INTRODUCCIÓN

El presente trabajo pretende ser una breve introducción al paquete Samba, centrada en la última versión disponible hasta la fecha -samba2.0.6-, así como una breve guía de instalación y configuración para pequeños entornos en los que se realizan las operaciones más comunes que implementa el protocolo SMB. Por último, al final de este trabajo se incluye una referencia bibliográfica de las fuentes usadas, a donde es recomendable acudir en caso de buscar más información o bien para realizar una configuración de la aplicación en un caso muy específico.

1. OBJETIVOS Y DESCRIPCIÓN DEL SERVICIO.

La existencia de redes de ordenadores heterogéneas con Windows y Linux son cada vez más frecuentes. Por ello surge la clara necesidad de establecer protocolos para que equipos con diferentes sistemas operativos puedan compartir recursos. SAMBA es una buena solución para ello.

En la actualidad son cada vez más las empresas y organizaciones que mediante una red local, LAN (Local Area Network), mantienen conectados sus equipos con el fin de compartir hardware y software. Dentro de estos sistemas, por la potencia y seguridad que ofrecen, son necesarios equipos con sistemas operativos tipo UNIX (SunOS, Linux, etc.); y por la facilidad de uso y por el numerosos software de gestión disponible, equipos con sistemas operativos de Microsoft (Windows 95, 98, NT, 2000). Esto hace que deban coexistir en una misma red equipos con sistemas operativos diferentes.

Cuando se producen este tipo de escenarios se plantea un problema:)es posible hacer que equipos con sistemas operativos diferentes compartan recursos? La respuesta a esta pregunta es afirmativa y existen diferentes caminos para ello. EN este artículo se presenta SAMBA, un paquete con capacidad para que equipos con sistema operativo tipo UNIX Ase entiendan≅ con equipos con sistema operativo tipo Windows.

El paquete SAMBA no está únicamente limitado al dúo UNIX-Windows, sino que permite la utilización de otros sistemas operativos, como OS/2, VMS, NetWare, entre otros. Este artículo se centrará, únicamente, en el dúo anterior, concretamente en máquinas con Linux y Windows 9x. A pesar de ello, no es un problema utilizar SAMBA con otros sistemas operativos una vez se hayan adquirido los conocimientos que se desprenden de este artículo.

Como ya se comentaba en la introducción los objetivos de este trabajo son los de proporcionar la información necesaria para instalar, configurar y utilizar los servicios del protocolo Session Message Block (Bloque de Sesión de Mensajes, SMB o Samba) bajo Linux. Con Samba se puede:

- Compartir un sistema de archivos de Linux con Windows 95, 98, o NT.
- Compartir un sistema de archivos de Windows 95, 98, o NT con Linux.
- Compartir una impresora conectada al sistema Linux con los sistemas Windows 95, 98, o NT.
- Compartir una impresora de Windows 95, 98, o NT con Linux.

Microsoft e Intel desarrollaron el sistema de protocolo SMB en 1987, y posteriormente el australiano, Andrew Tridgell lo integró a los distintos sistemas UNIX y después a Linux.

En resumen, usando samba, se puede convertir cualquier máquina UNIX en un potente servidor de ficheros e impresión para máquinas DOS y Windows. Desde sus inicios en 1991, Samba ha sido mejorado para ser un producto estable y manejable que está hecho con el motivo de que en las compañías y servidores como suplemento o incluso remplazo de servidores NetWare o Windows NT.

NOTA:

En la actualidad, Microsoft propone otro estándar para la compartición de archivos, llamado Common Internet File System (Sistema Común de Archivos de Internet, CIFS). Este estándar se ha presentado ante el Grupo de Trabajo Especial de Ingeniería de Internet (Internet Engineering Task Force) pero CIFS todavía no se ha adoptado de forma generalizada, y menos aún por la comunidad de desarrollo de Linux.

2. REQUISITOS.

Los requisitos del paquete samba son muy simples, por ser su finalidad muy específica; simplemente necesitamos un sistema que reúna las características comentadas en el punto anterior: que necesite compartir recursos entre dos entornos distintos -normalmente Windows y Unix/Linux-.

Hoy en día hay pocos clientes SMB disponibles, sin embargo, hay muchos servidores disponibles. Los principales clientes son los derivados de Microsoft, incluidos en Windows 95/98 y NT/2000. Éstos son visibles cuando se utiliza el Explorador de Windows para acceder a un recurso -directorio o impresora- de un servidor SMB, y también cuando se abren ficheros usando un UNC (Universal Naming Convention). Otros clientes SMB son:

smbclient de Samba y smbfs para Linux.

Las implementaciones de servidores de SMB son más numerosas: Samba, Microsoft Windows 95/98/NT/2000, PathWorks family de Digital, LAN Manager para OS/2,SCO,etc.., VisionFS de SCO, TotalNET Advanced Server de Syntax, Adavanced Server for UNIX de AT&T y LAN Server para OS/2 de IBM.

En nuestro caso nos centraremos en los que es el paquete samba bajo Linux, además de algunos aspectos de carácter general en la conectividad con sistemas Windows.

3. ARQUITECTURA DEL SERVICIO.

En primer lugar nos describiremos el protocolo SMB, así como su arquitectura de protocolos sobre la que se apoya, para luego pasar a la descripción del paquete samba.

3.1. EL PROTOCOLO SMB (Server Message Block).

Los conceptos de PCs y UNIX difieren notablemente, es por ello que debemos de echar un vistazo a la arquitectura de los PCs con NetBIOS.

