando a lo largo de la red. De nuevo, te recomendamos que no utilices esto a no ser que estés totalmente seguro de que tu red es segura.

Si las contraseñas no están encriptadas, puedes indicarlo en tu fichero de configuración de Samba:

```
[global]
  security = user
  encrypt passwords = no
```

6.4.2. El fichero smbpasswd.

Samba almacena sus contraseñas encriptadas en un fichero llamado smbpasswd, que por defecto reside en /usr/local/samba/private. El fichero smbpasswd ha de ser guardado tan celosamente con el fichero passwd; debe ser colocado en un directorio donde solo el usuario root tenga derechos de lectura/escritura. Todos los demás usuarios no deberán ser capaces de leer el directorio. Además el fichero ha de tener todos los accesos cerrados a todo el mundo excepto a root.

Antes de que puedas usar contraseñas encriptadas, necesitas crear una entrada para cada usuario Unix en el fichero smbpasswd. La estructura de este fichero es similar a la del fichero passwd, pero tiene diferentes campos. La figura 6-3 ilustra el diseño del fichero smbpasswd; La entrada que se muestra es, de hecho, una sola línea del fichero.

Vamos a ver un poco cada uno de los campos:

Username (Nombre de usuario). Este es el nombre de usuario de la cuenta. Se toma directamente del fichero de contraseñas del sistema.

UID. Es el ID Unix del usuario. Como el nombre de usuario, se toma directamente del fichero de contraseñas del sistema y debe coincidir con el usuario que representa en este.

Información sobre la contraseña LAN Manager. Es una secuencia hexadecimal de 32 bits que representa la contraseña que utilizaran los clientes Windows 95 y Windows 98. Se obtiene del resultado de encriptar la secuencia KGS!@#$%

con un algoritmo DES de 56 bits usando la contraseña del usuario (reducida a 14 bytes y convertida a mayúsculas) repetida dos veces como clave. Si no hay contraseña para el usuario, los primeros 11 caracteres consistirán en la secuencia NO PASSWORD seguida por tantas X como sea necesario. Todo el mundo puede acceder a un recurso que no tenga establecida una contraseña. Por otro lado, si la contraseña ha sido desactivada, consistirá en 32 caracteres X. Samba no dará acceso a ningún usuario a menos que se establezca la opción null passwords.

Información sobre la contraseñaÑT. Es una secuencia hexadecimal de 32 bits que representa la contraseña que utilizarán los clientes NT. Se obtiene del resultado de cifrar la contraseña del usuario (representada como una secuencia Unicode 'little-endian' de 16 bits) con una secuencia de cifrado MD4. La contraseña no se convierte primero a mayúsculas.

Datos de la cuenta. Este campo consiste en 11 caracteres entre 2 corchetes ([]). Cualquiera de los siguientes caracteres puede aparecer en cualquier orden, el resto serán espacios:

U Esta cuenta es una cuenta de usuario estándar de Unix.

D Esta cuenta está desactivada y Samba no permitirá ningún acceso.

N Esta cuenta no tiene contraseña asociada

W Esta es una cuenta de confianza que puede ser utilizada para configurar Samba como Controlador Primario de Dominio (PDC) cuando se permita a equipos Windows NT unirse a su dominio.

Hora del ultimo cambio. Este código consiste en los caracteres LCT seguido por la representación hexadecimal del numero de segundos transcurridos desde el inicio de los tiempos (medianoche del 1 de Enero de 1970 en tiempo Unix) hasta que se modificó el fichero por ultima vez.

6.4.2.1. Añadiendo entradas a smbpasswd.

Hay varias maneras en que puedes añadir una entrada al fichero smbpasswd:

1. Puedes usar programa smbpasswd con la opción -a para añadir automáticamente cualquier usuario que ya posea un cuenta estándar Unix en el sistema. Este programa esta en el directorio /usr/local/samba/bin.

2. Puedes usar el programa addtosmbpass que esta en el directorio /usr/local/samba/bin. Es un script AWK que recorre un fichero de contraseñas del sistema y extrae el nombre de usuario y UID de cada entrada que quieras añadir al fichero de contraseñas SMB. Entonces añade campos con valores por defecto para el resto de las entradas de usuarios, que pueden ser actualizados mas tarde usando el programa smbpasswd. Para poder ejecutar este programa, probablemente deberás editar la primera línea para que apunte correctamente al awk de tu sistema.

3. En el caso de que ninguna de estas dos opciones sea la correcta para ti, puedes crear una entrada por defecto a mano en el fichero smbpasswd. Esta entrada debe estar en una sola línea y cada campo debe ir separado por dos puntos (:):

```
dave:500:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:[U ]:LCT-00000000:
```

Como vemos esta formado por el nombre de usuario y el UID tal como se especifica en el fichero de contraseñas del sistema, seguido por dos conjuntos de, exactamente 32 caracteres X, seguido por los datos de la cuenta y el momento del ultimo cambio. Una vez que añadas esta entrada, debes usar el programa smbpasswd para cambiar la contraseña del usuario.

6.4.2.2. Cambiando la Contraseña Encriptada.

Si necesitas cambiar las contraseñas encriptadas del fichero smbpasswd, puedes usar el programa smbpasswd. Ten en cuenta que comparten nombre, por lo que asegúrate de no confundir el fichero de contraseñas con el programa de cambio de contraseñas.

El programa smbpasswd es en su mayor parte, idéntico al programa passwd que se usa para cambiar las contraseñas de las cuentas Unix. Simplemente pide que entres tu antigua clave (a menos que seas el usuario root) y duplica la petición para la nueva. En la pantalla no se muestra ningún carácter.

```
# smbpasswd dave
Old SMB password:
New SMB password:
Retype new SMB password:
Password changed for user dave
```

Puedes mirar el fichero smbpasswd una vez que este comando termine su ejecución para comprobar que las hashes de las contraseñas de NT y Lan Manager se han grabado en sus respectivas posiciones. Una vez que los usuarios tienen contraseñas encriptadas en la base de datos, podrán conectar a los recursos compartidos usando contraseñas encriptadas.

6.5. Sincronización de las Contraseñas.

Tener una versión encriptada y una normal de la misma contraseña puede ser un problema cuando necesitas modificar las dos. Por suerte, Samba te ofrece una capacidad limitada para mantener las contraseñas sincronizadas. Samba tiene un par de opciones de configuración que pueden ser utilizadas para actualizar automáticamente una contraseña Unix normal cuando se actualiza la encriptada. Esta posibilidad puede activarse especificando la opción de configuración global *unix password sync*:

```
[global]
  encrypt passwords = yes
  smb passwd file = /usr/local/samba/private/smbpasswd
  unix password sync = yes
```

Con esta opción activada, Samba intentará cambiar la contraseña del sistema (como root) cuando la contraseña encriptada se cambie con smbpasswd. De todas formas, hay otras dos opciones que han de estar correctamente activadas para que esto funcione. La más fácil de las dos es passwd program. Esta opción solo establece el comando Unix usado para cambiar la contraseña de usuario estándar en el sistema. Por defecto su valor es /bin/passwd %u. En alguno sistemas Unix, esto es suficiente y no necesitas modificar nada, En otros, como Red Hat Linux, utiliza /usr/bin/passwd en su lugar.

Además, puedes querer cambiar esto a otro programa o script en algún momento en el futuro. Por ejemplo, suponemos que quieres usar un script llamado changepass para cambiar la contraseña de un usuario. Recuerda que puedes usar la variable %u para representar el nombre de usuario actual. El ejemplo seria:

```
[global]
  encrypt passwords = yes
  smb passwd file = /usr/local/samba/private/smbpasswd
  unix password sync = yes
  passwd program = changepass %u
```

Este programa ha de ser llamado como el usuario root cuando esta activada la opción *unix password sync*. Esto se debe a que Samba no tiene, necesariamente, la antigua contraseña de texto plano del usuario.

La opción mas complicada de configurar es passwd chat. Esta funciona como un script de chat Unix. Establece una serie de cadenas para enviar así como las respuestas que se esperan desde el programa especificado en la opción passwd program. Por ejemplo, el passwd chat es el siguiente por defecto. Los delimitadores son los espacios entre los grupos de caracteres:

```
passwd chat = *old*password* %o\n *new*password* %n\n *new*password* %n\n *changed*
```

El primer grupo representa la respuesta esperada desde el programa de cambio de contraseñas. Fíjate en que puede contener comodines (*), que ayudan a generalizar los programas de 'chat' para ser capaces de manejar una amplia variedad de salidas similares. Aquí *old *password * indica que Samba espera cualquier línea desde el programa de contraseñas que contenga las letras old seguidas por las letras password, sin que importe lo que venga a cada lado o en medio de ellas. Hecho esto, Samba esperará indefinidamente una coincidencia. Si no recibe la respuesta esperada, la contraseña fallará.

El segundo grupo indica lo que Samba enviara como respuesta una vez que coincidan los datos del primer grupo. En este caso se ve %o\n. Esta respuesta tiene dos partes: la variable %o representa la contraseña anterior, mientras que /n es un carácter de nueva-linea. Por tanto, realmente es como si escribiéramos la contraseña antigua en la entrada estándar del programa de cambio de contraseñas y 'pulsáramos' Enter.

Después de esto hay otro grupo de respuesta, seguido por los datos que se enviaran de vuelta al programa de cambio de contraseñas. (De hecho, este proceso de envío/respuesta continua indefinidamente en cualquier 'script' de 'chat' estándar de Unix). Este script continua hasta que se encuentra el patrón final.

Puedes ayudar a que coincidan las cadenas de respuesta enviadas desde el programa de contraseñas con los caracteres que se muestran en la Tabla 6-6. Además, puedes usar los que se muestran en la Tabla 6-7 para ayudarte a formular tu respuesta.

Cuadro 6.7: Caracteres de Respuesta para el programa de 'chat' de contraseñas

Carácter	**Definición**
*	Cero o más apariciones de cualquier caracter
" "	Te permite incluir cadenas que contengan espacios. Los asteriscos se siguen considerando comodines aun dentro de las comillas, y puedes representar una respuesta nula con unas comillas vacias.

Por ejemplo, puedes querer cambiar tu 'chat' de cambio de contraseñas al siguiente. Este maneja distintas situaciones donde no te será necesario introducir tu antigua contraseña. Además, manejará la nueva cadena all tokens updated successfully que envía Red Hat Linux:

```
passwd chat = *new password* %n\n *new password* %n\n *success*
```

De nuevo, el 'chat' por defecto será suficiente para la mayoría de los sistemas Unix. Si no es tu caso, puedes usar la opción global passwd chat debug para crear un nuevo 'chat' para el programa de cambio de contraseñas. Esta opción registra todo durante el proceso de 'chat' y su valor es lógico, como se muestra:

```
[global]
  encrypted passwords = yes
  smb passwd file = /usr/local/samba/private/smbpasswd
  unix password sync = yes
  passwd chat debug = yes
  log level = 100
```

Una vez que actives la prestación de depuración del 'chat', todas las entradas y salidas hechas o recibidas por Samba a través del 'chat' se enviaran a los registros de Samba con un numero máximo de 100, porque este es el valor que hemos introducido en la opción log level. Como esto puede generar multitud de errores de registro, puede ser más eficaz usar tu propio script, a través de la opción passwd program, en lugar de que */bin/passwd* registre todo lo que pasa durante el intercambio. De todas formas, asegúrate de proteger tus ficheros de registro con unos permisos estrictos y borrarlos una vez que tengas la información que necesitas, ya que en ellos aparece la contraseña en texto plano.

El sistema operativo en el que corra Samba puede tener unos requisitos muy estrictos para las contraseñas validas, con el fin de hacerlas blindadas frente a ataques. Los usuarios han de ser conscientes de estas restricciones cuando cambien sus contraseñas.

Antes hemos dicho que la sincronización de contraseñas es limitada. Esto es así porque no hay una sincronización inversa del fichero encriptado smbpasswd cuando un usuario cambia una contraseña estándar Unix. Hay varias soluciones para esto, incluyendo NIS e implementaciones gratuitas del standard de módulos de autenticación (PAM), pero realmente y por ahora ninguno de ellos soluciona el problema. En un futuro, cuando Windows 2000 se extienda, veremos una solución con el Protocolo Ligero de Acceso al Directorio (LDAP) que promete hacer de la sincronización de contraseñas un asunto del pasado.

6.5.1. Opciones de Configuración de las Contraseñas.

Las opciones de la Tabla 6-8 te ayudaran a trabajar con las contraseñas en Samba:

6.5.1.1. unix password sync.

Esta opción global permite a Samba actualizar el fichero de contraseñas estándar de Unix cuando un usuario cambia su contraseña encriptada. La contraseña encriptada esta almacenada en el servidor Samba en el fichero *smbpasswd*, que por defecto se encuentra en el directorio */usr/local/samba/private*. Puedes activar esta opción de la forma siguiente:

```
[global]
  unix password sync=yes
```

Con esta opción activada, Samba modifica la contraseña encriptada y, además, intenta modificar también la contraseña Unix estándar pasando el nombre de usuario y la contraseña nueva al programa de cambio de contraseñas especificado en la opción passwd program (de la que hablamos antes). Fíjate en que Samba no necesita, obligatoriamente, tener acceso a la contraseña en texto plano para este usuario, por lo que el programa de cambio de contraseñas debe ser ejecutado como root [1]. Si no se realiza el cambio de la contraseña Unix, sea por la razón que sea, tampoco lo hará la contraseña encriptada.

6.5.1.2. encrypt passwords.

Esta opción establece el uso de contraseñas encriptadas o de texto plano para la autenticación. Si la opción esta activada con yes Samba esperara que los clientes envíen contraseñas encriptadas:

```
encrypt passwords = yes
```

Por defecto, Windows NT con Service Pack 3 o superior y Windows 98 envían contraseñas encriptadas a través de la red. Si activas estas contraseñas, has de tener un fichero smbpasswd válido con los nombres de los usuarios que autenticarás con contraseñas encriptadas (Mira la sección "*El fichero smbpasswd*", en este mismo capitulo). Además, Samba debe saber la localización del fichero *smbpasswd*; si no está en el directorio por defecto (normalmente */usr/local/samba/private*) has de dárselo usando la opción smb passwd file. Si lo deseas, puedes usar la opción update encrypted para forzar a Samba a actualizar el fichero smbpasswd con las contraseñas encriptadas cada vez que un usuario se conecte con una contraseña no encriptada. Una estrategia muy común para asegurarse de que los equipos que necesitan autenticación con contraseñas encriptadas las reciben es con la opción include. Con ella , puedes crear ficheros individuales de configuración que se leerán en función del sistema operativo (%a) o el nombre del cliente (%m). Estos ficheros de configuración específicos pueden contener una opción *encrypted passwords=yes* que se activará sólo cuando los clientes estén conectados al servidor.

6.5.1.3. passwd program.

Se usa para especificar un programa en el servidor Samba Unix que Samba utilizará para actualizar las contraseñas estándar del sistema cuando se modifiquen las contraseñas encriptadas. Esta opción por defecto apunta al programa passwd localizado normalmente en el directorio */bin*. Cuando el programa se ejecuta pide el nombre de usuario, que se le suministra a través de la variable %u. El seguimiento de las entradas y salidas de este programa se hace a través de la opción passwd chat. La sección 'Sincronización de Contraseñas', en este capitulo, trata este tema en detalle.

[1] Esto se debe a que el programa Unix passwd, que es el usado la mayor parte de las veces, permite a root modificar la contraseña de un usuario sin solicitar la contraseña antigua.

6.5.1.4. passwd chat.

Establece una serie de envíos/respuestas similares a un programa 'chat' de Unix, que se usan para ejecutar el programa de cambio de contraseñas en el servidor Samba. La sección 'Sincronización de Contraseñas', en este capitulo, trata este tema en detalle.

6.5.1.5. passwd chat debug.

Si se establece a yes, la opción global passwd chat debug registra todo lo que Samba envía y recibe durante un 'chat' de contraseñas y se envían a los registros de Samba con un nivel de detalle de 100; necesitarás especificar log *level=100* para registrar esa información. La sección '*Sincronización de contraseñas*' en este capitulo, describe esta opción con más detalle. Sé consciente de que, si activas esta opción, las contraseñas en texto plano serán visibles en los ficheros de registro, lo que puede ser un problema de seguridad si no se tratan apropiadamente.

6.5.1.6. password level.

Con SMB, las contraseñas no encriptadas (o de texto plano) se envían en mayúsculas, igual que los nombres de usuarios de los que ya hablamos. Muchos usuarios Unix, de todas formas, eligen contraseñas con letras mayúsculas y minúsculas. Samba, por defecto, sólo intenta verificar la contraseña en minúsculas, y sin poner en mayúsculas la primera letra.

Como username level, hay una opción password level que se puede usar para intentar varias permutaciones de la contraseña con letras mayúsculas. Esta opción toma un valor entero que establece cuantas letras de la contraseña deben convertirse a mayúsculas cuando el cliente intente conectarse al recurso. Puedes establecer esta opción como sigue:

```
[global]
  password level = 3
```

En este caso, Samba intentara todas las permutaciones que pueda computar de la contraseña con tres letras mayúsculas. Cuanto mayor sea el nombre, mas cálculos habrá de hacer Samba para comprobar la contraseña y más tiempo llevara el proceso de conexión a un recurso.

6.5.1.7. update encrypted.

Para casos en los que esta en proceso de cambio entre contraseñas encriptadas y planas, esta opción proporciona ayuda durante la transición. Puedes activarla como sigue:

```
[global]
  update encrypted = yes
```

Esta opción indica a Samba que ha de crear una versión encriptada de la contraseña Unix de cada usuario cada vez que este se conecte a un recurso. Cuando esta opción se activa debes tener la opción *encrypt passwords=no* para que el cliente pase contraseñas planas que Samba utilizará para actualizar estos ficheros. Una vez que el usuario se conecta al menos una vez, puedes establecer *encrypt passwords=yes*, permitiendo usar sólo contraseñas encriptadas. El usuario debe tener una entrada válida en el fichero *smbpasswd* para que esta opción funcione.

6.5.1.8. null passwords.

Esta opción global le dice a Samba si permitir o no el acceso a usuarios que tienen contraseñas nulas (encriptadas o no) en sus cuentas. El valor por defecto es no, pero puedes activarlo con:

```
null passwords=yes
```

Te recomendamos encarecidamente que no utilices esta opción a menos que estés familiarizado con los riesgos de seguridad que implica, incluyendo acceso inadvertido de usuarios del sistema (como bin) en el fichero de contraseñas del sistema que tienen establecidas contraseñas nulas.

6.5.1.9. smb passwd file.

Esta opción global indica la localización del fichero de contraseñas encriptadas. Por defecto, esta establecida a /usr/local/samba/private. Puedes establecerlo a otro valor como sigue:

```
[global]
  smb passwd file = /etc/smbpasswd
```

Esta localización, por ejemplo, es común en las distribuciones Red Hat.

6.5.1.10. hosts equiv.

Esta opción global especifica el nombre de un fichero estándar Unix hosts.equiv que permitirá a los servidores o usuarios acceder a los recursos sin dar una contraseña. Puedes establecer la ubicación de este fichero como sigue:

```
[global]
  hosts equiv = /etc/hosts.equiv
```

El valor por defecto para esta opción no especifica ningún fichero. Te recomendamos que no utilices esta opción ya que supone un grave riesgo de seguridad.

6.5.1.11. use rhosts.

Esta opción global de configuración establece el nombre de un fichero de usuarios estándar Unix .rhosts que permitirá a los servidores ajenos acceder a recursos sin dar una contraseña. Puedes establecer la ubicación del fichero como sigue:

```
[global]
  use rhosts = /home/dave/ .rhosts
```

El valor por defecto para esta opción no especifica ningún fichero. Al igual que con la opción *hosts equiv* te recomendamos que no la utilices ya que supone un grave riesgo de seguridad.

6.6. Dominios Windows

Ahora que ya te sientes a gusto entre usuarios y dominios, te enseñaremos como configurar Samba para convertirse en un controlador primario de dominio (PDC) para los equipos Windows 9X y NT. ¿Por qué usar dominios? La respuesta no parece obvia hasta que miras detrás del telón, sobre todo con Windows 9X.

Hay que remarcar que, con los grupos tradicionales, Windows 9X simplemente acepta cualquier usuario y contraseña que utilices al acceder al equipo. En Windows 9X no existen los usuarios no autorizados; cuando un usuario nuevo accede al sistema, éste simplemente le pide una contraseña nueva y lo autentifica contra esa misma contraseña. La única vez que Windows 9X intenta utilizar la contraseña es cuando te conectas a otro recurso.

Por otro lado, la autenticación en los dominios es similar a los sistemas Unix. Para poder acceder al dominio son necesarios un nombre de usuario y una contraseña válidos, que son autenticados a través de la base de datos de contraseñas del controlador primario de dominio. Si la contraseña no es válida, se notifica inmediatamente al usuario y este no puede acceder al dominio.

Hay más buenas noticias: una vez que te has conectado al dominio, puedes acceder a cualquiera de los recursos de este para los que tengas derechos sin tener que reautenticarte. Dicho de otra forma, el PDC devuelve una señal al cliente que le permite acceder a cualquier recurso sin tener que consultar otra vez al PDC. Seguro que ya te has dado cuenta de la gran reducción que esto supone para el tráfico de la red, sin embargo puedes desactivar esto a través de la opción *revalidate*.

6.6.1. Configurando Samba para los Dominios Windows.

Si deseas que Samba actúe como PDC, utiliza la sección siguiente para configurar Samba y tus clientes de forma que admitan el acceso por dominios.

6.6.1.1. Clientes Windows 95/98.

Configurar Samba como PDC para Windows 9X es un poco desilusionante. Todo lo que necesitas hacer es asegurarte de que:

- Samba es el único PDC para ese grupo de trabajo.
- Hay un servidor WINS disponible en la red, sea un servidor Samba o Windows NT Server. (Consulta el *Capítulo 7, Resolución de Nombre e Impresión* para más información sobre WINS).
- Samba está usando seguridad a nivel de usuario, es decir, no permite la autenticación de contraseñas a nadie más. No querrás utilizar seguridad a nivel de dominio si el propio Samba está actuando como PDC.

En este punto, puedes insertar las siguientes opciones en tu fichero de configuración de Samba:

```
[global]
  workgroup = SIMPLE
  domain logons = yes
```

```
  # Be sure to set user-level security!
  security = user

  # Be sure to become the primary domain controller!
  os level = 34
  local master = yes
  preferred master = yes
  domain master = yes
```

La opción domain logons permite a Samba hacer una autenticación de dominio en nombre de otros clientes que lo soliciten. El nombre del dominio ha de ser el mismo que el del grupo de trabajo establecido en el fichero de configuración de Samba, en este caso: SIMPLE.

Después de esto, necesitas crear un recurso de disco compartido llamado [netlogon] que sea de solo lectura, no publico y no explorable. No importa a donde apunte sólo importa que los clientes Windows puedan conectarse a él.

```
[netlogon]
  comment = The domain logon service
  path = /export/samba/logon
  public = no
  writeable = no
  browsable = no
```

6.6.1.2. Clientes Windows NT

Si tienes clientes Windows NT en tu red, hay varios pasos más que has de seguir para que Samba pueda actuar como PDC para ellos.

> **AVISO**: Necesitaras usar Samba 2.1 o superior para asegurar que pueda funcionar como PDC para los clientes Windows NT. Antes de Samba 2.1, para los clientes Windows NT solo estaba disponible una autenticación limitada. En el momento en que se imprimió este libro, la ultima versión de Samba era la 2.0.5, pero la 2.1 esta disponible a través de una descarga CVS. Las instrucciones para descargar las versiones alpha de Samba están en el *Apéndice E, Descargando Samba con CVS*.

Igual que antes, necesitas asegurarte que Samba es el PDC para el grupo de trabajo actual y que esta usando la seguridad a nivel de usuario. Además, has de asegurarte de estar usando contraseñas encriptadas. Dicho de otra forma, modifica las opciones [global] del ejemplo anterior para incluir la opción *encrypted passwords=yes*:

```
[global]
  workgroup = SIMPLE
  encrypted passwords = yes
  domain logons = yes
  security = user
```

6.6.1.3. Crear cuentas de confianza para los clientes NT

Este paso es, exclusivamente, para los clientes NT. Todos los clientes NT que se conectan a un PDC hacen uso de las cuentas de confianza. Estas cuentas permiten a

una maquina conectarse con el PDC (y no con uno de sus recursos), lo que significa que el PDC puede verificar posteriores conexiones de los usuarios desde ese cliente. A nivel de utilización, una cuenta de confianza es idéntica a una cuenta de usuario. De hecho, vamos a utilizar cuentas de usuario estándar para simular cuentas de confianza.

El nombre de usuario de una cuenta de confianza para un equipo es el nombre del equipo con un signo de dólar añadido a el. Por ejemplo, si nuestra maquina NT se llama chimera el nombre de la cuenta será chimera$. La contraseña inicial de la cuenta será el nombre del equipo en minúsculas. Para reforzar la cuenta de confianza en el servidor Samba, necesitarás crear una cuenta Unix con el nombre de la maquina y una entrada con una contraseña encriptada en el fichero smbpasswd. Vamos a ver la primera parte. Aquí, sólo necesitaremos modificar el fichero /etc/passwd para soportar las cuentas de confianza; no hay necesidad de crear un directorio personal o asignar un 'shell' al usuario porque solo estamos interesados en que se permita el acceso ('login'). Por tanto, podemos crear una cuenta 'tonta' con la siguiente entrada:

```
chimera$:*:1000:900:Trust Account:/dev/null:/dev/null
```

Fíjate en que hemos desactivado el campo de contraseña insertando un *. Esto es porque Samba usará el fichero smbpasswd para guardar la contraseña, y no deseamos que nadie haga un telnet a la maquina usando esa cuenta. De hecho, el único valor aparte del nombre de la cuenta es el UID de la misma para la base de datos de contraseñas (1000). Este numero debe apuntar a un único ID de recurso en el servidor NT y no puede entrar en conflicto con ningún otro ID. Por ello, ningún usuario o grupo de NT puede apuntar a este recurso o se producirá un error de red.

Ahora, añade la contraseña encriptada usando el comando smbpasswd, como sigue:

```
# smbpasswd -a -m chimera
Added user chimera$
Password changed for user chimera$
```

La opción *-m* especifica que se está creando una cuenta de confianza. El programa smbpasswd establecerá la contraseña encriptada inicial como el nombre NetBIOS del ordenador en minúsculas; no necesitas introducirla. Cuando utilices esta opción en la línea de comandos, no pongas el signo de dólar ($) después del nombre del ordenador, se añadirá automáticamente. Una vez que se añade la contraseña encriptada, Samba está preparado para manejar accesos al dominio desde un cliente NT.

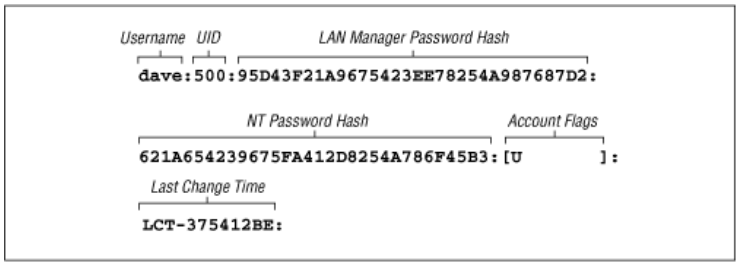
6.6.2. Configurando los clientes Windows para Accesos al Dominio

Una vez que tengas Windows configurado para accesos por dominio, necesitas configurar tus clientes Windows para que se conecten a este al iniciar.

6.6.2.1. Windows 95/98

Con Windows 95/98, esto puede hacerse a través de la configuración de Red del Panel de Control, seleccionando las Propiedades para el 'Cliente de redes Microsoft'. En este momento, verás una ventana de dialogo parecida a la de la figura 6-4. Selecciona la opción 'Conectarse a un dominio NT' en la parte de arriba de la ventana e introduce el nombre del dominio que aparece en el fichero de configuración de Samba como 'Nombre de dominio NT'. Ahora haz clic en Aceptar y reinicia el equipo.

Figura 6.3: Estructura una entrada en el fichero smbpasswd (actualmente es una sola linea).



Cuadro 6.8: Caracteres de Envio para el programa de 'chat' de contraseñas.

Carácter	**Definición**
%o	Antigua contraseña del usuario
%n	Nueva contraseña del usuario
\n	Caracter de Nueva-Linea
\r	Caracter de Retorno de Carro
\t	Caracter de Tabulacion
\a	Un espacio

Figura 6.4: Configurando un cliente Windows 9X para acceder al dominio

Cuadro 6.9: Opciones de configuración de las contraseñas.

Opción	**Parámetros**	**Función**	**Valor por defecto**	**Ámbito**
encrypt passwords	Lógico	Activa las contraseñas encriptadas.	no	Global
unix password sync	Lógico	Si su valor es *yes*, Samba actualiza las contraseñas estándar de Unix cuando un usuario cambia su contraseña enciptada.	no	Global
passwd chat	Carácter (comandos del "chat")	Establece la secuencia de comandos que se enviará al programa de contraseñas.	Mira la sección anterior en este capítulo	Global
passwd chat debug	Lógico	Envía la depuración del proceso de cambio de contraseñas a los ficheros de registro con una profundidad de 100	no	Global
passwd program	Carácter (comandos Unix)	Establece el programa a usar para cambiar las contraseñas.	/bin/passwd %u	Global
password level	Numérico	Establece el numero de permutaciones con letras mayúsculas que se usarán al comprobar una contraseña.	None	Global
update encrypted	Lógico	Si su valor es yes, Samba actualizará la contraseña encriptada cuando un usuario se conecte con una contraseña de texto plano.	no	Global
null passwords	Lógico	Si su valor es yes, Samba permitirá el acceso a usuarios con contraseñas nulas.	no	Global
smb passwd file	Carácter (Ruta completa al fichero)	Especifica el nombre del fichero de contraseñas encriptadas.	/usr/local/samba /private /smbpasswd	Global
hosts equiv	Carácter (Ruta completa al fichero)	Especifica el nombre del fichero que contiene los equipos y usuarios que se pueden conectar sin usar contraseña.	None	Global
use rhosts	Carácter (Ruta completa al fichero)	Especifica el nombre de un fichero .rhosts que permite a los usuarios conectarse sin usuar contraseña.	None	Global

AVISO: Si Windows te indica que ya estás conectado al dominio, probablemente ya tengas una conexión activa a un recurso de ese grupo de trabajo (como puede ser una unidad de red). Simplemente desconecta el recurso temporalmente haciendo clic con el botón derecho en su icono y seleccionando la opción 'Desconectar'.

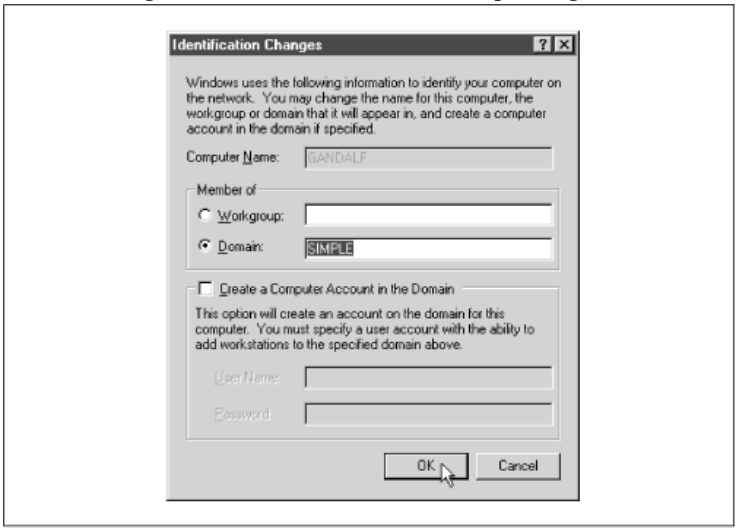
Cuando Windows reinicie, verás la ventana de acceso estándar pero con un campo adicional: el nombre de dominio. Este ya debería estar cubierto, así que introduce tu contraseña y haz clic en Aceptar. En este momento Windows consultara al PDC (Samba) para ver si la contraseña es correcta (puedes revisar los ficheros de registro si quieres ver esto en detalle). Si todo va bien, ¡felicidades!. Has configurado correctamente Samba para actuar como PDC para tus clientes Windows 95/98 y estos están correctamente conectados.

6.6.2.2. Windows NT 4.0

Para configurar los clientes Windows NT, abre la opción Red del Panel de Control y la primera pestaña que veas corresponderá a la identificación del ordenador.

Utiliza el botón 'Cambiar...' y veras una ventana de dialogo similar a la de la figura 6-5. En ella puedes convertir al cliente NT en miembro del dominio seleccionando el botón 'Dominio'. Ahí, escribe el nombre del dominio en el que quieres que se conecte; ha de ser el mismo que el nombre de grupo de trabajo que estableciste en el fichero de configuración de Samba. No marques la opción 'Crear una cuenta de equipo en el Dominio' porque Samba no soporta esto correctamente.

Figura 6.5: Configurando un cliente Windows NT para logeados de dominio.



AVISO: Como Windows 95/98, si NT te indica que ya estás conectado al dominio, probablemente ya tengas una conexión activa a un recurso de ese grupo de trabajo (como puede ser una unidad de red). Simplemente desconecta el recurso temporalmente haciendo clic con el botón derecho en su icono y seleccionando la opción 'Desconectar'.

Una vez que pulses el botón Aceptar, Windows te presentara una ventana de bienvenida al dominio. En este momento, necesitas reiniciar el ordenador y una vez que lo hagas te presentara una ventana de acceso similar a la de los clientes Windows 95/98. Ahora puedes acceder usando cualquier cuenta que ya tuvieras en el servidor Samba y que este configurada para permitir accesos.

AVISO: Asegúrate de seleccionar el dominio correcto en la ventana de dialogo de NT. Una vez seleccionado, puede llevarle un rato a NT construir la lista de dominios disponibles.

Una vez que introduzcas la contraseña, Windows NT consultará al PDC (Samba) si la contraseña es correcta. Como antes, puedes consultar los ficheros de registro para ver esta acción en detalle. Si funciona, ¡felicidades!. Has configurado correctamente Samba para actuar como PDC para tus clientes Windows NT y estos están correctamente conectados.

6.6.3. Opciones de Dominios

La tabla 6-9 muestra las opciones que se usan normalmente en los accesos por dominios.

Cuadro 6.11: Opciones de Logeado de Dominio de Windows 95/98

Opción	Parámetros	Función	Defecto	Ámbito
domain logons	Lógico	Indica si se utilizarán accesos a dominios Windows	no	Global
domain group map	Cadena (Ruta completa)	Nombre del fichero utilizado para traducir los grupos NT a grupos Unix	Ninguno	Global
domain user map	Cadena (Ruta completa)	Nombre del fichero usado para traducir los nombres de usuario de NT a nombres Unix	Ninguno	Global
local group map	Cadena (Ruta completa)	Nombre del fichero usado para traducir los grupos locales de NT a grupos Unix	Ninguno	Global
revalidate	Lógico	Si tiene valor yes, Samba fuerza a los usuarios a autenticarse cada vez que se conecten a un recurso.	no	Global

6.6.3.1. domain logons

Esta opción configura Samba para aceptar accesos por dominios actuando como PDC. Cuando un cliente se conecta correctamente al dominio, Samba devuelve un paquete especial al cliente que le permite acceder a los recursos de la red sin necesidad de consultar de nuevo al PDC para autenticarse. Fíjate en que Samba ha de utilizar la seguridad a nivel de usuario (security=user) y debe ser el PDC para que esta opción funcione. Además las maquinas Windows esperaran que en el servidor Samba exista un recurso compartido llamado [netlogon] (consulta la sección 'Configurando Samba para los Dominios Windows.')

6.6.3.2. domain group map

Esta opción señala la localización de un fichero de mapeado diseñado para traducir los nombres de grupos de Windows NT a nombres de grupo Unix. Este fichero debe residir en el servidor Samba. Por ejemplo:

```
/usr/local/samba/private/groups.mapping
```

El fichero tiene un formato sencillo:

```
UnixGroup=NTGroup
```

Un ejemplo es:

```
admin = Administrative
```

El grupo Unix especificado debe ser un grupo valido dentro del fichero /etc/group. El grupo NT ha de ser el nombre del grupo Unix al que quieres que pertenezca el cliente NT. Esta opción solo funciona con clientes NT.

6.6.3.3. domain user map

Esta opción especifica la localización de un fichero de mapeado diseñado para traducir los nombres de usuario Unix a nombres de usuario de Windows NT. Este fichero debe residir en el servidor Samba. Por ejemplo:

```
/usr/local/samba/private/domainuser.mapping
```

Este fichero tiene un formato sencillo:

```
UnixUsername = [\\ Domain\\] NTUserName
```

Una línea de ejemplo seria:

```
Joe= Joseph Miller
```

El nombre de usuario Unix especificado debe ser un nombre de usuario valido dentro del fichero /etc/passwd. El nombre Unix será aquel al que quieres que apunte el nombre de usuario de NT. Esta opción solo funciona con clientes NT.

> **AVISO**: Si deseas mas información sobre como usa Windows NT los nombres de usuario y grupos locales de dominio, te recomendamos el libro de Eric Pearce's 'Windows NT in a NutShell', publicado por O'Reilly.

6.6.3.4. local group map

Esta opción señala la localización de un fichero de mapeado diseñado para traducir los nombres de grupos locales de Windows NT a grupos Unix. Los grupos locales incluyen grupos como Administradores y Usuarios. Este fichero residirá en el servidor Samba. Por ejemplo:

```
/usr/local/samba/private/localgroup.mapping
```

Este fichero tiene un formato sencillo:

```
UnixGroup = [BUILTIN\] NTGroup
```

Un ejemplo seria:

```
root = BUILTIN\Administrators
```

Esta opción solo funciona con clientes NT. Para mas información te recomendamos el libro de Eric Pearce's 'Windows NT in a NutShell', publicado por O'Reilly.

6.6.3.5. revalidate

Esta opción a nivel de recurso le indica a Samba que fuerce a los usuarios a autenticar su contraseña cada vez que se conecten a un recurso diferente en un equipo, y no importa que nivel de seguridad este activado en el servidor Samba. El valor por defecto es no, lo que permite a los usuarios ser autenticados una sola vez. Puedes anular esto con:

```
revalidate=yes
```

Puedes usar esta opción para aumentar la seguridad en tu sistema. De todas formas, has de valorarlo frente a los inconvenientes de tener a los usuarios autenticándose cada vez que se conecten a un recurso.

6.7. Scripts de Entrada

Samba soporta la ejecución de scripts de entrada, que son scripts (.BAT o .CMD) que se ejecutan en el cliente cuando un usuario se conecta a un dominio Windows NT. Fíjate en que estos scripts se almacenan en el equipo Unix, pero son transportados a través de la red hasta el cliente y ejecutados una vez que el usuario se conecta. Tienen un valor incalculable como herramientas para configurar opciones de red para los usuarios cuando se conectan. El problema es que, como funcionan en Windows, deben usar los comandos de configuración de red de Windows.

> **AVISO**: Si deseas mas información sobre los comandos NET, te recomendamos el libro de Eric Pearce's 'Windows NT in a NutShell', publicado por O'Reilly.