3.1.1. NetBIOS

NetBIOS es una interface software (API) que ha sido diseñada para la comunicación entre máquinas. De ello que se le dé el nombre de servicio. Ésta permite a las máquinas que se conectan a la red, reservar nombres para ellas mismas. Después de la reserva, estas máquinas pueden ser accesibles a través de sus nombres. Cualquier máquina en la red puede reservar cuantos nombres quiera siempre y cuando el nombre no esté actualmente en uso.

Esta arquitectura dinámica tuvo su origen en las redes de PC donde la instalación de un nuevo nodo de red debería hacerse tan sencilla como fuese posible. Así configurar una máquina fue reducido a dale un nombre. Los problemas de nombres únicos, máxima longitud de los nombres limitada a 16 caracteres, no tienen relevante importancia si las redes no son muy extensas.

Además del servicio de nombres hay servicios de comunicación. Hay flujos de datos seguros e inseguros. Estos pueden ser comparables a los protocolos TCP y UDP en el mundo UNIX. Los altos protocolos como el protocolo SMB están en un nivel por encima de éstos.

La interfaz NetBIOS puede ser implementada por diferentes arquitecturas de red. Una implementación trabaja relativamente cercana con el hardware de red y se llama NetBEUI, la cual es habitualmente referida también como NetBIOS.

Para direccionar, los simples paquetes NetBEUI trabajan junto con las direcciones hardware del adaptador. Contrariamente a IPX o direcciones IP no se puede extraer información de enroutado de ella. Así pues, no es posible transferir paquetes NetBEUI vía routers y así el encaminamiento de una red NetBEUI está reducido al rango que se pueda conseguir con repetidores y bridges.

Otros protocolos de red que han sido implementados con NetBIOS son IPX de Novell y TCP/IP. El protocolo que sostiene a NetBIOS sobre TCP/IP se describe en la RFC1001 y RFC1002. Rfc1001, además, contiene una introducción en los conceptos de NetBIOS buena y fácil de entender que ayuda mientras intenta entender servicios como WINS. (No es otra cosa que una extensión del servidor de nombres NetBIOS y no una idea de Microsoft -aunque parezca por el nombre-).

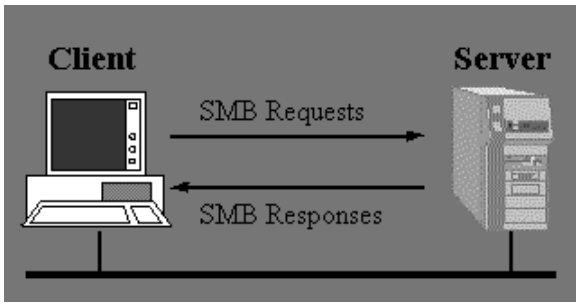
Los nombres NetBIOS que se dan a la vía exterior TCP/IP no tienen nada en común con los nombres de /etc/hosts o los dados vía DNS. NetBIOS tiene un espacio de nombres propio. Bien, es recomendable usar nombres que se correspondan a los hostnames DNS para hacer una administración más sencilla. Este es el uso por defecto de Samba.

3.1.2. SMB

El protocolo SMB (Server Message Block) sirve para trabajar con servicios de ficheros e impresión en los Windows y LAN Managers disponibles, el protocolo en realidad define como recursos que se pueden compartir con SMB: ficheros, impresoras, puertos serie, tuberías con nombre y mail slots. Entonces el protocolo SMB se basa en los servicios NetBIOS, luego es comparable a NFS. Aquí no hay diferencia con otros protocolos como NetWare Core Protocol. Sorprendentemente Microsoft ha hecho públicas las especificaciones del protocolo SMB así que otros pueden ahora soportar SMB también.

El primer documento escrito sobre el protocolo SMB es de IBM del año 1985 (IBM Personal Computer

Seminar Proceedings - Mayo de 1985), en donde lo nombra como el IBM PC Networks SMB Protocol. El siguiente documento corresponde a Microsoft/Intel y se llama Microsoft Networks/OpenNET-FILE SHARING PROTOCOL de 1987. Después de éste, el protocolo fue desarrollado por Microsoft y otros, existiendo gran cantidad de documentos que definen el protocolo SMB en ftp.microsoft.com (SMB documentation area).



SMB es un protocolo cliente-servidor basado en request-response (petición-respuesta). La única excepción a la naturaleza request-response del protocolo es cuando el cliente tiene una petición opportunistic locks (oplocks) y el servidor subsecuentemente tiene que romper un oplock garantizado pues otro cliente tiene una petición de abrir un fichero con un modo que es incompatible con el oplock garantizado. En este caso, el servidor envía un mensaje -no solicitado- al cliente señalándole que el oplock está roto.

Microsoft ha extendido el protocolo SMB recientemente y lo ha renombrado como CIFS (Common Internet File System). Microsoft ahora intenta establecer CIFS como el estándar internet para ganar más soporte para este protocolo. (Para más información sobre las especificaciones CIFS echar un vistazo a ftp://ftp.microsoft.com/developr/drg/CIFS/cifs; en /developr/drg/CIFS están las especificaciones viejas de SMB.)

Samba es un servidor que implementa el protocolo SMB bajo UNIX. Samba puede convertir cualquier máquina UNIX en un servidor de ficheros e impresión para la mayoría de los sistemas operativos de PC. Samba ha sido portado a otros sistemas operativos: corre sobre QNX en tiempo real, así como sobre un Cray; Novell ha portado Samba a NetWare 4.1 y le ha denominado AMigration Toolkit≅ facilitando un sentido sencillo para que los usuarios de Windows NT migrasen a NetWare.