Puedes indicar a Samba que utilice un script de entrada con la opción logon script, por ejemplo:

```
[global]
  domain logons = yes
  security = user
  workgroup = SIMPLE
  os level = 34
  local master = yes
  preferred master = yes
  domain master = yes
  logon script = %U.bat

[netlogon]
```

```
  comment = The domain logon service
  path = /export/samba/logon
  public = no
  writeable = no
  browsable = no
```

Fíjate en que en este ejemplo se usa la variable %U, que va a individualizar el script basándose en el usuario que se está conectando. Es una práctica común personalizar los scripts basándose en el nombre de usuario o del equipo que se está conectando. Estos scripts pueden usarse para configurar parámetros individuales para los usuarios o para los equipos cliente.

Cada script de entrada ha de ser grabado en la raíz del recurso [netlogon]. Por ejemplo, si la raíz del recurso [netlogon] es /export/samba/logon y el script de entrada es jeff.bat, el fichero debe estar grabado en /export/samba/logon/jeff.bat. Cuando un usuario se conecte a un dominio que contenga scripts de entrada, verá una pequeña ventana de dialogo que le informa de que el script se está ejecutando, y también verá en una ventana tipo DOS todas las salidas que genere ese script.

Un aviso: Debido a que estos scripts son cargados por Windows y ejecutados en el lado del cliente Windows, su formato ha de corresponder con caracteres de retorno de carro y nueva línea de DOS y no con los caracteres estándar de retorno de carro de Unix. Es mejor utilizar un editor basado en DOS o Windows para escribirlos.

El siguiente es un ejemplo de un script que establece la hora del equipo cliente para que coincida con la hora del servidor Samba y crea dos unidades de red h e i, que apuntan a distintos recursos compartidos en el servidor:

```
# Reset the current time to that shown by the server.
# We must have the "time server = yes" option in the
# smb.conf for this to work.
  echo Setting Current Time...
  net time \\hydra /set /yes

# Here we map network drives to shares on the Samba
# server echo Mapping Network Drives to Samba Server Hydra...
  net use h: \\hydra\data
  net use i: \\hydra\network
```

6.7.1. Perfiles Itinerantes

En Windows 95 y NT cada usuario puede tener su propio perfil. Un perfil reúne información como: la apariencia del escritorio, las aplicaciones que aparecen en los menús de Inicio, el fondo, y otros elementos similares. Si este perfil se almacena en un disco local, se llama perfil local, porque describe el entorno del usuario en una sólo equipo. Por otro lado, si se almacena en un servidor, el usuario puede descargarlo a cualquier equipo cliente que se conecte a ese servidor. Este último recibe el nombre de perfil itinerante porque el usuario puede ir cambiando de equipo en equipo y seguir usando el mismo perfil. Estos perfiles son muy interesantes cuando un usuario puede estarse conectando desde su despacho un día y desde un portátil al día siguiente. La figura 6-6 ilustra la diferencia entre los perfiles locales e itinerantes.

Samba proporciona perfiles itinerantes si está configurado para accesos por dominio y le proporcionas una serie de directorios establecidos por la opción logon path.

Esta opción se utiliza normalmente combinada con alguna de las variables de usuario, por ejemplo:

```
[global]
  domain logons = yes
  security = user
  workgroup = SIMPLE
  os level = 34
  local master = yes
  preferred master = yes
  domain master = yes
  logon path = \\hydra\profile\%U
```

Necesitamos crear un nuevo recurso para poder utilizar estos perfiles, que será un recurso de disco compartido básico accesible sólo por el usuario del proceso Samba (root). Este recurso ha de ser de escritura, pero no explorable. Además, debemos crear un directorio para cada usuario que desee conectarse (basándonos en lo que hemos establecido en la opción logon path del ejemplo anterior), que será accesible sólo por ese usuario. Para aumentar más la seguridad, usaremos las opciones directory mode y create mode para impedir a cualquiera que se conecte el ver o alterar los ficheros creados en esos directorios.

```
[profile]
  comment = User profiles
  path = /export/samba/profile
  create mode = 0600
  directory mode = 0700
  writable = yes
  browsable = no
```

Cuando un usuario se conecta, el cliente Windows va a crear un fichero user.dat o ntuser.dat (dependiendo del sistema operativo del cliente). Entonces el cliente envía en carpetas individuales el contenido del escritorio, el Menú Inicio, El Entorno de Red, y las carpetas de programa. Las siguientes veces que el usuario se conecte estas carpetas se descargarán del servidor y se activarán en el cliente con el que ese usuario se está conectando. Cuando se desconecte, estas carpetas se enviarán de nuevo al servidor hasta la próxima vez que el usuario se conecte. Si miras el directorio de una carpeta de perfil, verás lo siguiente:

```
# ls -al total 321 drwxrwxr-x 9 root simple Jul 21 20:44 .
drwxrwxr-x 4 root simple Jul 22 14:32 ..
drwxrwx--- 3 fred develope Jul 12 07:15 Application Data
drwxrwx--- 3 fred develope Jul 12 07:15 Start Menú
drwxrwx--- 2 fred develope Jul 12 07:15 cookies
drwxrwx--- 2 fred develope Jul 12 07:15 desktop
drwxrwx--- 7 fred develope Jul 12 07:15 history
drwxrwx--- 2 fred develope Jul 12 07:15 nethood
drwxrwx--- 2 fred develope Jul 19 21:05 recent
-rw------- 1 fred develope Jul 21 21:59 user.dat
```

Los ficheros user.dat son ficheros binarios de configuración, creados automáticamente por Windows. Pueden ser editados con el Editor de Perfiles en un cliente Windows, pero puede ser complicado. Samba los soporta correctamente para todos los clientes incluido NT 5.0 beta, pero aún son relativamente nuevos.

> **AVISO**: Pistas y Cómos para manejar los scripts de entrada están disponibles con la configuración de Samba, en docs/textdocs/DOMAIN.txt y docs/textdocs/PROFILES.txt.

6.7.2. Perfiles Obligatorios

Los usuarios también pueden tener perfiles obligatorios que son aquellos perfiles itinerantes que no pueden cambiar. Por ejemplo, con un perfil obligatorio, si el usuario añade una orden al Menú de Inicio el Martes, desaparecerá cuando se vuelva a conectar el Miércoles. El perfil obligatorio simplemente es un fichero user.dat que se ha renombrado a user.man y hecho de solo-lectura dentro del servidor Unix. Normalmente contiene instrucciones que el administrador quiere que se ejecuten obligatoriamente. Por ejemplo, si el administrador quiere crear una configuración de usuario fija puede hacer lo siguiente:

- Crear un directorio de solo lectura dentro del servidor Samba.
- Establecer la opción logon path en el fichero smb.conf para que apunte a ese directorio.
- Conectarse como el usuario desde un cliente Windows 95/98 para que el cliente rellene el directorio.
- Renombrar el fichero user.dat resultante a user.man.
- Hacer el directorio y sus contenidos de solo lectura.

Estos perfiles no se usan con frecuencia. Por otro lado, los perfiles itinerantes son una de las prestaciones de Windows más deseables de cara a que Samba la soporte.

6.7.3. Opciones de los scripts de entrada

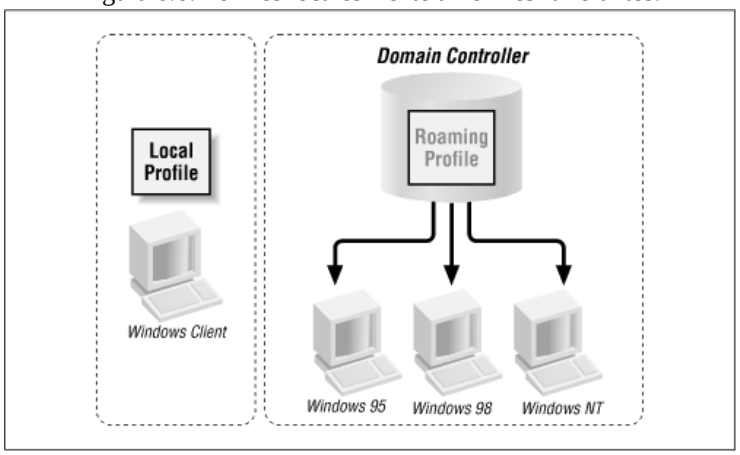
La tabla 6-10 resume las opciones más usadas con relación a los scripts de entrada para Windows:

6.7.3.1. logon script

Esta opción especifica un fichero Windows .BAT o .CMD con las líneas finalizadas por caracteres de retorno de carro/nueva línea que se ejecutara en el cliente una vez que el usuario se haya conectado al dominio. Cada script de entrada debe ser grabado en la raíz de un recurso de disco llamado [netlogon] (consulta la sección llamada *Configurando Samba para Accesos por Dominios Windows*). Esta opción a menudo usa las variables %U o %m (nombre de usuario o NetBIOS) para apuntar a un script individual. Por ejemplo:

```
logon script = %U.bat
```

Figura 6.6: Perfiles locales frente a Perfiles Itinerantes.



Cuadro 6.12: Opciones en los scripts de entrada.

Opción	Parámetros	Función	Defecto	Ámbito
logon script	Cadena (Ruta DOS)	Nombre del fichero ejecutable DOS/NT	Ninguno	Global
logon path	Cadena (Servidor UNC y ruta)	Situacion del perfil itinerante del usuario	\\%N\%U\perfil	Global
logon drive	Cadena (Letra de unidad)	Especifica la unidad de entrada para un directorio personal (Solo NT)	Z:	Global
logon home	Cadena (Servidor UNC y ruta)	Especifica la situacion de los directorios personales para los usuarios que se conecten al dominio	\\%N\%U	Global

ejecutará un script basado en el nombre del usuario y que está almacenado en la raíz del recurso [netlogon]. Si el usuario que se conecta es fred y la ruta del recurso [netlogon] apunta al directorio */export/samba/netlogon*, el script deberá ser */export/samba/netlogon/fred.bat*. Debido a que estos scripts son cargados por Windows y ejecutados en el lado del cliente Windows, su formato ha de corresponder con caracteres de retorno de carro y nueva línea de DOS y no con los caracteres estándar de retorno de carro de Unix.

6.7.3.2. logon path

Esta opción proporciona la situación de los perfiles itinerantes. Cuando el usuario se conecta, un perfil itinerante se descargará del servidor al cliente y se activará cuando el usuario se conecte. Cuando se desconecte, estos elementos se enviarán de nuevo al servidor hasta la próxima vez que el usuario se conecte.

Es mucho más seguro crear un recurso separado para almacenar los perfiles de usuario:

```
logon path = \\hydra\profile\%U
```

Para mas información, consulta la sección Scripts de Entrada en este mismo capitulo.

6.7.3.3. logon drive

Esta opción establece la letra de unidad en un cliente NT que se asignara al directorio de usuario que se establece en la opción logon home. Fíjate en que esta opción sólo funcionará con clientes Windows NT. Por ejemplo:

```
logon home = I:
```

Siempre deberás usar letras de unidad que no interfieran con unidades de disco fijo presentes en el equipo cliente. El valor por defecto es Z: , que es una buena elección porque se encuentra lo mas alejada posible de A:, C: y D:

6.7.3.4. logon home

Esta opción establece la ubicación del directorio personal de un usuario que utilizará el comando DOS NET. Por ejemplo, si quieres establecer el directorio personal como un recurso en el servidor Samba, usaras:

```
logon home = \\hydra\%U
```

Fíjate en que esto funciona bien con el servicio [homes], así que puedes establecer cualquier directorio que desees. Los directorios personales pueden ser mapeados a una unidad con un script de entrada usando el siguiente comando:

```
NET USE I: /HOME
```

Además, puedes usar, dentro del Administrador de usuarios de NT, la opción Perfiles de Usuario que está dentro de las propiedades del usuario para comprobar que el directorio personal se ha establecido automáticamente.

6.7.4. Otros scripts de conexión

Desde que un usuario se conecta con éxito a un recurso Samba, puedes querer que el servidor Samba ejecute un programa para preparar el recurso para su uso. Samba permite scripts que se ejecutaran antes y después de que alguien se conecte al recurso. No necesitas usar dominios Windows para utilizar esta opción. La Tabla 6-11 te presenta alguna de las opciones de configuración proporcionadas para configurar usuarios.

Cuadro 6.13: Opciones de configuración de usuarios.

Opción	Parámetros	Función	Valor Defecto	Ámbito
root preexec	Cadena (Comando Unix)	Establece una orden a ejecutar, como root, antes de la conexion al recurso	Ninguno	Recurso
preexec (exec)	Cadena (Comando Unix)	Establece una orden a ejecutar, como usuario, antes de la conexion al recurso	Ninguno	Recurso
postexec	Cadena (Comando Unix)	Establece una orden a ejecutar, como usuario, despues de la desconexion al recurso	Ninguno	Recurso
root postexec	Cadena (Comando Unix)	Establece una orden a ejecutar, como root, despues de la desconexion al recurso	Ninguno	Recurso

6.7.4.1. 6.6.4.1. root preexec

Esta opción establece un comando Unix que se ejecutará como el usuario root antes de que se complete cualquier conexión al recurso. Has de usar esta opción específicamente para realizar acciones que requieran privilegios de root. Por ejemplo, root preexec puede ser utilizada para montar CD-ROMs para un recurso que lo hagan disponible para los clientes, o para crear directorios. Si no se establece la opción root preexec no hay acción por defecto. Vamos a ver un ejemplo de cómo puedes usar el comando para montar un CD-ROM.

```
[homes]
  browseable = no
  writeable = yes
  root preexec = /etc/mount /dev/cdrom2
```

Recuerda que estos comandos se ejecutarán como usuario *root*. Por tanto, para garantizar la seguridad, los usuarios nunca deberán ser capaces de modificar el destino del comando root preexec.

6.7.4.2. 6.6.4.2. preexec

La próxima opción que se ejecuta antes de la entrada es preexec, algunas veces llamada simplemente exec. Es una orden que no necesita privilegios y que se ejecuta como el usuario especificado por la variable %u. Por ejemplo, un uso común de esta opción es llevar a cabo una entrada, por ejemplo:

```
[homes]
  preexec = echo "%u connected to %S from %m (%I)\" >>/tmp/.log
```

Ten en cuenta que cualquier información que el comando envíe a la salida estándar no será vista por el usuario, sino que será descartada. Si pretendes usar un script preexec asegúrate de que funciona perfectamente antes de hacer que Samba lo ejecute.

6.7.4.3. 6.6.4.3. postexec

Una vez que el usuario se desconecta del recurso, se ejecuta el comando especificado en la opción postexec con los privilegios del usuario. Esta opción es esencialmente la misma que preexec. De nuevo, recordar que el comando se ejecuta como si lo hiciera el usuario representado por %u y cualquier información enviada a la salida estándar será ignorada.

6.7.4.4. root postexec

Después de ejecutar la opción postexec, se ejecuta root postexec, en caso de haber sido especificado. De nuevo, esta opción establece un comando Unix que se ejecutará como si lo hiciera el usuario root antes de desconectarse de un recurso. Debes usar esta opción únicamente para realizar operaciones que requieran privilegios de root.

6.7.5. Trabajando con NIS y NFS

Finalmente, Samba tiene la posibilidad de trabajar con NIS y NIS+. Si hay más de un servidor de ficheros, y cada uno de ellos ejecuta Samba, puede ser una buena idea que el usuario esté conectado al servidor que actualmente tiene los discos en los que está grabado su directorio personal. Normalmente no es buena idea enviar los ficheros a través de la red vía NFS al servidor Samba, para ser enviados de nuevo a través de la red hacia el cliente vía SMB. (por una razón: es lento - cerca de un 30 % sobre la velocidad normal de Samba). Aun así, hay un par de opciones para decirle a Samba que NIS sabe cual es el servidor correcto y le indica en que mapa NIS está la información.

La tabla 6-12 presenta otras opciones de configuración especificas para configurar usuarios:

Cuadro 6.14: Opciones de configuración de usuarios.

Opción	Parámetros	Función	Defecto	Ámbito
nis homedir	Lógico	Si su valor es yes, utiliza NIS en vez de /etc/passwd para buscar la ruta del directorio personal del usuario	no	Global
homedir map	Cadena (nombre del mapa NIS)		Ninguno	Global

6.7.5.1. nis homedir y nis homedir map

Estas opciones se utilizan con servidores Samba en redes donde los directorios Unix se utilizan a través de NFS, el automontador y NIS (las páginas amarillas).

La opción nis homedir especifica que se ha de buscar el directorio personal para el usuario a través de NIS. La opción homedir map le indica a Samba que mapa NIS

mirar para buscar el servidor que contiene el directorio del usuario. Este servidor ha de ser un servidor Samba, para que el cliente pueda hacer una conexión a el, y los otros servidores Samba han de tener NIS instalado para que puedan hacer la búsqueda.

Por ejemplo, si el usuario joe pregunta por un recurso llamado [joe], y la opción nis homedir esta marcada como yes, Samba buscará en el fichero especificado por homedir map un directorio llamado joe. Si lo encuentra, Samba devolverá el nombre del equipo asociado al cliente. Entonces el cliente intentará conectarse a esa máquina y recuperar el recurso desde ella. Para activar las búsquedas NIS hacemos lo siguiente:

```
[globals]
  nis homedir = yes
  homedir map = amd.map
```

Capítulo 7

Impresión y Resolución de Nombres

Este capítulo se encarga de dos características de Samba: configurar impresoras para usar junto con un servidor Samba y configurar Samba para usar o convertirse en un servidor de "Servicio de Nombres de Internet Windows"[1] (WINS). Samba permite a las máquinas clientes enviar documentos a impresoras conectadas al servidor Samba. Además samba te puede ayudar a imprimir documentos UNIX en una maquina Windows. En la primera parte de este capitulo hablaremos de como configurar impresoras para que funcionen en ambos sentidos.

En la segunda parte del capitulo, hablaremos del "Servicio de Nombres de Internet Windows", una implementación de Microsoft del Servidor de Nombres NetBIOS (NBNS). Como se mencionaba en el capítulo 1, un NBNS permite a las maquinas resolver nombres NetBIOS evintando los mensajes de difusión en la red. En lugar de eso cada máquina sabe donde se encuentra el servidor WINS y le pregunta a el las direcciones IP del resto de máquinas de la red.

7.1. Enviando tareas de impresión a SAMBA

Una impresora conectada a un servidor Samba se ve en la lista de recursos compartidos del Entorno de Red. Si la impresora está capturada en la máquina cliente y esta dispone del controlador de impresión adecuado, el cliente puede mandar perfectamente tareas de impresión a la impresora conecteda al servidor Samba. La Figura 7.1 muestra como aparece una impresora Samba en el Entorno de Red del cliente Windows.

Para administrar impresoras con Samba, deberias entender el proceso basico mediante el cual se produce una impresión a través de una red. Enviar una taréa de impresión a un servidor Samba implica cuatro pasos:

1. Abrir y autenticar una conexión a la impresora.
2. Copiar el fichero a través de la red.
3. Cerrar la conexión.
4. Imprimir y borrar la copia del fichero.

Cuando una taréa de impresión llega al servidor Samba, los datos a impremir son guardados en disco en el directorio especificado por la opción "path" del recurso impresora. Entonces Samba ejecuta un comando de impresión UNIX para enviar los datos a la impresora. El trabajo es impreso como el usuario atenticado del recurso. Fijate que podria ser un usuario invitado dependiendo de la configuración del recurso.

7.1.1. Comandos de Impresión

Para poder imprimir, necesitas decirle al servidor Samba cual es el comando para imprimir y borrar un fichero. En Linux, este comando es:

```
lpr`r`P
impresora
archivo
```

Esto le dice a que copie el documento en el directorio de la cola, habitualmente */var/spool/*, busque el nombre de la impresora en el fichero de configuración del sistema (*/etc/printcap*), e interprete las reglas que encuentre en el para decidir como procesar los datos y a que dispositivo físico enviarlo. Al usar la opción *-r*, el fichero se elimina despues de ser impreso. Por supuesto, el fichero eliminado en sólo una copia en el servidor Samba, el fichero original situado en el cliente no se ve afectado.

Linux usa el estilo de impresión Berkeley (BSD). De todas formas, el proceso es similar en UNIX System V, En estos sistemas, imprimir y borrar se convierte en un proceso compuesto:

```
lp`d
impresora`s
fichero; rm
fichero
```

Con System V, el fichero */etc/printcap* es reemplazado con diferentes juegos de ficheros de configuración colgando de */usr/spool/lp*, y no hay opción para borrar el fichero. Esto debe ser realizado manualmante, este es el montivo por el que se incluye *rm* al final.

7.1.2. Variables de Impresión.

Samba proporciona cuatro variables especificas para usar en las opciociones de configuración del la impresión, estas variables se muestran en la Tabla 7.1.

Cuadro 7.1: Variables de Impresión.

Variable	Definición
%s	La ruta completa del archivo a ser impreso en el servidor Samba.
%f	El nombre del archivo (sin ruta) a serr impreso en el servidor Samba.
%p	Nombre de la impresora UNIX a usar.
%j	El número de la taréa de impresión. (Para usar con lprm, lppause, y lpresume)

[1] Windows Internet Name Server. (n. del t.)

7.1.3. Una configuración de impresión Mínima.

Comencemos con un simple pero ilustrativo recurso de impresión. suponiendo que estás en un sistema Linux y que tienes una impresora llamada *lp* listada en el archivo de configuración de impresión, añadir las siguientes lineas a tu fichero *smb.conf* hará a la impresora accesible a través de la red.

```
[printer1]
  printable = yes
  print command = /usr/bin/lpr -r  %s
  printer = lp
  printing = BSD
  read only = yes
  guest ok = yes
```

Esta configuración le permite a cualquiera enviar datos a la impresora, algo que querremos cambiar mas adelante. De momento, lo que es importante entender es que la variable *%s* en el comando de impresión será sustituida con el nombre del archivo a ser impreso cuando Samba ejecute el comando. Cambiar el comando de impresión para usar en un tipo distinto de máquina UNIX habitualmente sólo implica cambiar la parte que está a la derecha de la variable con cualquier comando que necesites en tu sistema y cambiar el parámetro de la opción *printing*.

Por ejemplo, para Unix System V, la opción *print command*, se transforma en:

```
print command = lp -d%p -s %s; rm %s
```

Como se mencionó antes, la variable *%p* contiene el nombre de la impresora, mientras que *%s* contiene el nombre del fichero. después de esto, puedes cambiar el parámetro de la opción *printing* así:

```
printing = SYSV
```

Si estás usando seguridad a nivel de recursos, presta una atención especial a la cuenta de invitado usada por Samba. La opción habitual a nobody, pede no poder imprimir dependiendo del sistema operativo. Si eso pasara con tu sistema, deberias poner una opción de cuenta de invitado en el recurso de la impresora (o quizá en los recursos globales) especificando una cuenta que sí pueda imprimir. Una opción muy popular entre los autores de Samba es la cuenta de ftp, que habitualmente está preconfigurada para ofrecer seguridad para usuarios invitados en los que no se puede confiar. Puedes establecerlo con el comando siguiente:

```
guest account = ftp
```

Otro detalle a tener en cuanta, es que los clientes pueden querer saber el estado de de una taréa de impresión enviada al servidor Samba. Samba no rechazará un documento que se envia a una impresora que se encuentre ocupada. En consecuencia, Samba no sólo necesita cominicar el estado de una taréa de impresión al cliente, sino que también necesita saber que docomentos se encuentranen la cola de cada impresora. samba también proporcia al cliente la posibilidad de pausar trabajos, continuarlos o borrarlos de la cola de impresión. Samba proporciona opciones para cada una de esas taréas, como es de esperar, se aprovecha del funcionamiento de otros comandos Unix, estos son:

- lpq
- lprm
- lppause
- lpresume

Trataremos esas opciones con mas detenimiento mas adelante. de todas formas, para la mayor parte de ellas, los valores de la configuración de impresión determinará sus valores, y no necesitarias cabiar los valores por defecto de esas opciones.

Aqui tenemos una serie de copceptos básicos a recordar acerca de compartir impresoras:

- Debes poner *printable = yes* en todos los recursos de impresora (y/o en [printers]), de forma que Samba pueda saber que se trata de impresoras. Si se olvida, estos recurosos no podrán ser usados para imprimir y apareceran como recursos de disco.
- Si estableciste un valor en la opción de configuración *path*, cualquier fichero que se envíe a la(s) impresora(s) será copiado a ese directorio en lugar del sitio por defecto *tmp*. Como el espacio asignado a *tmp* en algunos sistemas puede ser relativamente pequeño, muchos administradores optan por usar */var/spool* o algún otro directorio en su lugar.
- La opción de sólo lectura es ignorada en las impresoras.
- Si pones *guest ok = yes* en un recurso de impresora y Samba está configurado para seguridad a nivel de recursos, permitirá a cualquiera enviar datos a la impresora como usuario invitado.

Usar una o varias máquinas con Samba como servidores de impresión, te da una gran flexibilidad en tu LAN. Puedes facilmente particionar tus impresoras disponibles, restringir alguna a miembros de un departamento, o puedes mantener una serie de impresoras disponibles para todos. Además, puedes restringir una impresora a un grupo de usuarios seleccionados añadiendo la opción de comprobar usuarios válidos en la configuración del recurso:

```
[deskjet]
  printable = yes
  path = /var/spool/samba/print
  valid users = gail sam
```

Todas las demás opciones de acceso a recursos definidas en el capítulo anterior deberían funcionar para impresoras también. Como Samba accede a las impresoras en sí mismas usando su nombre, es tambien simple distribuir los servicios de impresión entre varios servidores usando comandos Unix para tareas como el **balance de carga** y el mantenimiento.

7.1.4. El recurso [Printers].

El capítulo 4, Compartición de unidades de disco, se habla por encima del recurso [printers], un recurso especial para crear servicios de impresión. Miremos como funciona: si creas un recurso llamado [printers] en el fichero de configuración, samba automáticamente lee el fichero de capacidades de impresora y crea un recurso para cada impresora que aparece en él. Por ejmplo, si el servidor samba tiene tres impresoras, lp, pcl y ps, en su fichero de capacidades de impresora, samba creará tres recursos de impresión con esos nombres, cada uno de ellos configurado con las opciones del recurso [printers].

Ten en cuenta que samba sigue los siguientes pasos cuando un cliente pide un recurso que no ha sido creado en el fichero smb.conf:

- Si el nombre del recurso coindice con el nombre de un usuario en el fichero de passwords del sistema y esiste un recurso [home], se crea un nuevo recurso con el bnombre del usuario y es inicializado usando los valores de las secciones [home] y [global].
- Si no, si el nombre del recursos coincide con el de una impresora en el fichero de capacidades de impresora, y existe el recurso [printers]. se crea un nuevo recurso con el nombre de la impresora y los valores de la sección [printers]. (Las variables de la sección global, no se usan aqui).
- Si no se ha producido ninguna de las anteriores, Samba busca si se ha definido un recurso por defecto, si no es así, devuelve un error.

Esto nos ilumina un punto importante, nhay que tener cuidado de no nombrar a una impresora como a un usuario ya que si lo hacemos terminaremos conectando un recurso de disco en lugar de una impresora como queriamos.

Aqui tenemos un ejemplo del recurso [printers] para un sistema Linux (BSD). Alguna de esas opciones ya se asignan por defecto; de todas formas las hemos incluido con propositos ilustrativos.

```
[global]
  printing = BSD
  print command = /usr/bin/lpr -P %p -r %s
  printcap file = /etc/printcap
  min print space = 2000

[printers]
  path = /usr/spool/public
  printable = true
  guest ok = true
  guest account = pcguest
```

Aqui le hemos pasado a samba unas opciones globales que le indican el tipo de sistema (BSD), la orden de impresión para enviar ingormación a la impresora y eliminar el fichero temporal, nuestro fichero de capacidades de impresora por defecto y un espacio minimo de impresión de 2 Mb.

Además, hemos creado un recurso [printers] para cada una de nuestras impresoras. Nuestro directorio de colas de impresión será */usr/spool/public*. Cada uno de los recursos es marcado como imprimible - eso es necesario incuso en la sección [printers]. Las

dos opciones de cuentas de invitado, son utiles en el caso en que estemos la seguridad a nivel de recursos: permitimos a invitados usar las impresoras, e indicamos que usuario ejecutará las ordenes de impresión.

7.1.5. Probando la Impresión.

Ahora veremos como puedes probar la impresión desde tu servidor Samba, consideremos el caso mas complejo y usando una cuenta de invitado. Primero, se ejecuta la orden *testparm* en el fichero de configuración que contiene los recursos de impresión. Como hicimos en el capitulo 2, Instalando Samba en un Sistema Unix, Esto te dirá si hay problemas de sintaxis en el fichero de configuración. Por ejemplo, esto es lo que verias si hubiesemos quitado la opción *path* del ejemplo anterior:

```
# testparm
Load smb config files from /usr/local/samba/lib/smb.conf
Processing configuration file "/usr/local/samba/lib/smb.conf"
Processing section "[global]"
Processing section "[homes]"
Processing section "[data]"
Processing section "[printers]"
No path in service printers - using /tmp
Loaded services file OK.
Press enter to see a dump of your service definitions

Global parameters:
  load printers: Yes
  printcap name: /etc/printcap

Default service parameters:
  guest account: ftp
  min print space: 0
  print command: lpr -r -P%p %s
  lpq command: lpq -P%p
  lprm command: lprm -P%p %j
  lppause command:
  lpresume command:

Service parameters [printers]:
  path: /tmp
  print ok: Yes
  read only: true
  public: true
```

Segundo, prueba la orden *testprns printername*. Este orden comprueba que *printername* se encuentra disponible en tu fichero *printcap*. Si el fichero *printcap* no está en el sitio habitual, puedes especificar su ruta como segundo parámetro de *testprns*:

```
# testprns lp /etc/printcap
Looking for printer lp in printcap file /etc/printcap
Printer name lp is valid.
```

Después, inicia sesión como usuario invitado, vete al directorio donde se almacenan las colas de impresión, y asegurate de que puedes imprimir usuando la misma orden que *testprns* dice que Samba usará. Como se comentó antes, esto te dirá si necesitas cambiar la cuenta de invitado, como la cuenta por defecto, puede no tener permisos para imprimir.

Para terminar, imprime algo a través de *smbclient* y comprueba que ocurra lo siguiente:

- El trabajo aparece, (momentaneamente) en el directorio de las colas de impresión especificado por samba.
- El trabajo aparece en el directorio de colas de impresión del sistema.
- El trabajo desaparece del directorio de colas de Samba.

Si {\tt smbclient} no es capaz de imprimir, puedes redefinir la orden de impresión para recojer información de depurado:

```
print command = /bin/cat %s >>/tmp/printlog; rm %s
```

o

```
print command = echo "printed %s on %p" >>/tmp/printlog
```

Un problema común con la configuración de impresoras con samba es olvidar usar las rutas completas a las ordenes a usar; muchas veces una orden suelta no funciona porque el PATH del usuario invitado no la incluye, otro problema común es no tener los permisos adecuados en los directorios de colas de impresión.

Hay mas información acerca del depurado en la documentación de Samba *Printers.txt*. Además los sistemas de impresión bajo Unix son tratados en detalle en el libro de AEleen Frisch, Essential Systems Administration, (publicado por O'Reilly).

7.1.6. Configurando y Probando un Cliente Windows.

Ahora que Samba está ofreciendo una impresora a los usuarios de la red, necesitamos configurar su uso en un cliente Windows. Busca el servidor Samba en el Entorno de Red. Este debería ahora mostrar cada una de las impresoras disponibles. Por ejemplo, en la Figura 7.1., vimos una impresora llamada *lp*.

A continuación, necesitas que el cliente Windows reconozca la impresora. Haz doble click sobre el icono de la impresora. Si intentas seleccionar una impresora que no está instalada, Windows te preguntará si debería ayudarte a configurarla para añadirla al sistema Windows. Responde 'Sí', lo cual abrirá el Asistente de Configuración de Impresoras.

Lo primero que te va a preguntar el asistente es si necesitarás imprimir desde programas basados en DOS. Asumiendo que no va a sernos necesario, selecciona 'No' y pulsa sobre el botón 'Siguiente' para introducir el modelo de impresora tal y como se muestra en la Figura 7.2.

En este cuadro de diálogo, deberías ver una larga lista de fabricantes y modelos para casi todas las impresoras imaginables. Si no ves tu impresora en la lista, pero sabes que es una impresora PostScript, selecciona **Apple** como fabricante, y **Apple LaserWriter** como modelo. Esto te dará una configuración básica de impresora Postscript, y

lo más seguro es que te funcione sin problemas. Si ya tienes impresoras Postscript instaladas, se te preguntará si deseas reemplazar o reutilizar el driver existente. Cuidado si reemplazas el existente con otro nuevo, puede que esas otras impresoras te dejen de funcionar. De cualquier forma, recomendamos que mantengas esos drivers existentes y pruebes si con ellos te funcionaría.

A continuación, El Asistente te preguntará por un nombre para la impresora. La Figura 7.3 te muestra el caso, en el que el nombre seleccionado será el que se ponga por defecto para nuestra segunda impresora. Cambia el nombre de 'Apple Laserwriter (Copia 2)' a 'ps en Samba server'. En realidad, puedes usar el nombre que quieras.

Finalmente, el Asistente te preguntará si debería hacer una impresión de página de pruebas. Selecciona 'Sí', y te deberá aparecer el diálogo que muestra la Figura 7.4.

Si la impresión de pruebas no tuvo éxito, pulsa el botón 'No' de la Figura 7.4. y el Asistente te mostrará algunos pasos a seguir para la comprobación de errores, desde la parte del cliente. Si el test de impresión te funcionó ¡Enhorabuena! La impresora remota estará ahora disponible para todas tus aplicaciones a través del menú de impresión.

7.1.7. Configurando Automáticamente Drivers de Impresión.

La sección anterior describió cómo configurar manualmente un driver de impresión para tu sistema Windows. Como administrador del sistema, sin embargo, no siempre puedes tener la garantía de que los usuarios serán capaces de realizar el proceso de la configuración de la impresora sin cometer errores. Por suerte, puedes indicarle a Samba que automáticamente configure los drivers de impresión para una determinada impresora.

Samba tiene tres opciones que puedes usar para configurar automáticamente drivers de impresión para los clientes que se conectan por primera vez al sistema. Estas opciones son *printer driver*, *printer driver file*, y *printer driver location*. Esta sección explica cómo usar estas tres opciones para permitir a los usuarios evitar la necesidad de usar el Asistente de Configuración de Impresoras.

Para más información sobre cómo hacer esto, mira el fichero PRINTER_DRIVER.TXT en la documentación de la distribución de Samba.

Hay cuatro pasos principales:

1. Instalar los drivers para la impresora sobre un cliente Windows (la impresora no debe estar conectada).
2. Crear un fichero de definición de impresora.
3. Crear un recurso PRINTER$ donde los ficheros conteniendo los drivers puedan ser ubicados.
4. Modificar el fichero de configuración de Samba.

¡Vamos allá con cada uno de estos pasos!

7.1.7.1. Instalando los Drivers sobre un Cliente Windows.

Usa Windows 95/98 para este paso. No importa el cliente que elijas, lo que importa es que sea capaz de cargar los drivers aproppiados para la impresora. De hecho, no necesitas tener la impresora conectada al equipo. Todo lo que te interesa aquí es obtener el/los ficheros con el driver adecuado e instalarlos en el directorio Windows. Primero,

Figura 7.1: Una Impresora Samba en el Entorno de Red.

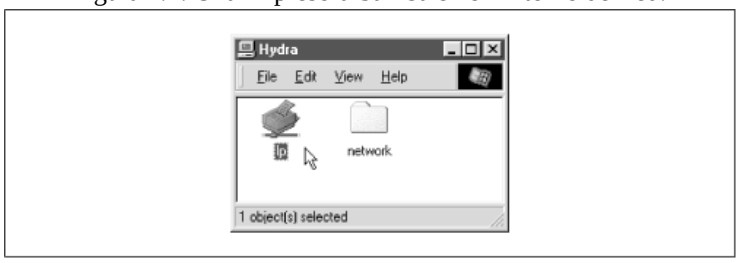


Figura 7.2: Una Impresora en el Entorno de Red.

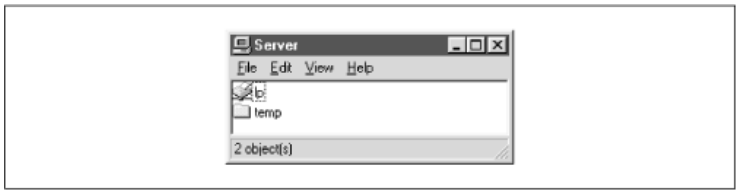


Figura 7.3: Fabricantes y Modelos de Impresoras.

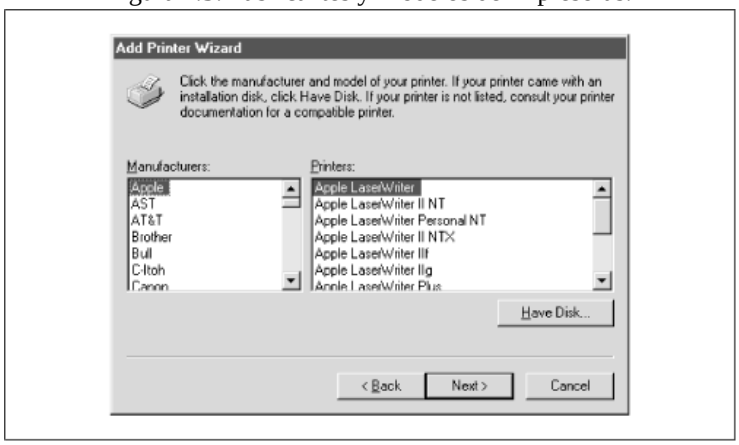


Figura 7.4: Impresión de Pruebas Completada.

vete a la ventana de impresoras de *My Computer* y haz doble click sobre el icono 'Añadir Impresora', como se muestra en la Figura 7.5.

En este punto, puedes seguir los pasos típicos del Asistente para localizar el fabricante y modelo de la impresora. Cuando te pregunte si vas a imprimir en DOS, responde 'No'. Windows debería cargar los recursos adecuados para el driver en cuestión desde el CD de instalación y a continuación preguntarte por la impresión de una página pruebas. De nuevo, responde 'No' y cierra el Asistente.

7.1.7.2. Crear un Fichero de Definición de Impresora.

Puedes crear un fichero de definición de impresora usando el script *make_ printerdef* que tienes en el directorio */usr/local/samba/bin*. Para usarlo, necesitas copiar los siguientes cuatro ficheros desde un cliente Windows[2]:

```
C:\WINDOWS\INF\MSPRINT.INF
C:\WINDOWS\INF\MSPRINT2.INF
C:\WINDOWS\INF\MSPRINT3.INF
C:\WINDOWS\INF\MSPRINT4.INF
```

Una vez tengas los cuatro ficheros puedes crear un fichero de definición de impresora usando el driver y su fichero .INF apropiados. Si el driver comienza con las letras A-K, usa en su lugar el fichero MSPRINT.INF o el fichero MSPRINT3.INF. Si comienza por las letras L-Z, usa MSPRINT2.INF o MSPRINT4.INF. Puedes necesitar abrir los ficheros para ver cuál es el específico de tu driver. Para el siguiente ejemplo, hemos localizado nuestro driver en MSPRINT3.INF y creado un fichero de definición de impresora para una HP DeskJet 560C:

```
$grep "HP DeskJet 560C Printer" MSPRINT.INF
MSPRINT3.INF MSPRINT3.INF: "HP DeskJet 560C Printer"=DESKJETC.DRV,HP_DeskJet_ ...
$make_printerdef MSPRINT3.INF "HP DeskJet 560C Printer" > printers.def
FOUND:DESKJETC.DRV
End of section found
CopyFiles: DESKJETC,COLOR_DESKJETC
Datasection: (null)
Datafile: DESKJETC.DRV
Driverfile: DESKJETC.DRV
Helpfile: HPVDJC.HLP
LanguageMonitor: (null)
```

Copia los siguientes ficheros a la ubicación de tu recurso printer$:

```
DESKJETC.DRV
HPVCM.HPM
HPVIOL.DLL
HPVMON.DLL
HPVRES.DLL
HPCOLOR.DLL
HPVUI.DLL
HPVDJCC.HLP
color\HPDESK.ICM
```

Recuerda los ficheros que el script te ha solicitado que copies. Los necesitarás para el siguiente paso.