3.1.3. Clientes

Los clientes se suelen conectar a los servidores usando TCP/IP (NetBIOS sobre TCP/IP está especificado en RFC1001 y RFC1002), NetBEUI o IPX/SPX.

Excepto para DOS y Windows 3.1 cualquier sistema operativo de PC actual soporta el protocolo SMB para importar y exportar. Windows para trabajo en grupo también soporta SMB; en su instalación estándar solo vía IPX y NetBEUI. Para usar Samba que solo suministra SMB vía TCP/IP se necesita instalar un software adicional a obtener de ftp://ftp.microsoft.com/bussys/Clients.

Una vez establecida una conexión, los clientes pueden enviar comandos (SMBs) al servidor para que les permita tener acceso a los compartidos, ficheros abiertos, ficheros de lectura y escritura, y en general cualquier cosa que se haría con un sistema de ficheros, que en el caso de SMB está sobre la red.

El siguiente diagrama muestra los protocolos sobre los que puede correr SMB:

<table>
<tr><td>OSI</td><td colspan="4"></td><td>TCP/IP</td></tr>
<tr><td>Application
Presentation</td><td colspan="4">SMB</td><td rowspan="2">Application</td></tr>
<tr><td>Session</td><td>NetBIOS</td><td rowspan="2">NetBEUI</td><td>NetBIOS</td><td>NetBIOS</td></tr>
<tr><td>Transport</td><td rowspan="2">IPX[1]</td><td rowspan="2">DECnet</td><td>TCP&UDP</td><td>TCP/UDP</td></tr>
<tr><td>Network</td><td>IP</td><td>IP</td></tr>
<tr><td rowspan="2">Link</td><td rowspan="2">802.2,
802.3,802.5</td><td>802.2</td><td rowspan="2">Ethernet V2</td><td rowspan="2">Ethernet V2</td><td rowspan="2">Ethernet or
others</td></tr>
<tr><td>802.3,802.5</td></tr>
<tr><td>Physical</td><td colspan="4"></td></tr>
</table>

Los servidores SMB proporcionan un espacio hardware a sus clientes bajo el significado de así llamado compartido. Aquí un share (compartido) incluye un directorio y sus subdirectorios. Se exporta por el nombre y puede ser accesible por su nombre. Por supuesto el nombre compartido (Share name) puede ser fijado a cualquier nombre. No tiene porque ser el nombre del directorio exportado. A una impresora también se le asigna un nombre; los clientes pueden acceder a la impresora por su nombre.

3.1.4. Permisos

Un servidor NFS se configura vía /etc/exports. Restricciones adicionales son posibles solamente en una máquina sola. En NFS esto tiene sentido puesto que fue designado para workstations UNIX y ellas mismas comprueban los permisos y autentificaciones. Hoy en día en tiempos de Windows y OS donde cualquier usuario puede asumir permisos de Aroot≅ el protocolo NFS no sería adecuado. Los clientes NFS para DOS deben ser vigilados por sus inmensos agujeros de seguridad.

El protocolo SMB viene del mundo DOS y cubre directamente los asuntos de seguridad. Cualquier acceso a un share (compartido), puede ser protegido por un password. SMB ahora tiene dos alternativas para proveer esto:

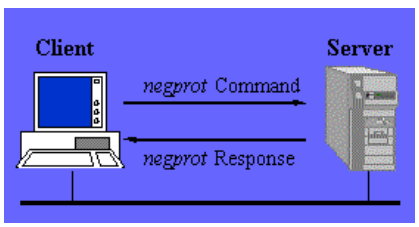
- Share Level Security: Un password se asigna directamente a cualquier Ashare≅. Cualquiera que conozca el password puede acceder al Ashare≅.
- User Level Security: Esta variante introduce el concepto de usuario en SMB. Cada usuario tiene que hacer login en el servidor usando su login y su password. Después de un login exitoso el servidor puede ahora dar acceso dependiendo de los permisos del usuario.

La distinción entre Share Level Secutity y User Level Security ha de ser fijada por el servidor como un todo. No es posible exportar algunos shares como share level security, mientras que otros son exportados como user level security.

Por defecto Samba se fija a share level security. Aquí los directorios raíz (home) de los usuarios son protegidos por el user password normal. Para otros shares, un usuario tiene que entrar dando la opción de usuario, cuyo password se usa como protección. El parámetro **security** puede ser dado como **security=user** en smb.conf. Entonces los usuarios son validados como habitualmente se hace bajo UNIX, con /etc/passwd y /etc/group. Samba ofrece también una tercera posibilidad de proporcionar **security = server**. Si esta opción está activada Samba valida al usuario en otro servidor (NT) que ha sido previamente fijado por la opción **password server**.

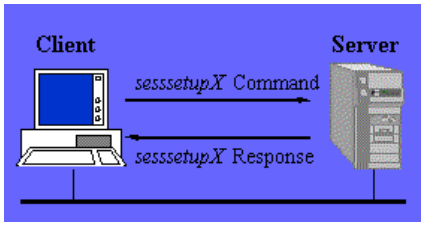
3.1.5. Un ejemplo de transferencia SMB.

Los elementos del protocolo (peticiones y repuestas) que clientes y servidores intercambian se llaman SMBs. Tienen un formato específico que es muy similar para ambos. Cada operación (petición o respuesta) está formada por una cabecera de tamaño fijo, seguida de un parámetro de tamaño variable, así como de un fragmento de datos.