[2] Los viejos clientes Windows 95 pueden tener sólo los dos primeros ficheros.

7.1.7.3. Creando el Recurso PRINTER$.

Esta parte es relativamente sencilla. Crea un recurso llamado [PRINTER$] en tu *smb.conf* que apunte a un directorio vacío en el servidor Samba. Una vez hecho, copia allí los ficheros que el script *make_ printerdef* te indicó que copiaras para el recurso [PRINTER$]. Por ejemplo, puedes poner lo siguiente en tu fichero de configuración:

```
[PRINTER$]
  path = /usr/local/samba/print
  read only = yes
  browsable = no
  guest ok = yes
```

Los ficheros solicitados por el script *make_ printerdef* normalmente se encuentran en el directorio C:\WINDOWS\SYSTEM, aunque puedes usar los siguientes comandos par saber exactamente dónde están:

```
cd C:\WINDOWS
dir

filename /s
```

En este caso, *filename* es cada uno de los ficheros que necesitaremos copiar en el directorio */usr/local/samba/print* del servidor Samba. Además, copia también el fichero *printers.def* que creaste sobre el recurso. Una vez lo hayas hecho todo, podremos seguir.

7.1.7.4. Modificando el Fichero de Configuración de Samba.

El último paso es modificar el fichero de configuración de Samba añadiendo las siguientes tres opciones:

- printer driver
- printer driver file
- printer driver location

La opción *printer driver file* es una opción global que apunta al fichero *printers.def*; coloca esta opción en tu sección *[global]*. Las otras opciones deberían ponerse en la sección del recurso de impresión para el cual vamos a crear la configuración automática de drivers. El valor para *printer driver* debería coincidir con la cadena que se mostraba eb el Asistente de Impresoras del cliente Windows. El valor para *printer driver location* es la ruta completa del recurso PRINTER$, no la ruta Unix en el servidor. Dicho esto, podrías usar la siguiente configuración:

```
[global]
  printer driver file = /usr/local/samba/print/printers.def [hpdeskjet]
  path = /var/spool/samba/printers
  printable = yes
  printer driver = HP DeskJet 560C Printer
  printer driver location = \\%L\PRINTER$
```

Ahora ya estamos preparados para ponerlo a prueba. Llegados aquí, desinstala la impresora Windows que configuraste en el primer paso de la lista de impresoras de ese equipo. Si Windows te pregunta si deseas eliminar los ficheros innecesarios, responde 'Sí'. Estos ficheros serán reemplazados en el cliente, ya que existen en servidor Samba.

7.1.7.5. Testeando la Configuración.

Reinicia los demonios Sambay busca el recurso [hpdeskjet] en el Entorno de Red. Si haces click sobre el icono de la impresora, deberías haber iniciado el proceso de configuración y llegar al cuadro de diálogo que muestra la Figura 7.6.

Como ves es diferente al que vimos antes al configurar la impresora en Windows. Esencialmente, el cuadro de diálogo te pregunta si aceptas el driver que 'está a punto de instalarse'. Continúa manteniendo el driver existente, y pulsa el botón 'Siguiente'. En este punto, puedes darle un nombre a la impresora e imprimir una página de pruebas. Si funciona, la configuración está completada. Deberías poder repetir el proceso desde cualquier cliente Windows.

7.2. Impresión sobre Impresoras de Cliente Windows.

Si tienes impresoras conectadas a clientes Windows 95/98 o NT 4.0, esas impresoras pueden ser también accedidas desde Samba. Samba viene equipado con una herramienta llamada *smbprint* que puede ser usada para enviar trabajos de impresión a impresoras Windows. Para poder usar esto, sin embargo, necesitas configurar la impresora como recurso compartido en la máquina cliente. Si no la tienes compartida, hazlo desde la Ventana de Impresoras, como ves en la Figura 7.7.

Selecciona una impresora que esté localmente conectada (por ejemplo, la nuestra en la impresora Canon), presiona el botón derecho del ratón, y selecciona 'Compartir'. Esto te llevará a la ventana de compartición de recursos de las Propiedades de la Impresora, como muestra la Figura 7.8. Si quieres hacerla disponible para todo el mundo en tu red, como la impresora por defecto para usuarios anónimos, introduce una contraseña en blanco.

Una vez hecho esto, puedes añadir tu impresora a la lista de impresoras por defecto y Samba puede hacerla disponible al resto de equipos de la red. Para hacer la instalación en Unix más sencilla, la distribución Samba proporciona dos scripts de ejemplo: *smbprint* y *smbprint.sysv*. El primero funciona con impresoras tipo BSD; el segundo es el utilizado para impresoras System V.

7.2.1. Impresoras BSD.

Hay dos pasos necesarios para que un sistema Unix BSD reconozca una impresora remota:

1. Colocar una entrada para la impresora en el fichero */etc/printcap* (o equivalente).
2. Colocar un fichero configuración en el directorio */var/spool* para esa impresora.

Primero, edita tu fichero */etc/printcap* y añade una entrada para la impresora remota. Advierte que el filtro de entrada (if) necesita apuntar al programa *smbprint* si la máquina está sobre Windows 95/98. El siguiente conjunto de líneas iría en una máquina Linux machine:

```
laserjet:\
:sd=/var/spool/lpd/laser:\

# spool directory
:mx#0:\
```

Figura 7.5: La Ventana de Impresoras.

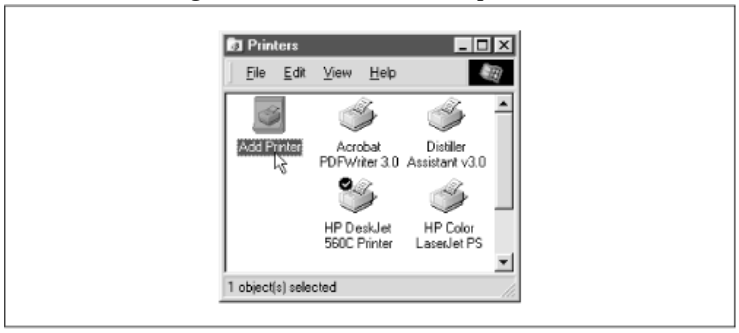


Figura 7.6: Configuración Automática del Driver de Impresión.

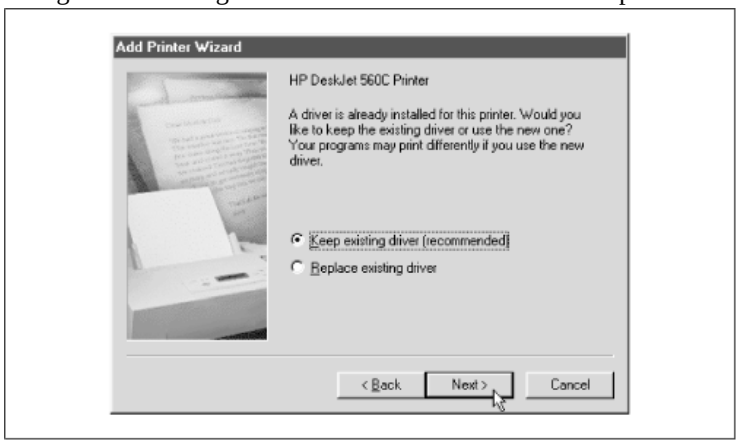
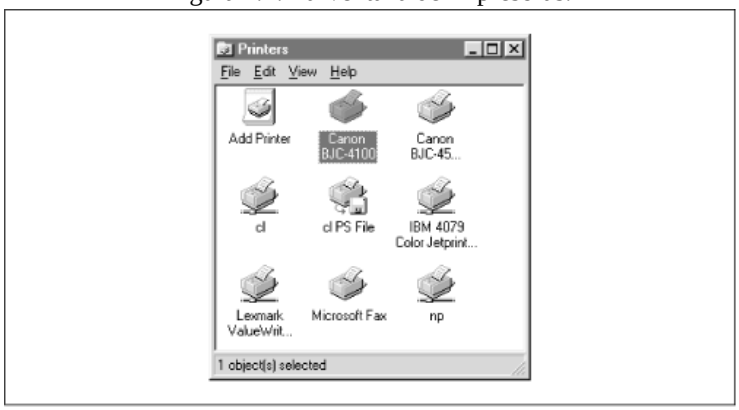


Figura 7.7: La Ventana de Impresoras.



```
# maximum file size (none)
:sh:\

# surpress burst header (no)
:if=/usr/local/samba/bin/smbprint:

# text filter
```

Una vez hecho esto, necesitas crear un fichero de configuración en el directorio de colas (spool) que especificaste con el anterior parámetro *sd* (puede que necesites crear dicho directorio). El fichero debe tener el nombre *.config* y debería contener la siguiente información:

- El nombre NetBIOS de la máquina Windows que tiene la impresora.
- El nombre del servicio que representa a la impresora.
- La contraseña usada para acceder al servicio.

Los dos últimos parámetros fueron configurados en el cuadro de Compartición de la máquina Windows. En este caso, el fichero *.config* debería tener tres líneas:

```
server = phoenix
service = CANON
password = ""
```

Una vez hecho esto, reinicia el server Samba e intenta imprimir usando cualquier programa Unix.

7.2.2. Impresoras System V.

Enviar trabajos de impresión a una impresora Unix System V es algo más sencillo. Aquí, necesitas obtener el script *smbprint.sysv* del directorio */usr/local/samba/examples/printing* y hacer lo siguiente:

- Cambia los parámetros *server*, *service*, y *password* en el script para que coincidan con la máquina NetBIOS, su impresora compartida, y su password, respectivamente. Por ejemplo, las siguientes entradas deberían ser correctas para el servicio del ejemplo que pusimos antes:

```
server = phoenix
service = CANON
password = ""
```

- Ejecuta los siguientes comandos, que crearán una referencia para la impresora en el fichero de capacidades de impresora. Advierte que la nueva entrada de impresora Unix se llama *canon_ printer*:

```
# lpadmin -p canon_printer -v /dev/null -i./smbprint.sysv
# enable canon_printer
# accept canon_printer
```

Una vez hayas hecho esto, reinicia los demonios Samba e intenta imprimir usando cualquier programa Unix. Deberías poder enviar datos a una impresora de un cliente Windows a través de la red.

7.2.3. Opciones de Impresión de Samba.

La Tabla 7.2 resume las opciones de impresión de Samba.

7.2.3.1. printing.

La opción de configuración *printing* le indica a Samba algo acerca de vuestro sistema de impresión Unix, en éste caso qué intérprete o parseador de impresión utilizar. Con Unix, hay varias familias diferentes de comadnos para control de impresión y control de estado de impresión. Samba soporta siete tipos diferentes, como ves en la Tabla 7.3.

El valor para esta opción será uno de los siete posibles. Por ejemplo:

```
printing = SYSV
```

El valor por defecto para esta opción es 'Dependiente de Sistema' y se configura cuando Samba es compilado por vez primera. Para la mayoría de sistemas, el script *configure* automáticamente detectará el sistema de impresión a ser usado y lo configurará adecuadamente en el *Makefile* de Samba. Sin embargo, si tu sistema es PLP, LPRNG, o QNX, necesitarás especificarlo explícitamente en el *Makefile o en el recurso de impresión.*

Los tipos de sistemas más comunes son BSD y SYSV. Cada una de las impresoras en un sistema BSD Unix están descritas en el fichero de capacidades de impresoras (normalmente */etc/printcap*).

Estableciendo la opción de configuración *printing* automáticamente se establecen también al menos otras tres opciones de impresión para el servicio en cuestión: *print command*, *lpq command*, y *lprm command*. Si estás ejecutando Sambasobre un sistema que no tiene soporte para ninguno de estos estilos de impresión, simplemente establece el valor para cada uno de estos comandos manualmente.

7.2.3.2. printable.

La opción *printable* debe establecerse a *yes* en orden a marcar un recurso como servicio de impresión. Si esta opción no se establece, el recurso será tratado como una unidad de disco. Puedes configurar esta opción así:

```
[printer1]
  printable = yes
```

7.2.3.3. printer.

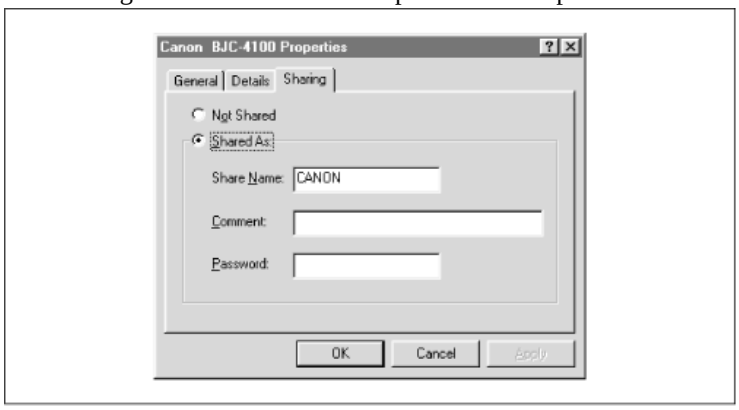
Esta opción, también llamada *printer name*, especifica el nombre de la impresora en el servidor a la cual el recurso apunta. Esta opción no tiene valor por defecto y debería ser configurada explícitamente en el fichero de configuración, aunque los sistemas Unix por sí solos frecuentemente reconocen un nombre por defecto tal como *lp* para una impresora. Por ejemplo:

```
[deskjet]
  printer = hpdkjet1
```

Cuadro 7.2: Opciones de Configuración de Impresoras.

Opción	Parámetros	Función	Defecto	Ambito
printing	bsd, sysv, hpux, aix, qnx, plp, softq, o lprng	Establece el tipo de sistema de impresión para nuestro sistema Unix.	Dependiente de Sistema	Recurso
printable (print ok)	booleano	Marca un recurso como recurso de impresión.	no	Recurso
printer (nombre impresora)	string (nombre impresora Unix)	Establece el nombre de la impresora a ser mostrado a los clientes.	Dependiente de Sistema	Recurso
printer driver	string (nombre driver impresora)	Establece el nombre del driver que debería ser usado por el cliente para enviar datos a la impresora.	Ninguno	Recurso
printer driver file	string (ruta completa)	Establece el nimbre del fichero del driver de impresora.	Ninguno	Global
printer driver location	string (nombre ruta de red)	Especifica el nombre de la ruta del recurso para el fichero del driver de impresora.	Ninguno	Recurso
lpq cache time	numerico (tiempo en segundos)	Establece la cantidad de tiempo en segundos que Samba cacheará el lpq status.	10	Global
postscript	booleano	Trata todos los trabajos de impresión enviados como postscript precediendolos con %! al principio de cada fichero.	no	Recurso
load printers	booleano	Carga automáticamente cada una de las impresoras en el fichero *printcap* como recursos compartidos.	no	Global
print command	string (ruta completa)	Establece el comando Unix para realizar la impresión.	Ver más abajo	Recurso
lpq command	string (ruta completa)	Establece el comando Unix para retornar el status de la cola de impresión.	Ver más abajo	Recurso
lprm command	string (ruta completa)	Establece el comandoUnix para eliminar un trabajo de la cola de impresión.	Ver más abajo	Recurso
lppause command	string (ruta completa)	Establece el comando Unix para pausar un trabajo en la cola de impresión.	Ver más abajo	Recurso
lpresume command	string (ruta completa)	Establece el comando Unix para reactivar un trabajo pausado de la cola de impresión.	Ver más abajo	Recurso
printcap name (printcap)	string (ruta completa)	Especifica la localización del fichero de capacidades de impresora.	Dependiente de Sistema	Global
min print space	numerico (tamaño en kilobytes)	Establece la cantidad mínima de espacio en disco	0	Recurso

Figura 7.8: Ventana de Compartición de Impresora.



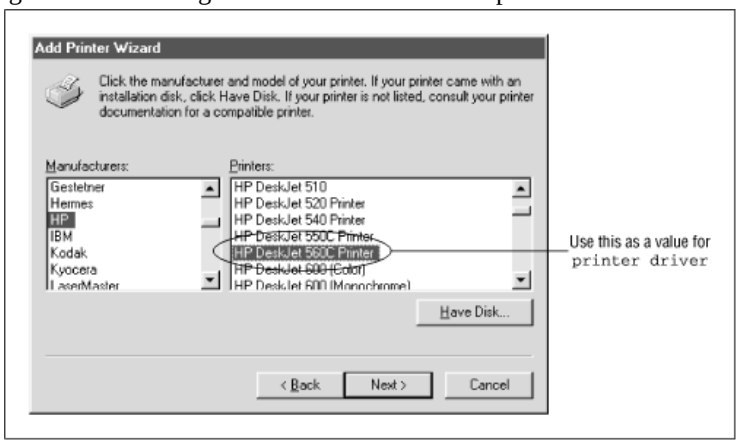
Cuadro 7.4: Tipos de Impresión.

Variable	**Definición**
BSD	Berkeley Unix system
SYSV	System V
AIX	AIX Operating System (IBM)
HPUX	Hewlett-Packard Unix
QNX	QNX Realtime Operating System (QNX)
LPRNG	LPR Next Generation (Powell)
SOFTQ	SOFTQ system
PLP	Portable Line Printer (Powell)

7.2.3.4. printer driver

La opción *printer driver* establece la cadena que Samba usa para indicarle a Windows cuál es la impresora. Si esta opción se configura correctamente, El Asistentee de Configuración de Impresoras de Windows ya sabrá cuál es la impresora, haciendo su instalación más simple para los usuarios finales gracias a un diálogo menos que se les tiene que presentar. La cadena dad debería coincidir con la cadena que se muestra en el Asistente de Impresoras, como se ve en la Figura 7.9. Por ejemplo, una impresora Apple LaserWriter normalmente usa 'Apple LaserWriter'; una Hewlett Packard Deskjet 560C usa 'HP DeskJet 560C Printer'.

Figura 7.9: El Diálogo de Inserción de nueva Impresora en Windows 98.



La configuración automática de drivers de impresora con Samba se explica en más detalle en la sección 7.1.77.1.7, 'Configurando Automáticamente Drivers de Impresión'.

7.2.3.5. printer driver file

Esta opción global da la localización del fichero driver de la impresora Windows 95/98, el cual es necesario para proporcionar los drivers de la impresora a los clientes que usen un impresora Samba. El valor por defecto para esta opción es */usr/local/samba/lib/printers.def*. Puedes machacar este valor por defecto como sigue:

```
[deskjet]
  printer driver file = /var/printers/printers.def
```

Esta opción es explicada con más detalle en la sección 7.1.77.1.7.

7.2.3.6. printer driver location

Esta opción especifica un recurso que contiene drivers de impresoras Windows 95 y 98 y ficheros de definición. No hay valor por defecto. Puedes especificar la localización como una ruta de red. Una aproximación frecuente es usar un recurso en tu propia máquina, como se muestra aquí:

```
[deskjet]
  printer driver location = \\%L\PRINTER$
```

Esta opción es explicada con más detalle en la sección 7.1.77.1.7.