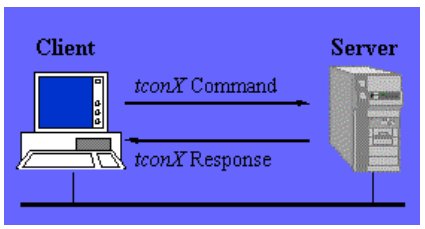
Después de la conexión al nivel de NetBIOS, el cliente está preparado para pedir servicios al servidor. Sin embargo, el cliente y el servidor deberán primero identificar que variante del protocolo entiende cada uno.

El cliente envía un *negprot SMB* al servidor, listando las variantes del protocolo que él entiende. El servidor responde con el índice de la variante que quiere usar o bien com 0xFFFF si ninguna de las variantes es aceptada.



Variantes posteriores de Core y CorePlus -variantes de SMB-, incluyen en la respuesta negprot indicaciones de máximo tamaño de buffer, nombres de ficheros canónicos, etc..

Una vez establecido el protocolo. El cliente procede a hacer logon al servidor, si se requiere. Esto se hace con un *sesssetupx SMB*. La respuesta indica si el par username,password es válido. Uno de los aspectos más importantes de la respuesta es el UID del usuario. Este UID deberá ser enviado con cada SMBs siguiente sobre esa conexión y en ese servidor.



Una vez hecho el logon, el cliente se conecta a un árbol.

El cliente envía un *tcon* o *tconX SMB* especificando el nombre de red del share al que se quiere conectar. Si todo ha ido bien, el servidor responde con un TID que el cliente usará en todos los futuros SMBs referidos a ese share.

3.2. SAMBA.

SAMBA es un conjunto de utilidades para Linux mantenida por Samba-Team (http://samba.org) que permite compartir y utilizar recursos entre máquinas Linux y Windows. Antes de comenzar a hablar sobre su instalación y funcionamiento presentaremos una breve reseña histórica.

En los primeros días del PC, IBM desarrolló el protocolo NetBIOS (Network Basic Input Output System), que servía de interfaz entre el software y el hardware de red. Posteriormente desarrolló el NetBEUI (NetBIOS Enhanced User Interface), que permitía el paso de paquetes NetBIOS a redes Ethernet o Token Ring. Otros protocolos para este fin son: DECnet, IPX/SPX y TCP/IP. Microsoft incorporó en su sistema el protocolo NetBEUI, también conocido como protocolo SMB (Server Message Block) o LanManager y, actualmente, como CIFS (Common Internet File System). Este protocolo es el utilizado en máquinas Windows para compartir directorios e impresoras. En aquel tiempo, Andrew Tridgell disponía de un servidor UNIX y unos PCs con sistemas operativos de Microsoft. Tenía el problema de compartir dispositivos entre estas máquinas y, por suerte para todos nosotros, decidió desarrollar el protocolo SMB en máquinas UNIX, creando el paquete que se conoce como SAMBA.

El conjunto de utilidades que componen SAMBA tiene como finalidad implementar, para Linux en particular y UNIX en general, el protocolo SMB. En concreto, cubre cuatro de los servicios básicos del protocolo CIFS, como son: - Servicios de ficheros e impresión.

- Autentificación y autorización de usuarios.
- Resolución de nombres.
- Presentación de servicios.

El servicio de ficheros e impresión va a permitir compartir estos recursos entre máquinas Linux y Windows, mientras que la autentificación y autorización permite la identificación de usuarios y así otorgarle o denegarle el recurso. La resolución de nombres y la presentación de servicios permitirán a los usuarios conocer qué recursos se están compartiendo. El paquete SAMBA está compuesto, principalmente, por dos programas continuamente activos (demonios), conocidos como: el demonio de implementación del protocolo SMB, *smbd*, y el demonio para el servidor de nombres, *nmbd*. Como puede deducirse por su nombre, el primero de los demonios se ocupa de las funciones de servicio de ficheros e impresión y autentificación y autorización de usuarios; y el segundo de las funciones de resolución de nombres y presentación de servicios. Además, el paquete incluye un conjunto de utilidades que completan la funcionalidad del mismo. El cuadro muestra de forma resumida los componentes de SAMBA:

COMPONENTES DEL PAQUETE SAMBA:	
smbd	demonio que permite el uso compartido de ficheros e impresoras entre máquina Linux y Windows
nmbd	demonio que permite buscar y usar elementos compartidos entre máquinas Linux y Windows
smbclient	utilidad que implementa un cliente Linux para utilizar recursos de máquinas Windows
testparm	utilidad para comprobar el fichero de configuración de SAMBA smb.conf
testprns	utilidad que permite comprobar las impresoras definidas en el fichero /etc/printcap
smbstatus	utilidad para listar las conexiones actuales registradas por smbd
nmblookup	utilidad que permite la consulta de recursos compartidos desde una máquina Linux
make_smbcodepage	utilidad que permite definir un fichero de definición de códigos de página para smbd
	utilidad que permite el cambio de clave de encriptado sobre SAMBA y Windows

smbpasswd	NT
swat	utilidad para crear el fichero de configuración smb.conf mediante un navegador

4. CONFIGURACIÓN.

En este apartado daremos las pautas para configurar el servicio. Nos centraremos como ya hemos comentado en un servidor samba bajo Linux y clientes bajo Windows 95/NT.