7.2.3.7. lpq cache time

Esta opción te permite establecer el número de segundos durante los cuales Samba recordará el estado actual de la impresora. Una vez este tiempo transcurra, Samba ejecutará un comando *lpq* (o el que hayas especificado en la opción *lpq command*) para tener información de última mano acerca del estado. Por defecto se establece a 10 segundos, pero puede ser incrementado si tu comando *lpq* se toma un tiempo inusualmente largo en ejecutarse o tienes muchos clientes. El siguiente ejemplo establece el tiempo a 30 segundos:

```
[deskjet]
  lpq cache time = 30
```

7.2.3.8. postscript

La opción *postscript* fuerza a la impresora a tratar todos los datos enviados como si fueran Postscript. Esto se ahce precediendo los caracteres %! al principio de la primera línea de cada trabajo. Se usa normalmente con PCs que inserten un ^D (control-D o 'marca de fin de fichero') al principio de la primera línea de un fichero PostScript. Esto, obviamente, no convierte a una impresora no-PostScript en una PostScript. El valor por defecto para esta opción es 'no'. Puedes cambiar el valor así:

```
[deskjet]

  postscript = yes
```

7.2.3.9. print command, lpq command, lprm command, lppause command, lpresume command

Estas opciones le indican a Samba qué comandos Unix usar para controlar y enviar datos a la impresora. Los comandos Unix que se ven involucrados aquí son: *lpr* (send to Line PRinter), *lpq* (List Printer Queue), *lprm* (Line printer ReMove), y opcionalmente *lppause* y *lpresume*. Samba proporciona una opción nominada para cada uno de estos comandos, para el caso de que necesites modificar cualquiera de los valores por defecto del sistema. Por ejemplo:

```
lpq command = /usr/ucb/lpq %p
```

Esto establecería como comando *lpq* al ejecutable */usr/ucb/lpq*. Similarmente:

```
lprm command = /usr/local/lprm -P%p %j
```

indicaría usar como comando de eliminación de trabajos al ejecutable */usr/local/lprm*, y le proporciona el número de trabajo de impresión mediante la variable *%j*.

Los valores por defecto para cada una de estas opciones son dependientes del valor de la opción *printing*. La Tabla 7.4 muestra los comandos por defecto para cada una de las opciones de impresión. El sistema de impresión más popular es el BSD.

Normalmente no es necesario modificar estas opciones en Samba, con la posible excepción de *print command*. Esta opción puede que sea necesacio definirla explícitamente si tu sistema de impresión no tiene una opción -r (eliminar tras imprimir) en el comando de impresión. Por ejemplo:

```
/usr/local/lpr -P%p %s; /bin/rm %s
```

Con un mínimo de juiciosa programación, estas opciones de *smb.conf* también se pueden usar par depuración de errores:

```
print command = cat %s >>/tmp/printlog; lpr -r -P%p %s
```

Por ejemplo, esta configuración puede verificar qué ficheros están actualmente siendo enviados al servidor Samba. Si los hay, sus contenidos se mostrarán en el fichero */tmp/printlog.*

Tras BSD, el siguiente sistema de impresión más popular es SYSV (o System V), además de algunas variantes SYSV para IBM's AIX y Hewlett-Packard HP-UX. Este sistema no tiene un fichero */etc/printcap*. En su lugar, la opción *printcap file* puede configurarse con un apropiado comando *lpstat*. Esto le dice a Samba que obtenga una lista de impresoras desde el comando *lpstat*. Alternativamente, puedes establecer la opción de coniguración global *printcap name* al nombre de una un fichero printcap desmilitarizado (dummy) que tú proporciones. En este último caso, el fichero debe contener una serie de líneas como estas:

```
lp|print1|My Printer 1
print2|My Printer 2
print3|My Printer 3
```

Cada línea nomina un impresora, y proporciona alias para ella. En este ejemplo, la primera impresora es llamada lp, print1, o My Printer 1, cualquiera que el usuario prefiera usar es válida. El primer nombre será usado en lugar de %p en cualquier comando que Samba ejecute para esa impresora.

Dos tipos adicionales de impresión son también soportados por Samba: LPRNG (LPR New Generation) y PLP (Public Line Printer). Estos son sistemas de dominio público y Open Source, y son utilizados por muchos sitios con problemas de adquisición de licencias de software. En adición, los sistemas SOFTQ y QNX también son soportados por Samba.

7.2.3.10. load printers

Esta opción le dice a Sambaque cree recursos para todos los nombres de impresoras conocidos y los cargue en la lista de navegación. Samba creará y listará un recurso de impresión por cada nombre de impresora en */etc/printcap* (o equivaente de tu sistema). Por ejemplo, si tu fichero *printcap* se parece a este[3]:

```
lp:\
  :sd=/var/spool/lpd/lp:\

# directorio de cola
  :mx#0:\

# tamaño máximo fichero (ninguno)
  :sh:\
```

[3]Hemos colocado comentarios par el caso de que no te hayas metido nunca conn este fichero.

```
# surpress burst header (no)
  :lp=/dev/lp1:\

# nombre dispositivo para salida
  :if=/var/spool/lpd/lp/filter:

# texto filtro
laser:\
  :sd=/var/spool/lpd/laser:\

# directorio de cola
  :mx#0:\

# tamaño máximo fichero (ninguno)
  :sh:\

# surpress burst header (no)
  :lp=/dev/laser:\

# nombre dispositivo para salida
  :if=/var/spool/lpd/lp/filter:

# texto filtro
```

y tú especificas:

```
load printers = yes
```

Los recursos [lp] y [laser] serán creados automáticamente como recursos válidos de impresión cuando se inicie Samba. Ambos recursos tomarán las opciones de configuración especificadas en la sección [printers] para configurarse a sí mismos, y estarán disponibles en la lista de navegación para el servidor Samba.

7.2.3.11. printcap name

Si la opción *printcap name* (también llamada *printcap*) aparece en un recurso de impresión, Samba usará el fichero especificado como fichero de capacidades de impresión del sistema. Este es normalmente */etc/printcap*. sin embargo, puedes cambiarlo a otro fichero que contenga sólo las impresoras que quieras dar a compartir sobre la red. El valor debe ser el nomrbe de un fichero de capacidades de impresoras completamente cualificado en el servidor:

```
[deskjet]
  printcap name = /usr/local/printcap
```

7.2.3.12. min print space

Esta opción establece la cantidad de espacio de cola que debe estar disponible en disco antes de que se permita la impresión. Estableciendo dicho valor a cero (por defecto) se desactiva la opción; estebleciendo cualquier otro número se establece la cantidad

de espacio libre requerida en kilobytes. Esta opción te ayuda a evitar tener trabajos de impresión ocupando el espacio libre restante del disco, lo cual puede provocar que caigan otros procesos:

```
[deskjet]
  min print space = 4000
```

7.2.3.13. queuepause command

Esta opción especifica un comando que le dice a Samba cómo pausar una cola de impresión por completo, no sólo un único trabajo. El valor por defecto depende del tipo de sistema de impresión escogido. No deberías necesitar modificar este valor.

7.2.3.14. queueresume command

Esta opción especifica un comando que le dice a Samba cómo reactivar una cola de impresión interrumpida al completo. El valor por defecto depende del tipo de sistema de impresión escogido. No deberías necesitar modificar este valor.

7.3. Resolución de Nombres con Samba

Antes de que entraran en juego los Servidores de Nombres NetBIOS, o NetBIOS Name Servers (NBNS) , la resolución de nombres trabajaba enteramente mediante difusión (broadcast). Si necesitabas la dirección de una máquina, simplemente hacías multidifusión de su nombre sobre toda la red y, en teoría, la máquina en cuestión debería responder a la llamada. Esta aproximación es todavía posible: cualquiera que esté buscando a una máquina llamada *fred* puede todavía hacer un broadcast con la petición, y encontrarla si existe y averigurar cuál es su dirección IP (nosotros usaremos esta capacidad para resolver los problemas del servicio de nombres de Samba con em comando *nmblookup* en el *Capítulo 9, Resolución de Problemas en Samba*).

Como ya vimos en el primer capítulo, sin embargo, el broadcasting o multidifusión no pasa fácilmente de unas subredes a otras, a través de múltiples subredes. En adición, muchos broadcasts tienden a tirar las redes. Para resolver este problema, Microsoft ahora proporciona el *Windows Internet Naming Service* (WINS), un NBNS a nivel de red, el cual Samba soporta. Con él, un administrador puede designar una única máquina para que actúe como servidor WINS, y puede entonces proporcionar a cada cliente que requiera resolución de nombres la dirección del servidor WINS. Consecuentemente, las peticiones de registro y de resolución de nombres pueden ser dirigidas a una misma máquina desde cualquier punto de la red, en lugar de hacer broadcast.

WINS y el broadcasting no son los únicos mecanismos para la resolución de nombres. Hay actualmente cuatro mecanismos que pueden ser usados con Samba:

- WINS
- Broadcasting
- Unix /etc/hosts or NIS/NIS+ matches
- LMHOSTS file

Samba can use any or all of these name resolution methods in the order that you specify in the Samba configuration file using the name resolve order parameter. However, before delving into configuration options, let's discuss the one that you've probably not encountered before: the LMHOSTS file.

7.3.1. El Fichero LMHOSTS

LMHOSTS es el fichero de gestión de máquinas de red standard, usado para resolver nombres a direcciones IP en el sistema. Este es el equivalente NBT del fichero */etc/hosts* que es un estandard en todos los sistemas Unix. Por defecto, el fichero es normalmente almacenado como */usr/local/samba/lib/LMHOSTS* y comparte un formato similar al de */etc/hosts*. Por ejemplo:

```
192.168.220.100   hydra
192.168.220.101   phoenix
```

La única diferencia es que los nombres de la columna derecha son nombres NetBIOS en vez de nombres DNS. Debido a que son nombres NetBIOS, también les puedes asignar tipos de recursos, como sigue:

```
192.168.220.100   hydra#20
192.168.220.100   simple#1b
192.168.220.101   phoenix#20
```

Aquí, hemos asignado a la máquina *hydra* el ser el controlador de dominio primario para el dominio SIMPLE, lo indicamos con el tipo de recurso <1B>asignado tras el nombre de máquina en la segunda de las tres líneas. Las otras dos son para definir estaciones de trabajo standard.

Si deseas ubicar el fichero LMHOSTS en algúuna otra ubicación, necesitarás notificarlo al proceso nmbd en el arranque, como sigue:

```
nmbd -H /etc/samba/lmhosts -D
```

7.3.2. Configurando Samba para usar otro Servidor WINS

Puedes configurar Samba para usar otro server WINS que se encuentre en la red, simplemente apuntando a la dirección IP de dicho servidor WINS. Esto se hace con la opción de configuración global *wins server*, como se muestra aquí:

```
[global]
  wins server = 192.168.200.122
```

Con esta opción activada, Samba redirigirá todas las peticiones WINS al servidor que se encuentra en 192.168.200.122. Advierte que debido a que la petición es redirigida a una única máquina, no tenemos que preocuparnos sobre los problemas inherentes a una multidifusión. Sin embargo, aunque hayas especificado una dirección IP de un servidor WINS en el fichero de configuración, Samba no usará necesariamente ese server WINS antes que otras formas de resolución de nombres. El orden en que Samba usa las diferentes técnicas para resolución de nombres es definido por la opción de configuración *name resolve order*, la cual discutiremos dentro de poco.

Si tienes un server Samba sobre una subred que todavía usa broadcasting, y el servidor Samba conoce la correcta localización de un servidor WINS en otra subred,

puedes configurar el servidor Samba para que reenvíe cualquier petición de resolución, con la opción *wins proxy*:

```
[global]
  wins server = 192.168.200.12
  wins proxy = yes
```

Usa esto sólo en aquellas situaciones en las que el servidor WINS resida en otra subred. De lo contrario, el broadcast enlazará con el servidor WINS a través de cualquier proxy.

7.3.3. Configurando Samba como Servidor WINS

Puedes configurar Samba como sever WINS configurando dos opciones globales del fichero de configuración:

```
[global]
  wins support = yes
  name resolve order = wins lmhosts hosts bcast
```

La opción *wins support* convierte a Sambaen un servidor WINS. Lo creas o no, ¡Es todo lo que necesitas hacer! Samba maneja el resto de detalles detrás del escenario, convirtiéndote en un relajado administrador. Las opciones *wins support=yes* y *wins server* son mutuamente excluyentes; no puedes al mismo tiempo ofrecer a Samba Samba como server WINS y además apuntar a otro sistema para que actúe como servidor.

Si Samba está actuando como server WINS, deberías familiarizarte con la opción *name resolve order* mencionada anteriormente. Esta opción le dice a Samba el orden a seguir en cuanto a la utilización de métodos para la resolución de un nombre NetBIOS. Puede tomar hasta cuatro valores:

lmhosts Usa el fichero de control de red LMHOSTS.

hosts Usa los métodos de resolución de nombres standard de un sistema Unix system, */etc/hosts*, *DNS*, *NIS*, o una combinación (según esté configurado en dicho sistema).

wins Usa el servidor WINS.

bcast Usa un método de multidifusión o broadcast.

El orden en que los especificas es el orden en que Samba intentará la resolución de nombres cuando actúe como servidor WINS. Por ejemplo, echemos un vistazo al valor expuesto arriba:

```
name resolve order = wins lmhosts hosts bcast
```

Esto significa que Samba intentará usar primero sus entradas WINS par la resolución de nombres, y a continuación el fichero LMHOSTS de su sistema. Después, el valor *hosts* provoca que use los métodos Unix para la resolución de nombres. La palabra *hosts* puede llegar a engaño; no sólo cubre el fichero */etc/hosts*, sino también el uso de DNS o NIS (según esté configurado en el sistema Unix). Finalmente, si ninguno de los tres funcionó, usará broadcast para intentar localizar la máquina correcta.

Por último, puedes instruir al server Samba para que actúe como WINS que chequee co el servidor DNS del sistema si una petición no pudo ser encontrada en su base de datos WINS. Con un típico sistema Linux, por ejemplo, puedes encontrar la dirección IP del servidor DNS buscando el fichero */etc/resolv.conf*. En su interior, deberías ver una entrada parecida a esta:

```
nameserver 127.0.0.1
nameserver 192.168.200.192
```

Esto nos indica que un servidor DNS se encuentra en 192.168.220.192 (La IP 127.0.0.1 es la dirección de la máquina local, y nunca es una dirección DNS válida).

Usa la opción global *dns proxy* para indicar a Samba que use el servidor DNS:

```
[global]
  wins support = yes
  name resolve order = wins lmhosts hosts bcast
  dns proxy = yes
```

7.3.4. Opciones de Configuraión de Resolución de Nombres

Las opciones se muestran en la Tabla 7.5.

7.3.4.1. wins support

Samba proporcionará servicio de nombres WINS a todas las máquinas de la red si estableces lo siguiente en la sección [global] del fichero *smb.conf*:

```
[global]
  wins support = yes
```

El valor por defecto es *no*, lo cual es usado normalmente para permitir a un servidor Windows NT convertirse en server WINS. Si activas esta opción, recuerda que un servidor WINS Samba actualmente no puede intercambiar datos con ningún servidor de seguridad (o respaldo) WINS NT. Si lo activas, esta opción es mutuamente excluyente con el parámetro *wins server*; no puedes activar ambas opciones a *yes* al mismo tiempo, o Samba dará error.

7.3.4.2. wins server

Samba usará un server WINS existente en la red si especificas esta opción en tu fihero de configuración. El valor para esta opción es la dirección IP o el nombre DNS (no el nombre NetBIOS) del servidor WINS. Por ejemplo:

```
[global]
  wins server = 192.168.220.110
```

o:

```
[global]
  wins server = wins.example.com
```

Para que esto funcione, la opción *wins support* debe estar a *no* (por defecto). De lo contrario, Samba reportará un error. Puedes especificar sólo un servidor WINS usando esta opción.

Cuadro 7.5: Comandos por defecto para varios comandos de impresión.

Opción	BSD, AIX, PLP, LPRNG	SYSV, HPUX	QNX	SOFTQ
print command	lpr -r -P %p %s	lp -c -d %p %s; rm %s	lp -r -P %p %s	lp -d %p -s %s; rm %s
lpq command	lpq -P %p	lpstat -o %p	lpq -P %p	lpstat -o %p
lprm command	lprm -P %p %j	cancel %p- %j	cancel %p- %j	cancel %p- %j
lppause command	lp -i %p- %j -H hold (sólo SYSV)	Ninguno	Ninguno	Ninguno
lpresume command	lp -i %p- %j -H resume(sólo SYSV)	Ninguno	Ninguno	qstat -s -j %j -r

Cuadro 7.6: Opciones WINS

Opción	Parámetros	Función	Defecto	Ambito
wins support	booleano	Si se estabelce a *yes*, Samba actuará como server WINS.	no	Global
wins server	string (dirección IP o nombre DNS)	Identifica un server WINS para ser usado por Samba para registro y resolución de nombres.	ninguno	Global
wins proxy	booleano	Permite a Samba actuar como proxy de un server WINS en otra subred.	no	Global
dns proxy	booleano	Si vale *yes*, un server Samba buscará DNS si no puede encontrar un nombre en WINS.	no	Global
name resolve order	lmhosts, hosts, wins, o bcast	Especifrica un orden para los métodos usados para resolver nombres NetBIOS.	lmhosts hosts wins bcast	Global
max ttl	numérico	Especifica el tiempo de vida máximo en segundos para una petición de nombres NetBIOS.	259200 (3 días)	Global
max wins ttl	numérico	Especifica el tiempo máximo de vida en segundos para nombres NetBIOS distribuidos por Samba como server WINS.	518400 (6 días)	Global
min wins ttl	numérico	Especifica el mínimo tiempo de vida en segundos para nombres NetBIOS distribuidos por Samba como server WINS.	21600 (6 horas)	Global

7.3.4.3. wins proxy

Esta opción permite que Samba actúe como proxy para otro servidor WINS, y que además haga 'relay' o reenvío de las peticiones de registro y resolución de nombres que le lleguen a él hacia el verdadero servidor WINS, normalmente fuera de la actual subred. El server WINS puede ser indicado a través de la opción *wins server*. El proxy retornará la respuesta WINS al cliente. Puedes activar esta opción especificando lo siguiente en la sección [global]:

```
[global]
  wins proxy = yes
```

7.3.4.4. dns proxy

Si queires que el Servidor de Nombres de Dominio (DNS) sea usado si un nombre no es encontrado en WINS, puedes establecer la siguiente opción:

```
[global]
  dns proxy = yes
```

Esto provocará que *nmbd* haga peticiones para nombres de máquinas usando el servicio de nombres de dominio estandar. Puede que desees desactivar esta opción si no tienes una conexión permanente con tu servidor DNS. Recomendamos usar un servidor WINS. Si no tienes ningún server WINS en tu red, convierte a la máquina Samba en server WINS. No conviertas, sin embargo, a dos máquinas Samba en servidores WINS (uno primario y el otro de respaldo) ya que actualmente no pueden intercambiar sus bases de datos WINS.

7.3.4.5. name resolve order

La opción global *name resolve order* especifica el orden de los servicios que Samba usará cuando intente resolución de nombres. El orden por defecto es usar el fichero LMHOSTS, seguido de los métodos de resolución Unix estandares (una combinación de */etc/hosts*, *DNS*, y *NIS*), después interroga a una server WINS, y finalmente usa broadcasting para determinar la dirección de un nombre NetBIOS. Puedes modificar esto especificando el cambio como sigue:

```
[global]
  name resolve order = lmhosts wins hosts bcast
```

Esto causa que la resolución use primero el fichero LMHOSTS, luego interroge a un server WINS, el fichero de máquinas del sistema, y finalmente haga broadcasting. No necesitas usar las cuatro opciones si no quieres. Esta opción se cubre con más detalle en la sección *7.3.3, Configurando Samba como Servidor WINS.*

7.3.4.6. max ttl

Esta opción le da el tiempo máximo de vida (TTL) durante el cual un nombre NetBIOS registrado en el servidor Samba permanecerá activo. No deberías alterar nunca este valor.

7.3.4.7. max wins ttl

Esta opción da el máximo tiempo de vida (TTL) durante el cual un nombre NetBIOS resuelto por un server WINS permanecerá activo. No deberías alterar nunca este valor.

7.3.4.8. min wins ttl

Esta opción da el tiempo mínimo de vida (TTL) durante el cual un nombre NetBIOS resuelto por un server WINS permanecerá activo. No deberías alterar nunca este valor.

Capítulo 8

Informacion adicional sobre Samba

Este capitulo incorpora a nuestra revision del archivo de configuracion smb.conf algunas otras opciones interesantes que nos permitiran llevar a cabo distintas tareas.

Vamos a ver, en breve, opciones que nos permitiran dar soporte a programadores, internalizacion, como mandar mensajes y solucionar bugs tipicos de sistemas Windows. La mayoria de los casos, estas opciones solo las usaremos en contadas ocasiones. Tambien cubriremos el como hacer copias de seguridad con smbtar, asi como la automatizacion de este proceso, pero eso sera al final del capitulo. Asi que, sin mas preambulos, vayamos al primer punto...

8.1. Dando soporte a Programadores

Si disponemos de un servidor Samba, al que tengan acceso programadores, son de especial interes las opcienes listadas en la siguiente relacion:

8.1.1. Sincronizando el Tiempo

La sincronizacion del tiempo puede ser muy importante para un programador. Consideremos las siguientes acciones:

```
time service=yes
dos filetimes=yes
fake directory create times=yes
dos filetimes resolution=yes
delete readonly=yes
```

Si establecemos estas opciones afirmativamente (como se puede ver arriba), las comparticiones Samba proveeran el tipo de ficheros de tiempo que visual C++, nmake y otras aplicaciones Microsoft requieren para poder programar con ellas.

De no hacer esto, los programas de tipo PC Make tenderan a pensar que todos los ficheros en un determinado directorio han de ser recompilados a cada momento. Evidentemente, este no es el tipo de resultado, de conducta, que nosotros esperamos y queremos de este tipo de programas.

8.1.1.1. time server

Si nuestro servidor Samba dispone de un reloj preciso, o si ese servidor es un cliente a su vez de alguno de los servidores de tiempo UNIX de la Red, podemos "instruirlo" para que se identifique a si mismo como un servidor de tiempo SMB. Esto se puede hacer fijando la opcion *time service* de la forma:

```
[global]
  time service=yes
```

Una vez hecho esto, desde el cliente DOS, hemos de pedirle al servidor el tiempo correcto, lo que haremos poniendo en una linea de comandos del DOS el siguiente comando:

```
c:\>NET TIME

\\server /YES /SET
```

Donde Server es el nombre, naturalmente, de nuestro servidor Samba. Luego, para no tener que repetir este proceso en cada inicio de un cliente Windows, podemos meter este comando en alguno de los sripts de inicio de Windows (Para mas informacion, ver capitulo *6, Usuarios, Seguridad y Dominios*).

Por defecto, nos encontraremos con que esta opcion se encuentra desactivada. Si activamos este servicio, podremos usar el anterior comando para conseguir que los relojes de los clientes no vayan a la deriva.

La sincronizacion del tiempo es muy importante para clientes que usen programas del tipo *make*, que compila en base al tiempo en el que fue modificado por ultima vez el fichero. Una sincronizacion incorrecta de tiempos puede causar que este tipo de programas "rehaga" todos los archivos que se encuentren en ese directorio, lo que desperdicia tiempo, o lo que es peor, podria no compilar un fichero en codigo fuente que fuese modificado, a causa de un leve error en el reloj.

8.1.1.2. time offset

Para tratar clientes que no procesen adecuadamente el tiempo del dia que es, es decir, clientes que no sepan la franja horaria que ocupan, de forma adecuada, Samba nos ofrece la opcion *time offset*. Si esta activada esta opción, lo que hace es añadir un numero especifico de minutos al tiempo actual del sistema. Esto seria muy util si nos encontrasemos, por ejemplo, en el caso del pais Newland, y Windows no reconociese los 30 minutos de diferencia con la franja horaria mas crecana, entonces hariamos lo siguiente:

```
[global]
  time offset=30
```

8.1.1.3. dos filetimes

Normalmente en un sistema UNIX, tan solo el superusuario (root) y el dueño de un fichero, pueden modificar la fecha en la que fue modificado el fichero por ultima vez.

La opcion, a nivel de comparticion, *dos filetimes* permite a nuestro servidor Samba imitar las caracteristicas de un sistema DOS/Windows, es decir, cualquier usuario podra modificar la fecha de la ultima modificacion hecha a un archivo dentro de una

comparticion, siempre y cuando tenga permiso de escritura sobre ese fichero. Para poder hacer esto, Samba usa sus privilegios de root para modificar el registro de tiempo del fichero.

Por defecto, esta opcion esta desactivada. Deberiamos de activarla para que programas del tipo PC make, citados anteriormente, funcionasen correctamente. Si se encontrase desactivada, este tipo de programas no podrian cambiar la fecha de la ultima modificacion por si mismos. Esto suele resultar en que el programa "piensa" que ha de compilar todos los ficheros, incluso los que no necesitan ser compilados.

8.1.1.4. dos filetime resolution

dos filetime resolution es una opcion a nivel de comparticion de Samba. Si la acivamos, Samba se las arreglará para redondear la fecha del fichero a la frontera dos segundos mas cercana. Esta opcion sirve principalmente para satisfacer un 'quirk' en Windows que prevenga al lenguaje Visual C++ para que reconozca debidamente, si un fichero ha cambiado, es decir, si ese fichero ha sido modificado. Podemos activar esta opcion de la siguiente forma:

```
[data]
  dos filetime resolution=yes
```

Es recomendable usar esta opcion si vamos a usar Microsoft Visual C++ en una comparticion Samba que soporte bloqueos eventuales.

8.1.1.5. fake directory create times

Esta opcion existe para mantener a los programas del tipo PCMake "sanos", es decir, de protegerlos de posible eventualidades. Los sistemas de archivos VFAT y NTFS guardan la fecha de creacion de un directorio,mientras que los sistemas UNIX no lo hacen.

Sin esta opcion, Samba coge la fecha mas reciente que tiene, la ultima fecha de la que hay constancia que se ha modificado algo en algun fichero, o se ha agregado algun fichero, y devuelve dicha fecha al cliente.

Si esto no fuese suficiente, debemos de poner la opcion siguiente, bajo una comparticion:

```
[data]
  fake directory create times=yes
```

Al activar esta opcion, Samba ajustara la fecha de creacion del directorio que ha de pasar al cliente, en la fecha 1 de enero de 1980. Esto sirve basicamente, para convencer al nmake del lenguaje Visual C++ de que cualquier fichero que se encuentre dentro de un directorio es realmente mas reciente que la fecha de creacion del directorio que lo contiene, y que entonces ha de ser compilado.

8.2. Magic Scripts (Scripts Magicos)

Los "Scripts Magicos" son un método que consiste en ejecutar una aplicacion en un sistema UNIX y luego, de algun modo, redireccionar las salidas de ese programa a un cliente Samba.

Por ahora se encuentra en fase experimental (por lo menos mientras estoy traduciendo esto). Pero de todas formas se puede utilizar, y llegado el momento podriansernos de utilidad.

Este tipo de "Scripts Magicos" no son muy fiables, y su uso es extremadamente desaconsejado por los desarrolladores de Samba. En las siguientes tablas podemos ver mas informacion sobre estas opciones.

8.2.1. magic script

Si esta opcion se encuentra activada para un nombre de un fichero, y el cliente crea un fichero con ese nombre en esa comparticion Samba, Samba ejecutara el fichero tan pronto como el usuario lo abra y lo cierre (ese fichero). Vamos a tomar el siguiente ejemplo, esta opcion ha sido creada en el recurso [accounting]:

```
[accounting]
  magic script = tally.sh
```

Samba monitoriza continuamente los ficheros en esa comparticion. Si cualquier fichero con el nombre *tally.sh* es cerrado, despues de ser abierto por un usuario, Samba ejecutara los contenidos de ese fichero en modo local. El archivo es pasado a la shell para que sea ejecutado por la misma, por lo que deberemos de tener en el sistema alguna shell UNIX valida. Esto tambien significa que el fichero ha de tener caracteres de retorno de carro al final de cada linea en lugar del sistema usado por windows.

Ademas, ayuda bastante si hacemos uso de la directiva #! al principio del fichero, para indicar bajo qué shell deberia de ejecutarse el script.

8.2.2. magic output

Esta opcion especifica un fichero donde guardar las salidas o outputs que cree el script especificado por el Magic script. Se debe especificar el nombre de un fichero en un directorio en el que se pueda escribir, de otra forma al intentar escribir sobre el directorio, dara error.

```
[accounting]
  magic script = tally.sh
  magic output = /var/log/magicoutput
```

Si no activasemos esta opcion, el fichero en el que son grabadas las salidas del script por defecto, es el nombre de ese script (el que especificamos en la opcion magic script), pero con la extension .out.

8.3. Internationalización

Samba tiene una habilidad limitada para hablar otras lenguas: si necesitas usar caracteres que no pertenecen al ASCII standard, algunas opciones te pueden ayudar, com las que se muestran e la Tabla 8.3. Si no lo necesitas, te puedes saltar esta sección.

Cuadro 8.1: Opciones de Configuración de Programadores.

Opción	Parámetros	Función	Defecto	Alcance
Time Server	Boolean	Si esta activada, nmbd, se presenta a si mismo como un servicio de tiempo SMB para clientes Windows.	No	Global
Time Offset	Numericos (numero de minutos)	Añade un determinado numero de minutos al tiempo mostrado.	0	Global
Dos Filetimes	Boolean	Permite que usuarios que no sean propietarios de un archivo, le puedan cambiar el valor del tiempo, la fecha, siempre que tengan permiso para escribir sobre él.	No	Share
Dos Filetime Resolution	Boolean	Hace que el tiempo de un archivo, se redondee al siguiente segundo.	No	Share
Fake Directory Create Times	Boolean	Fija los tiempos de directorio para evitar posibles bugs o fallos provocados por Microsoft nmake	No	Share

Cuadro 8.2: Opciones de Configuración de Red.

Opción	Parámetros	Función	Defecto	Ambito
magic script	Cadena de caracteres (string), se refiere al nombre completo del archivo	Establece el nombre de un archivo para que este sea ejecutado por Samba, como un usuario conectado, cuando este no esta realmente conectado.	Ninguno	Share
magic output	idem que Magic Script	Establece un archivo para que guarde ahi la salida del Magic Script, es algo asi como un log	Nombre_Magic_Script.out	Share

Cuadro 8.3: Opciones de Configuración de Red.

Opción	Parámetros	Función	Defecto	Ambito
client code page	Descritos en esta sección	Establece un código de página para los clientes	850	Global
character set	Descritos en esta sección	Traslada páginas de códigos a juegos de caracteres Unix alternativos	ninguno	Global
coding system	Descritos en esta sección	Traslada páginas de códigos 932 a un juego de caracteres Asiático	ninguno	Global
valid chars	string (juego de caracteres)	Obsoleto: normalmente añadia caracteres individuales a una página de códigos, y tenía que ser usado tras establecer la página de códigos del cliente.	ninguno	Global

8.3.1. client code page

El juego de caracteres en las plataformas Windows recuerdan al concepto original de una página de códigos. Estas páginas de códigos son usadas por los clientes DOS y Windows para determinar reglas para mapear letras minúsculas a letras mayúsculas. Samba puede ser instruido para usar una variedad de páginas de códigos a través del uso de la opción global *client code page* en orden a establecer la correspondiente página de códigos en uso en el cliente. Esta opción carga un fichero de definición de página de códigos, y puede tomar los valores especificados en la Tabla 8.4.

Cuadro 8.4: Páginas de Códigos Válidas en Samba 2.0.

Página de Códigos	Definición
437	MS-DOS Latin (United States)
737	Windows 95 Greek
850	MS-DOS Latin 1 (Western European)
852	MS-DOS Latin 2 (Eastern European)
861	MS-DOS Icelandic
866	MS-DOS Cyrillic (Russian)
932	MS-DOS Cyrillic (Russian)
936	MS-DOS Simplified Chinese
949	MS-DOS Korean Hangul
950	MS-DOS Traditional Chinese

Puedes establecer la página de códigos del cliente como sigue:

```
[global]
  client code page = 852
```

El valor por defecto para esta opción es 850. Puedes usar la herramienta *make_smbcodepage* que viene con Samba (por defecto en */usr/local/samba/bin*) para crear tus propias páginas de códigos SMB, en el caso de que ninguna de las listadas te sea suficiente.

8.3.2. character set

La opción global *character set* puede ser usada para convertir nombres de fichero ofrecidos a través de una página de códigos DOS (mira la sección anterior, *8.3.1, client code page*) a qeuvalentes que puedan ser representados por otros juegos de caracteres Unix que no sean los de USA. Por ejemplo, si quieres convertir el juego de caracteres MS-DOS Western European en los clientes al juego de caracterers Unix Western European en el servidor, puedes usar lo siguiente en tu fichero de configuración:

```
[global]
  client code page = 850
  character set = ISO8859-1
```

Advierte que debes incluir una opción *client code page* para especificar el juego de caracteres desde el cual estás convirtiendo. El juego de caracteres válido (y sus correspondientes páginas de códigos) que Samba 2.0 accepta se listan en la Tabla 8.5:

Normalmente, la opción *character set* está completamente desactivada.

8.3.3. coding system

La opción *coding system* es similar a la opción *character set*. Sin embargo, su propósito es determianr cómo convertir una página de códigos Japanese Shift JIS a un apropiado juego de caracteres Unix. En orden a usar esta opción, la opción *client code page* descrita anteriormente debe ser establecida a la página 932. Los sistemas de codificación válidos que Samba 2.0 accepta se listan en la Tabla 8.6.

8.3.4. valid chars

La opción *valid chars* es una vieja característica de Samba que añadirá caracteres individuales a una página de códigos. Sin embargo, esta opción está siendo abandonada, en favor de sistemas de codificación más modernos. Puedes usar esta opción como sigue:

```
valid chars = Î
valid chars = 0450:0420 0x0A20:0x0A00
valid chars = A:a
```

Cada uno de los caracteres en la lista especificada debería estar separado por espacios en blanco. Si hay dos puntos entre dos caracteres o sus equivalentes numéricos, el dato a la izquierda de los dos puntos se considera el carácter en mayúsculas, mientras que el de la derecha es considerado el carácter minúscula. Puedes representar ambos caracteres literalmente (si puedes teclearlos) y mediante sus equivalentes representaciones en octal, hexadecimal o decimal.

Recomendamos no usar ésta opción. En su lugar, usa una de las páginas de código standard listadas anteriormente. Si usas esta opción, esta debe estar listada tras la página de códigos cliente a la que quieres añadir el carácter. De los contrario, los caracteres no serán añadidos.

8.4. Mensajes Emergentes

Puedes usar la herramienta de mensajes emergentes o *WinPopup* (WINPOPUP.EXE) de Windows para enviar mensajes a usuarios, máquinas o grupos de trabajo enteros de la red. Esta herramienta se proporciona con Windows 95 OSR2 y viene por defecto en Windows 98. Con Windows 95 o 98, sin embargo, necesitas tener en ejecución WinPopup para recibir mensajes emergentes. Con Windows NT, puedes recibir mensajes sin tener activa la herramienta; aparecerá automáticamente en una pequeña caja de diálogo en la pantalla, cuando el mensaje sea recibido. La aplicación WinPopup se muestra en la Figura 8.1.

Samba tiene una única opción para la mensajería WinPopup, *message command*, como muestra la Tabla 8.7.

8.4.1. message command

La opción *message command* de Samba establece la ruta al programa que se ejecutará en el servidor cuando llegue a él un mensaje emergente desde Windows. El comando será ejecutado usando la cuenta de usuario anónimo. Lo que hacer con uno de estos es cuestionable, ya que es probablemente para el administrador de Samba, y Samba no conoce su nombre. Si sabes que hay un humano usando la consola, el equipo de Samba te sugiere hagas lo siguiente:

Cuadro 8.5: Juegos de Caracteres Válidos con Samba 2.0.

Juego de Caracteres	Página de Códigos Coincidente	Definición
ISO8859-1	850	Western European Unix
ISO8859-2	852	Eastern European Unix
ISO8859-5	866	Russian Cyrillic Unix
KOI8-R	866	Alternate Russian Cyrillic Unix

Cuadro 8.6: Parámetros de Sistemas de Codificación Válidos con Samba 2.0

Juego de Caracteres	Definición
SJIS	Standard Shift JIS
JIS8	Código JIS 8-bits
J8BB	Código JIS 8-bits
J8BH	Código JIS 8-bits
J8@B	Código JIS 8-bits
J8@J	Código JIS 8-bits
J8@H	Código JIS 8-bits
JIS7	Código JIS 7-bits
J7BB	Código JIS 7-bits
J7BH	Código JIS 7-bits
J7@B	Código JIS 7-bits
J7@J	Código JIS 7-bits
J7@H	Código JIS 7-bits
JUNET	Códigos JUNET
JUBB	Códigos JUNET
JUBH	Códigos JUNET
JU@B	Códigos JUNET
JU@J	Códigos JUNET
JU@H	Códigos JUNET
EUC	Códigos EUC
HEX	Código Hexadecimal 3-bytes
CAP	Código Hexadecimal 3-bytes (Columbia Appletalk Program)

Cuadro 8.7: Opción de Configuración WinPopup.

Opción	Parámetro	Función	Defecto	Ambito
message command	string (ruta completamente cualificada)	Establece un comando a ejecutar en Unix cuando se recibe un mensaje emergente	ninguno	Global

```
[global]
  message command = /bin/csh -c 'xedit %s; rm %s' &
```

Advierte el uso de variables aquí. la variable %s contendrá el fichero que contiene el mensaje. Este fichero debería ser eliminado cuando el comando termine su ejecución; de lo contrario, tendrás una colección de mensajes en el servidor Samba. En adición, el comando debe poner en segundo plano su propio proceso (advierte el & tras el comando); de lo contrario el cliente puede suspender y esperar la notificación de que el comando ha sido satisfactoriamente enviado antes de continuar.

En adición a las variables standard, la Tabla 8.8 muestra tres variables únicas que puedes usar en un *message command*.

Cuadro 8.8: Variables para *message command*

Variable	Definición
%s	El nombre del fichero en el cual reside el mensaje.
%f	El nombre del cliente que envía el mensaje.
%t	El nombre de la máquina destinataria del mensaje.

8.5. Opciones Añadidas Recientemente.

Samba tiene varias opciones que han aparecido en el tiempo del desarrollo de Samba 2.0, pero que no están completamente soportadas. Sin embargo, te daremos una breve introducción a sus funcionalidades en esta sección. Estas opciones se muestran en la Tabla 8.9.

Cuadro 8.9: Opciones recientemente introducidas.

Opción	Parámetros	Función	Defecto	Ambito
change notify timeout	numérico (nº segundos)	Establece el intervalo entre chequeos cuando un cliente solicita esperar a un cambio en un directorio específico.	60	Global
machine password timeout	numérico (nº segundos)	Establece el intervalo de renovación de contraseñas para una máquina de dominio NT	604,800 (1 semana)	Global
stat cache	booleano	Si es *yes*, Samba cacheará mapas de nombres recientes.	yes	Global
stat cache size	numérico	Establece el tamaño de la caché de status.	50	Global

8.5.1. change notify timeout

Esta opción emula a una característica SMB de Windows NT llamada 'notificación de cambio'. esto permite a un cliente solicitar que un server Windows NT periódicamente monitoree un directorio específico de un recurso para controlar cualquier cambio. Si ocurre un cambio, el servidor lo notificará al cliente.