4.1. INSTALACIÓN DEL SERVIDOR.

4.1.1. Instalación de samba en Linux.

La instalación de SAMBA no es un proceso difícil y basta tener conocimientos mínimos sobre instalación de aplicaciones en Linux para realizar dicha tarea. De todas formas, es necesario conocer antes de emprender la instalación que en la mayoría de las distribuciones el paquete SAMBA viene incorporado y, por lo tanto, es posible seleccionarlo durante la instalación de Linux. Si hubiese sido así ya tendrás disponible SAMBA en tu máquina. Si no lo seleccionaste durante la instalación de Linux, o necesitas actualizar la versión que dispones no te preocupes. Lo primero que necesitamos para poder instalar SAMBA son los fuentes del mismo. Los fuentes de la aplicación se pueden obtener en Internet en la dirección *http://samba.org/samba* en alguno de sus *mirrors*. Suele ser un fichero *samba-lastest.tar.gz*.

Si ya disponemos del fichero necesitamos descomprimirlo ejecutando la orden:

tar xvfz samba-lastest.tar.gz

Esto creará un directorio *samba-X.Y.Z*, donde *X.Y.Z* es la versión que se ha descomprimido. Por ejemplo, si se descomprime la versión 2.0.5a el directorio creado será *samba-2.0.5*. A partir de aquí supondremos que utilizamos este directorio. Si accedemos a él podremos observar que contiene una serie de directorios entre los que hay que destacar:

- *docs*: documentación sobre SAMBA en varios formatos, entre otros texto plano (textdocs/), HTML (htmldocs/), páginas man (manpages/).
- *examples*: diferentes ejemplos sobre configuración para *smb.conf* e impresión.
- *packaging*: paquetes compilados para diversas distribuciones.
- *source*: código fuente de SAMBA.

Los documentos incluidos en *docs/* son una gran fuente de información para algunos aspectos que no hayan sido abordados aquí con la suficiente profundidad, o que incluso no hayan sido tratados. El directorio de ejemplos *examples/* pretende servir de base para las diferentes configuraciones del fichero *smb.conf* incluyendo ejemplos con las situaciones más características. Los paquetes en *packaging/* permitirán instalar SAMBA sin necesidad de compilar.

Así, antes de comenzar la compilación de los fuentes sería aconsejable acceder al directorio *packaging/* y comprobar si en él se encuentran los paquetes de tu distribución. Si fuese así, sólo necesitas de la utilidad de instalación de paquetes de tu distribución para disponer de SAMBA. Si el paquete para tu distribución no está disponible no es un problema, sólo será necesario compilar los fuentes incluidos en el directorio *source/*. Los pasos a seguir son:

- Cambiar al directorio samba-2.0.5/source: cd samba-2.0.5/source
- Configurar el paquete para tu sistema: ./configure

  Esta orden permitirá generar el fichero de proyecto (*Makefile*) para SAMBA con la configuración apropiada para tu sistema. La orden *configure* puede ir acompañada de un grupo de parámetros, que inicialmente te aconsejamos no utilizar. Puedes obtener información sobre los parámetros y su sintaxis tecleando la orden *./configure -help*. Una vez configurado el sistema se habrá generado el fichero *Makefile*, que también podrá ser alterado para establecer diferentes directorios de instalación y otros elementos. Al igual que en los parámetros de configuración no aconsejamos cambios en el fichero a menos que seas un usuario experimentado. A lo largo de este artículo se supondrá que no se han realizado cambios en este fichero.
- Compilar: *make*

  Compilará el sistema, utilizando el fichero *Makefile* anterior, generando los demonios y utilidades.
- Instalar: *make install*

Mediante esta orden se instalará SAMBA en el directorio */usr/local/samba*, quedando una estructura de directorios como esta:

/usr/local/samba/

/----------	**bin/**	Ejecutables
/----------	**lib/**	Fichero smb.conf
/----------	**man/**	Páginas del manual
/----------	**private/**	Privado
/----------	**swat/**	Configuración con WEB
/----------	**var/**	Fichero .log

Es necesario aclarar que durante la instalación el fichero *smb.conf* no se incluye en el directorio *lib/*, por lo que tendremos que crearlo. Este aspecto lo abordaremos en el siguiente punto.

4.1.2. Fichero de configuración ASMB.CONF≅

El siguiente paso después de la instalación es la creación del fichero de configuración de SAMBA *smb.conf*. Este fichero almacenará la información necesaria para que los demonios *smbd* y *nmbd* establezcan la configuración para compartir directorios e impresoras de nuestro sistema Linux con otras máquinas Windows.

Como comentamos en el apartado anterior el fichero *smb.conf* no es generado durante el proceso de instalación, por lo que será necesario crearlo. La ubicación para el fichero será */usr/local/samba/lib*. Si se modificó el fichero *Makefile* antes de la compilación, para alojar este fichero en otro directorio tendremos que crearlo en ese lugar (otra ubicación habitual es */etc/smb.conf*).

La estructura del fichero consiste en un conjunto de secciones que incluyen un conjunto de parámetros a los que se les asigna un determinado valor, con el esquema siguiente:

```
[sección1]
par1 = val1
par2 = val2
...
[sección2]
par3 = val3
...
```

Cada sección dentro del fichero, excepto [global] es utilizada para especificar recursos compartidos por la máquina Linux. El nombre de la sección es el nombre del recurso compartido y los parámetros especifican sus atributos. Por ejemplo, la siguiente sección comparte un directorio llamado */tmp/compartir* dándolo por nombre al recurso *tmpcomp* asignando permiso de escritura.

```
[tmpcomp]
path = /tmp/compartir
writeable = 1
```

Existen tres secciones especiales:

- *[global]*: permite definir parámetros comunes o por defecto para todo el sistema.
- *[homes]*: permite facilitar la conexión de cada usuario del sistema a su directorio *$HOME*.
- *[printers]*: permite el uso de impresoras a los usuarios del sistema.

En cuanto a los parámetros que pueden incluirse en cada sección hay que aclarar que algunos sólo podrán aparecer en la sección *[global]*, mientras que existen otros, que pueden aparecer en cualquier sección, de tal forma que si aparecen en una sección que comparte un recurso afectará sólo a ese recurso, pero si aparece en la sección *[global]* afectará a todos los recursos.

Para usuarios impacientes, en el directorio *samba-2.0.5/examples* existen un gran número de ejemplos de ficheros de configuración. En concreto, en el directorio *samba-2.0.5/examples/simple* existe un fichero de configuración básico que servirá como el primer fichero de configuración. Si decide utilizar uno de estos ficheros cópielo al directorio */usr/local/samba/lib*.

Listado: Ejemplo del archivo de configuración de Samba smb.conf

```
; The global settings for a RedHat default install
; smbd re-reads this file regularly, but if in doubt stop and restart it:
; /etc/rc.d/init.d/smb stop
; /etc/rc.d/init.d/smb start
;====================== Global Settings ================================
[global]

; workgroup = NT-Domain-Name or Workgroup-Name, eg: REDHAT4
   workgroup = WORKGROUP
```

```
; comment is the equivalent of the NT Description field
   comment = RedHat Samba Server

; volume = used to emulate a CDRom label (can be set on a per share basis)
   volume = RedHat4


; printing = BSD or SYSV or AIX, etc.
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes

; Uncomment this if you want a guest account
; guest account = pcguest
   log file = /var/log/samba-log.%m
; Put a capping on the size of the log files (in Kb)
   max log size = 50

; Options for handling file name case sensitivity and / or preservation
; Case Sensivity breaks mana WfW and Win95 apps
;   case sensitive = yes
    short preserve case = yes
    preserve case = yes

; Security and file integrity related options
   lock directory = /var/lock/samba
   locking = yes
   strict locking = yes
;   fake oplocks = yes
   share modes = yes
; Security modes: USER uses Unix username/passwd, SHARE uses WfW type
; passwords
;        SERVER uses a Windows NT Server to provide authentication
; services
   security = user
; Use password server option only with security = server
;   password server = <NT-Server-Name>

; Configuration Options*****Watch location in smb.conf for side-effects*****
; Where %m is any SMBName (machine name, or computer name) for which a
; custom configuration is desired
;   include = /etc/smb.conf.%m

; Performance Related Options
; Before setting socket options read the smb.conf man page!!
   socket options = TCP_NODELAY
; Socket Address is used to specify which socket Samba
; will listen on (good for aliased systems)
;   socket address = aaa.bbb.ccc.ddd
; Use keep alive only if really needed!!!!
;   keep alive = 60

; Domain Control Options
; OS Level gives Samba the power to rule the roost. Windows NT = 32
;        Any value < 32 means NT wins as Master Browser, > 32 Samba gets it
;   os level = 33
; specifies Samba to be the Domain Master Browser
;   domain master = yes
; Use with care only if you have an NT server on your network that has been
; configured at install time to be a primary domain controller.
;   domain controller = <NT-Domain-Controller-SMBName>
; Domain logon control can be a good thing! See [netlogon] share section
; below!
;   domain logons = yes
; run a specific logon batch file por workstation (machine)
;   logon script = %m.bat
; run a specific logon batch file per username
;   logon script = %u.bat
; Windows Internet Name Serving Support Section
; WINS Support - Tells the NMBD component of Samba to enable its WINS Server
;        the default is NO.
;   wins support = yes
; WINS Server - Tells the NMBD components of Samba to be a WINS Client
;        Note: Samba can be either a WINS Server, or a WINS Client, but
; NOT both
```

```
;   wins server = w.x.y.z
; WINS Proxy - Tells Samba to answer name resolution queries on behalf of
; a non WINS Client capable client, for this to work there must be at
; least one WINS Server on the network. The default is NO.
;   wins proxy = yes

;========================== Share Declarations ============================
[homes]
   comment = Home Directories
   browseable = no
   read only = no
   preserve case = yes
   short preserve case = yes
   create mode = 0750

; Un-comment the following and create the netlogon directory for Domain
; Logons
; [netlogon]
;   comment = Samba Network Logon Service
;   path = /home/netlogon
; Case sensitivity breaks logon script processing!!!
;   case censitive = no
;   guest ok = yes
;   locking = no
;   read only = yes
;   browseable = yes ; say No if you want to hide the NETLOGON share
;   admin users = @wheel

; NOTE: There is NO need to specifically define each individual printer
[printers]
   comment = All Printers
   path = /var/spool/samba
   browseable = no
   printable = yes
; Set public = yes to allow user >guest account= to print
   public = no
   writable = no
   create mode = 0700

;[tmp]
;   comment = Temporary file space
;   path = /tmp
;   read only = no
;   public = yes

; A public accessible directory, but read only, except for people in
; the staff group
;[public]
;   comment = Public Stuff
;   path = /home/samba
;   public = yes
;   writable = yes
;   printable = no
;   write list = @users

; Other examples.
;
; A private printer, usable only by fred. Spool data will be placed in
; fred=s home directory. Note that fred must have write access to the
; spool directory, wherever it is.
;[fredsprn]
;   comment = Fred=s Printer
;   valid users = fred
;   path = /homes/fred
;   printer = freds_printer
;   public = no
;   writable = no
;   printable = yes
;
; A private directory, usable only by fred. Note that fred requires write
; access to the directory.
;[fredsdir]
;   comment = Fred=s Service
;   path = /usr/somewhere/private
;   valid users = fred
```

```
;   public = no
;   writable = yes
;   printable = no
;
; a service which has a different directory for each machine that connects
; this allows you to tailor configurations to incoming machines. You could
; also use the %u option to tailor it by user name.
; The %m gets replaced with the machine name that is connecting.
;[pchome]
;   comment = PC Directories
;   path = /usr/pc/%m
;   public = no
;   writable = yes
;
;
; A publicly accessible directory, read/write to all users. Note that all
; files created in the directory by users will be owned by the default
; user, so any user with access can delete any other user=s files.
; Obviously this directory must be writable by the default user. Another
; user could of course be specified, in which case all files would be
; owned by that user instead.
;[public]
;   path = /usr/somewhere/else/public
;   public = yes
;   only guest = yes
;   writable = yes
;   printable = no
;
;
; The following two entries demonstrate how to share a directory so
; that two users can place files there that will be owned by the specific
; users. In this setup, the directory should be writable by both users
; and should have the sticky bit set on it to prevent abuse. Obviously
; this could be extended to as many users as required.
;[myshare]
;   comment = Mary=s and Fred=s stuff
;   path = /usr/somewhere/shared
;   valid users = mary fred
;   public = no
;   writable = yes
;   printable = no
;   create mask = 0765
```