Desde la versión 2.0, Samba realizará esta función para sus clientes. Sin embargo, la realización de estos chequeos frecuentemte ralentiza el servidor considerablemente. Esta opción establece el período de tiempo que Samba debería esperar entre cada chequeo. El valor por defecto es un minuto (60 segundos); sin embargo, puedes usar esta opción para especificar un tiempo alternativo que Samba debería esperar entre chequeos:

```
[global]
  change notify timeout = 30
```

8.5.2. machine password timeout

La opción *machine password timeout* establece un período de retención para contraseñas de máquinas de dominio NT. El valor por defecto se establece actualmente al mismo período de tiempo que usa Windows NT 4.0: 604,800 segundos (una semana). Samba intentará periódicamente cambiar la contraseña de cuenta de máquina, la cual es una contraseña usada específicamente por otro servidor para repotarle cambios. Esta opción especifica el número de segundos que Samba debería esperar antes de intentar realizar el cambio de la password. El siguiente ejemplo lo cambia sólo un día, especificando lo siguiente:

```
[global]
  machine password timeout = 86400
```

8.5.3. stat cache

Esta opción global activa una caché de mapas de nombres no sensibles a mayúsculas recientes. El valor por defecto es *yes*. El equipo de Samba recomienda que nunca cambies este parámetro.

8.5.4. stat cache size

Esta opción establece el tamaño de las entradas de caché a ser usadas para la opción *stat cache*. El valor por defecto es 50. De nuevo, el equipo de Samba recomienda que nunca cambies este parámetro.

8.6. Otras Opciones.

Muchas opciones de Samba estan presentes para compatibilidad con características de sistemas Unix o Windows. Las opciones mostradas en la Tabla 8.10 resuelven específicamente algunos de los problemas conocidos. Normalmente no cambiaremos sus valores, y te aconsejamos hagas tú lo mismo.

8.6.1. deadtime

Esta oción global establece el número de minutos que Samba esperará para un cliente inactivo, antes de cerrar su sesión con el servidor Samba. Un cliente es considerado inactivo cuando no tiene ficheros abiertos y no hay datos enviandose desde él. El valor por defecto para esta opción es 0, lo que significa que Samba nunca cerrará ninguna conexión, no importa el tiempo que lleve inactiva. Puedes modificarlo así:

Cuadro 8.10: Otras opciones.

Opción	**Parámetros**	**Función**	**Defecto**	**Ambito**
deadtime	numérico (nº minutos)	Especifica el nº de minutos de inactividad antes de que una conexión debería ser terminada.	0	Global
dfree command	string (comando)	Usado para proporcionar un comando que retorna el espacio libre en disco en un formato reconocido por Samba.	ninguno	Global
fstype	NTFS, FAT, o Samba	Establece el tipo de sistema de ficheros reportado por el servidor al cliente.	NTFS	Global
keep alive	segundos	Establece el número de segundos entre chequeos para un cliente inoperativo.	0 (ninguno)	Global
max disk size	numérico (tamaño en MB)	Establece el mayor tamaño de disco para retornar a un cliente, aquellos de los cuales tienen límites. No afecta a operaciones actuales en el disco.	0 (infinito)	Global
max mux	numérico	Establece el número máximo de operaciones SMB simultáneas que los clientes pueden realizar.	50	Global
max open files	numérico	Número límite de ficheros abiertos para poner por debajo de los límites Unix.	10.000	Global
max xmit	numérico	Especifica el tamaño máximo de paquete que Samba enviará.	65.535	Global
nt pipe support	booleano	Desactiva una característica experimental de NT, para testeos o en caso de un error.	yes	Global
nt smb support	booleano	Desactiva una característica experimental de NT, para testeos o en caso de un error.	yes	Global
ole locking compatibility	booleano	Remapea peticiones bloqueadas fuera de rango usadas en Windows para ubicarlas en un rango admisible en Unix. Desactivarla provoca errores de bloqueos Unix.	yes	Global
panic action	comando	Programa a ejecutar si Samba cae; para depuración.	ninguno	Global
set directory	booleano	Si es *yes*, permite a los clientes VMS establecer comandos dir.	no	Global
smbrun	string (comando completamente cualificado)	Establece el comando que Samba usa como wrapper para comandos de shell.	ninguno	Global
status	booleano	Si es *yes*, permite a Samba monitorear el estado del comando smbstatus.	yes	Global
strict sync	booleano	Si es *no*, ignora las peticiones de aplicaciones Windows para realizar una sincronización a disco.	no	Global
sync always	booleano	Si es *yes*, fuerza a todos los clientes a guardar en disco antes de retornar de la llamada.	no	Global
strip dot	booleano	Si es *yes*, quita los puntos de nombres de ficheros Unix.	no	Global

```
[global]
  deadtime = 10
```

Esto le dice a Samba que termine cualquier sesión inactiva tras 10 minutos. Para la mayoría de las redes, establecer este valor funcionará bien, ya que las reconexiones desde elcleinte son realizadas generalmente de forma transparente al usuario.

8.6.2. dfree command

Esta opción global es usada en sistemas que determinan incorrectamente el espacio libre restante en el disco. Por ahora, el único sistema en el que está confirmado que necesita usar esta opción es Ultrix. no hay valor por defecto para esta opción, lo que significa que Samba ya cononce cómo computar el espacio libre de disco por sí mismo. Puedes modificarlo como sigue:

```
[global]
  dfree command = /usr/local/bin/dfree
```

Esta opción debería apuntar a un script que debería retornar el espacio total de disco en un bloque, y el número de bloques disponibles. La documentación de Samba recomienda usar el siguiente:

```
#!/bin/sh
df $1 | tail -1 | awk '{print $2" "$4}'
```

En máquinas System V, lo siguiente funcionará:

```
#!/bin/sh
/usr/bin/df $1 | tail -1 | awk '{print $3" "$5}'
```

8.6.3. fstype

Esta opción a nivel de recurso establece el tipo de sistema de ficheros que Samba reportará cuando le sea solicitado por un cliente. Hay tres cadenas que pueden ser usadas como valor para esta opción de configuración, como ves en la Tabla 8.11.

Cuadro 8.12: Tipos de Sistemas de Ficheros.

Variable	Definición
NTFS	Sistema de Ficheros Microsoft Windows NT
FAT	Sistema de Ficheros DOS FAT
Samba	Sistema de Ficheros Samba

El valor por defecto para esta opción es NTFS, que representa a un sistema de ficheros Windows NT. Probablemente no haya necesidad de especificar otro sistema de ficheros. Sin embargo, si quieres, puedes modificarlo:

```
[data]
  fstype = FAT
```

8.6.4. keep alive

Esta opción global especifica el número de segundos que Samba esperará entre envíos de paquetes tipo 'se mantiene activo' NetBIOS. Estos paquetes son usados para hacer ping a un cliente para detectar si está todavía activo y operativo en la red. El valor por defecto para esta opción es 0, lo que significa que Samba no enviará ningún paquete. Puedes modificarlo así:

```
[global]
  keep alive = 10
```

8.6.5. max disk size

Esta opción global especifica un límite ilusorio, en megabytes, para cada uno de los recursos que Samba está usando. Podrías usar esta opción para prevenir que clientes con viejos sistemas operativos procesaran grandes cantidades de disco, tales como más de un gigabyte.

El valor por defecto es 0, lo que significa que no hay limitación. Puedes modificarlo como sigue:

```
[global]
  max disk size = 1000
```

8.6.6. max mux

Esta opción especifica el número máximo de operaciones SMB concurrentes que Samba permite. El valor por defecto es 50. Puedes modificarlo así:

```
[global]
  max mux = 100
```

8.6.7. max open files

Esta opción especifica el número máximo de ficheros abiertos que Samba debería permitir en un momento dado para todos los procesos. Este valor debe ser igual o menor que la cantidad permitida por el sistema operativo, el cual varía de un sistema a otro. El valor por defecto es 10,000. Puedes modificarlo así:

```
[global]
  max open files = 8000
```

8.6.8. max xmit

Esta opción establece el tamaño máximo de los paquetes que Samba intercambia con un cliente. En algunos casos, establecer un tamaño menor puede incrementar el rendimiento, especialmente con Windows para Trabajo en Grupos. El valor por defecto es 65535. Y puedes modificarlo así:

```
[global]
  max xmit = 4096
```

La sección *B.2.2.6, La ventana de recepción TCP*, en el Apéndice B, *Afinando el Rendimiento de Samba*, muestra algunos usos para esta opción.

8.6.9. nt pipe support

Esta opción global es usada por desarrolladores para permitir o no a clientes Windows NT la capacidad de hacer conexiones a tuberías IPC$ específicas de NT SMB. Como usuario, nunca deberías necesitar cambiar el valor por defecto para esta opcion:

```
[global]
  nt pipe support = yes
```

8.6.10. nt smb support

Esta opción es usada por desarrolladores para negociar opciones SMB específicas de NT con clientes Windows NT. El equipo de Samba ha descubierto una mejora en el rendimiento estableciendo este valor a *no*. Sin embargo, como usuario, nunca deberías necesitar cambiar el valor por defecto para esta opcion:

```
[global]
  nt smb support = yes
```

8.6.11. ole locking compatibility

Esta opción desactiva la manipulación del bloqueo interno de rango de byte de Samba en ficheros, lo cual le da compatibilidad con aplicaciones *Object Linking and Embedding* (OLE) que usan bloqueos altos de rango de byte como método para procesos de comunicación. El valor por defecto es *yes*. Puedes cambiarlo así:

```
[global]
  ole locking compatibility = no
```

8.6.12. panic action

Esta opción especifica un comando a ejecutar en el caso de que Samba encuentre un error fatal cuando se inicie o esté en ejecución. No hay valor por defecto:

```
[global]
  panic action = /bin/csh -c
  'xedit < "¡Samba ha caído inesperadamente!'
```

8.6.13. set directory

Esta opción permite a clientes Digital Pathworks usar el comando *setdir* para cambiar directorios en el servidor. Si no estás usando el cliente Digital Pathworks, no necesitas cambiar esta opción. El valor por defecto es *no:*

```
[data]
  set directory = no
```

8.6.14. smbrun

Esta opción establece la localización del ejecutable *smbrun*, que usa Samba como wrapper para ejecutar comandos de shell. El valor por defecto para esta opción es automáticamente configurado por Samba cuando se compila. Si no instalaste Samba en el directorio de instalación standard, puedes especificar dónde se encuentra el binario de esta forma:

```
[global]
  smbrun = /usr/local/bin/smbrun
```

8.6.15. status

Esta opción indica si Samba debería registrar todas las conexiones activas en un fichero de status. Este fichero es uasdo sólo por el comando *smbstatus*. Si no tienes intenciones de usar este comando, puedes establecer esta opción a *no*, lo cual resulta en un pequeño incremento de velocidad en el servidor. El valor por defecto es *yes*:

```
[global]
  status = yes
```

8.6.16. strict sync

Esta opción determina si Samba admitirá todas las peticiones de realizar una sincronización de disco cuando se solicite por cualquier cliente. Muchos clientes solicitan una sincronización de disco cuando van a intentar pasar datos a sus propios ficheros abiertos. Como resultado, esto puede ralentizar sustancialmente el servidor Samba. El valor por defecto es *no:*

```
[data]
  strict sync = no
```

8.6.17. sync always

Esta ocpión decide si cada escritura a disco debería ser seguida de una sincronización de disco antes de que la llamada de escritura retorne el control al cliente. Aunque el valor para esta opción es no, los clientes pueden solicitar una sincronización de disco; mira la opción *strict sync*. El valor por defecto es *no*. Y lo puedes cambiar así:

```
[data]
  sync always = yes
```

8.6.18. strip dot

Esta opción determina si eliminar el punto de nombres de fichero Unix que están formateados con un punto al final. El valor por defecto para esta opción es *no*. Puedes cambiarlo así:

```
[global]
  strip dot = yes
```

Esta opción está considerada obsoleta; el usuario debería usar en su lugar la opción *mangled map*.

8.7. Copias de Seguridad (Backups) con smbtar

Nuestro tópico final en este capítulo es la herramienta *smbtar*. Un problema común en los modernos PCs es que las unidades de disquete e incluso las grabadoras de CDROM son frecuentemente demasiado pequeñas para ser usadas como sistemas de copia de seguridad. Y además, comprar una unidad de cinta para cada equipo sería algo poco inteligente. Consecuentemente, muchos sitios no mantienen copias de seguridad de los contenidos de sus PCs. En su lugar, los reinstalan usando unidades de disquete y CDROMs cuando fallan.

Por suerte, Samba nos proporciona otra opción: puedes hacer copia de seguridad de tus PCs usando la herramienta *smbtar*. Esto puede hacerse periódicamente, si mantienes los datos de los usuarios en tu sistema Samba, o sólo ocasionalmente, para almacenar las aplicaciones y ficheros de configuración locales, para poder hacer reparaciones y reinstalaciones rápidamente.

Para hacer copias de seguridad de PCs desde un servidor Unix, necesitas hacer tres cosas:

Asegurarte de que la opción 'Compartir Archivos e Impresoras' está instalado y activado, y que aparece enlazado al protocolo TCP/IP.

Compartir explícitamente un disco en el PC, de manera que pueda ser leído desde el servidor.

Usaremos Windows 95/98 para ilustrar los dos primeros pasos. Ve al icono de *Red* en la ventana del *Panel de Control*, y comprueba que la *Compartición de Archivos e Impresoras para Redes Microsoft* está actualmente listada al principio de la ventana, como se muestra en la Figura 8.2.

Si *'Compartir Archivos e Impresoras para Redes Microsoft'* no está instalado, puedes instalarlo pulsando sobre el botón *Añadir* del panel de *Red*. Tras pulsarlo, se te preguntará qué servicio quieres añadir. Selecciona *Servicio* y continuar; se te preguntará por vendedor y servicio a instalar. Finalmente, selecciona *'Compartir Archivos e Impresoras para Redes Microsoft'*, y pulsa el botón *Hecho* para instalar el servicio.

Una vez lo hayas instalado, retorna al panel de Red y selecciona el protocolo TCP/IP que está definido para tu adaptador de red. Entonces, haz click sobre el botón *Propiedades* y selecciona 'Enlaces' sobre el botón de *Propiedades*, y selecciona la pestaña *Enlaces*. Deberías ver una caja de diálogo similar a la de la Figura 8.3. Aquí, necesitarás verificar que el cuadro *'Compartir Archivos e Impresoras para Redes Microsoft' está marcado*, dándole acceso a TCP/IP. En este punto ya puedes compartir discos con otras máquinas de la red.

El siguiente paso es compartir el disco que quieras salvaguardar en la unidad de cinta del servidor. Vete a *'Mi PC'* y selecciona, por ejemplo, la carpeta *Mis Documentos*. Haz click con el botón derecho del ratón y selecciona *Propiedades*. Esto debería mostrarte la ventana que aparece en la Figura 8.4.

Selecciona la pestaña *Compartir* y activa la compartición. Ahora tienes la opción de compartir el disco en sólo lectura, lectura-escritura (Full), o ninguno, cada uno con una cotraseña distinta. Esta es la versión de Windows 95/98, la cual proporciona sólo protección a nivel de recurso. En este ejemplo, hemos hecho permiso de lectura-escritura y establecido una contraseña, como se muestra en la Figura 8.5. Cuando introduzcas la contraseña y pulses OK, se te pedirá que la confirmes. Tras esto, habrás terminado con el segundo paso.

Finalmente, el último paso es configurar un script de realización de la copia de seguridad sobre el servidor de cintas, usando el programa smbtar. El más simple de los posibles contiene una única línea y sería algo así:

Figura 8.1: La aplicación WinPopup.

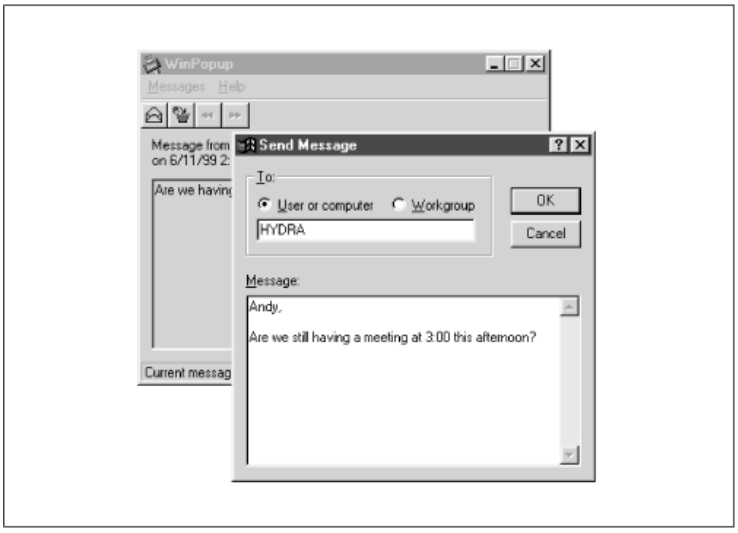


Figura 8.2: Ventana de Entorno de Red.

Figura 8.3: Enlaces TCP/IP.

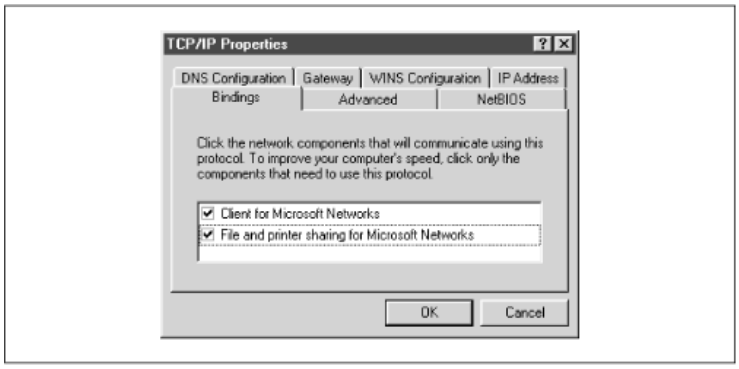
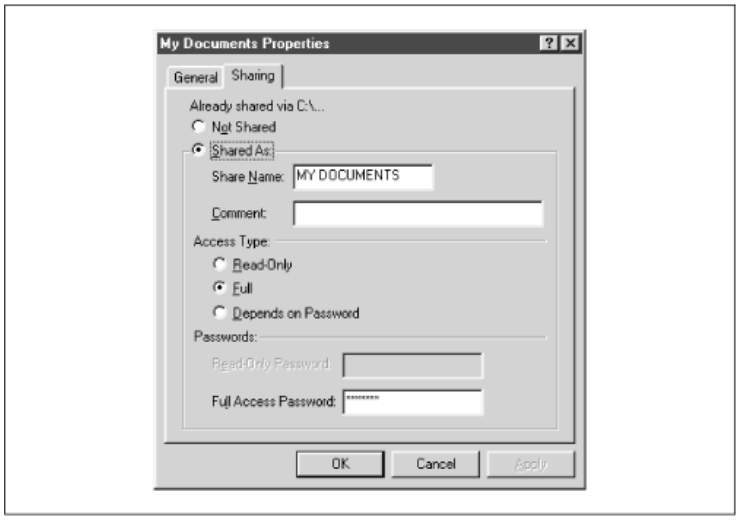


Figura 8.4: Propiedades de Mis Documentos.



```
smbtar -s client -t /dev/rst0 -x "My Documents" -p
```

password

Esto hace copia de seguridad del recurso //cliente/Mis Documentos al dispositivo /dev/rst0. Por supuesto, esto es excesivamente simple y bastante inseguro. Lo que quieras hacer dependerá de tu sistema de copias.

Sin embargo, para abrirte el apetito, aquí tienes algunas posibilidades de uso de smbtar:

Hacer Copias Incrementales usando el bit de archivo DOS (la opción -i). Esto requiere que el cliente sea accedido en modo lectura-escritura, para que el bit pueda ser eliminado por *smbtar.*

Hace Copias sólo de aquellos ficheros que hayan cambiado desde una fecha determinada (usando la opción *-N nombre_fichero*).

Copiar discos enteros, compartiendo todo C: o D:, por ejemplo.

Excepto para el primer ejemplo, cada uno de ellos puede ser ejecutado compartiendo los recursos en modo *sólo lectura*, reduciendo el riesgo de seguridad de tener contraseñas en scripts y pasándolos por la línea de comandos.

Figura 8.5: Propiedades de MyFiles como Recurso.