Como ya hemos dicho smb.conf contiene tres secciones especiales y una o más secciones personalizadas. Las secciones especiales son [global], [inicios], e [impresoras].

4.1.2.1. La sección [global]

La sección [global] controla los parámetros para todo el servidor smb. Esta sección también proporciona los valores por defecto para el resto de las secciones.

```
[global]

; workgroup = NT-Domain-Name or Workgroup-Name, eg: REDHAT4
   workgroup = WORKGROUP

; comment is the equivalent of the NT Description field
   comment = RedHat Samba Server

; volume = used to emulate a CDRom label (can be set on a per share basis)
   volume = RedHat4
```

La primera línea de la sección global que se presentaba en el listado define el grupo de trabajo al que pertenecerá la computadora en la red. Después, el archivo incluye un comentario para el sistema e identifica una etiqueta de volumen.

```
; printing = BSD or SYSV or AIX, etc.
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes
```

La entrada siguiente indica al servidor Samba el tipo de sistema de impresión que se encuentra disponible en su servidor, y la línea siguiente informa de la ubicación del archivo de configuración de la impresora.

La línea que sigue a continuación ordena a Samba que ponga a disposición de la red todas las impresoras definidas en el archivo printcap.

```
; Uncomment this if you want a guest account
; guest account = pcguest
   log file = /var/log/samba-log.%m
; Put a capping on the size of the log files (in Kb)
   max log size = 50
```

La entrada siguiente proporciona el nombre de usuario para una cuenta de invitado o *guest* en el servidor. Esta cuenta se utiliza para autentificar a los usuarios que deseen utilizar los servicios Samba disponibles para las conexiones *guest*.

La entrada de archivo de registro especifica la ubicación del archivo de registro para cada cliente que acceda a los servicios Samba. El parámetro %m indica al servidor Samba que cree un archivo de registro aparte para cada cliente. La entrada max log size establece un tamaño máximo de archivo para todos los registros creados.

4.1.2.2. La sección [inicios]

La sección [inicios] permite a los clientes de red conectarse a un directorio inicial de usuario en su servidor sin tener una entrada específica en el archivo smb.conf. Cuando se solicita el servicio, el servidor Samba busca el archivo smb.conf para la sección específica correspondiente al servicio solicitado. Si no encuentra el servicio en cuestión, Samba comprueba la existencia de una sección [inicios]. Si dicha sección existe, Samba busca el archivo de contraseña para localizar el directorio inicial del usuario que está realizando la consulta. Si no lo encuentra, este directorio para a compartirse con la red.

```
[homes]
   comment = Home Directories
   browseable = no
   read only = no
   preserve case = yes
   short preserve case = yes
   create mode = 0750
```

La entrada comment se muestra a los clientes para que sepan qué comparticiones están disponibles. La entrada browseable indica a Samba cómo debe mostrar dicha compartición en una lista de navegación por red. El parámetro read-only controla la creación y modificación de archivos por parte de los usuarios en el directorio inicial cuando éste está siendo compartido en la red. Los parámetros preserve case y short preserve case indican al servidor que respete el tipo de letra de cualquier información escrita referida al servidor. Esto es importante porque aunque los sistemas de archivo de Windows normalmente no son sensibles al uso de mayúsculas y minúsculas, los sistemas de archivo de Linux sí lo son. La última entrada determina los permisos de archivo para cualquier archivo creado en el directorio compartido.

4.1.2.3. La sección [impresoras]

La sección [impresoras] define la forma en que se controlan los servicios de impresión si no se encuentran las entradas específicas en el archivo smb.conf. Al igual que la sección [inicios], si no se localiza una entrada específica para un servicio de impresión, Samba utiliza la sección [impresoras] (si existe) para permitir que un usuario pueda conectarse a cualquier impresora definida en el archivo /etc/printcap.

```
[printers]
   comment = All Printers
   path = /var/spool/samba
   browseable = no
   printable = yes
; Set public = yes to allow user >guest account= to print
   public = no
   writable = no
   create mode = 0700
```

Las entradas comment, browseable y create mode tienen el mismo significado que el especificado anteriormente en la sección [inicios]. La entrada path indica la ubicación del archivo de espera que debe utilizarse al

ejecutar una petición de impresión vía SMB.

Si el valor printable está predeterminado como Así≅, significa que dicho recurso de impresión puede utilizarse para imprimir. La entrada public controla el acceso a las tareas de impresión para las cuentas de *guest*.

4.1.2.4. Compartición de directorios

Una vez configurados los valores por defecto para el servidor Samba, pueden crearse directorios compartidos específicos, restringiendo o no su acceso a otros de usuarios. Por ejemplo, supongamos que quisiera crear un directorio al que sólo pudiera acceder un usuario. Para ello, tendría que crear una nueva sección y suministrar la información pertinente. Por lo general, tendría que especificar al servidor SMB los datos del usuario, la ruta del directorio y la información de configuración, tal y como se muestra a continuación:

```
[jacksdir]
   comment = Jack=s remote source code directory
   path = /usr/local/src
   valid users = tackett
   browsable = yes
   public = no
   writable = yes
   create mode = 0700
```

Con el ejemplo anterior, se crearía el directorio compartido denominado jacksdir. La ruta del directorio en el servidor local es /usr/local/src. Puesto que la entrada browsable está configurada de forma afirmativa (Así≅), jacksdir aparecerá en la lista de navegación de la red. Sin embargo, como la entrada public tiene asignada el valor Ano≅, sólo el usuario llamado tackett puede acceder a este directorio utilizando Samba. Podría permitir que otros usuarios accedieran a este directorio con sólo incluirlos en la entrada valid users.

El paquete SAMBA incluye entre sus aplicaciones una utilidad que facilita la creación del fichero de configuración, denominada *swat*. Es una herramienta para administración mediante una página *www* del fichero *smb.conf*.

Para disponer de ella bastará con modificar los ficheros de configuración del super servidor *inetd*. Así, en el fichero */etc/services* será necesario añadir la línea:

```
swat 901/tcp
```

Y al fichero */etc/inetd.conf* la línea:

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

Realizados estos cambios será necesario volver a lanzar a inetd. Para ello ejecutamos la orden:

```
kill -HUP <pid.inetd>
```

La herramienta necesita de un navegador que nos permita utilizar este servicio. Para ello sólo debemos indicarle en el navegador la máquina y puerto (901 en nuestro ejemplo) que contiene la herramienta. Para utilizarlo de forma local bastará especificar: *http://localhost:901/*. Una vez generado el fichero de configuración, por cualquier camino que hayamos seguido, y habiéndolo situado en el lugar adecuado (*/usr/local/samba/lib/smb.conf*) es conveniente, sobre todo antes de lanzar los demonios *sbmd* y *nmbd*, chequear que la configuración establecida no presenta errores. Para ello podemos utilizar la herramienta *testparm*. Si después de ejecutar la orden el resultado es OK se puede pasar al siguiente apartado, si no tendrás que volver a editar el fichero de configuración y buscar el error.

4.1.3. Iniciar los demonios ASMBD≅ y ANMBD≅

Los programas *smbd* y *nmbd* son los encargados de controlar la posibilidad de compartir recursos. Para ello utilizan el fichero de configuración *smb.conf*. *smbd* se encarga de compartir recursos y autorizar a usuarios a su uso, mientras que *nmbd* se encarga del servicio de nombres y mostrar recursos compartidos. Para que ellos puedan atender estas tareas necesitarán estar lanzados como demonio en el sistema. Existen dos formas de lanzar estos programas: como demonios independientes o mediante inetd.

Iniciar como demonios independientes

Este primer método va a consistir en crear un *script*, que se podría llamar *iniciarsmb*, donde se incluyan las órdenes para lanzar los demonios de forma independiente. Para crear el *script* se puede utilizar cualquier editor de texto plano (ASCII) como vi, joe, etc. Por ejemplo:

```
joe iniciarsmb
```

El contenido del script debe ser el siguiente:

```
#!/bin/sh
```

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

Una vez que se guarde el fichero necesitará asignarle permiso de ejecución mediante la orden:

```
chmod +x iniciarsmb
```

Para utilizar los servicios SAMBA bastará ejecutar el *script*. Pero si deseamos disponer de los servicios de SAMBA desde el inicio del sistema será necesario hacer una llamada a este fichero desde algún fichero de inicio del sistema, por ejemplo */etc/rc.d/rc.local* o */etc/rc.local*.

Por petición en inetd

Una segunda forma de utilizar servicios en un sistema Linux es mediante *inetd*, el superservidor. *Inetd* se conoce con el nombre de superservidor (super server) por ser el demonio encargado de atender los servicios solicitados.

Para ello *inted* utiliza ficheros de configuración */etc/services* y */etc/inetd.conf*. El primero permite definir los servicios que ofrece el sistema y que va a atender *inetd*. El segundo es la configuración para *inetd* y, básicamente, su función es especificar como *inetd* debe tratar un servicio concreto.

Para poder utilizar este segundo método y que los demonios de SAMBA sean lanzados por peticiones a *inetd* será necesario modificar estos ficheros. En concreto, en el fichero */etc/services* será necesario añadir las líneas:

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

En el fichero */etc/inetd.conf* será necesario incluir las líneas:

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

Una vez hechas las modificaciones habrá que reiniciar el demonio *inetd* para disponer de los servicios añadidos. Para ello, utilizaremos la orden:

```
kill -HUP <pid_inetd>
```

4.1.4. Utilidad SMBCLIENT

La utilidad *smbclient* es la otra parte principal de SAMBA, ésta nos va a permitir, desde una máquina Linux, utilizar recursos compartidos por máquinas Windows. *smbclient* es un cliente Linux que permite utilizar el protocolo SMB o CIFS.

Su formato se podría resumir de la siguiente forma:

```
smbclient [-L] servicio [clave] [-N] [-P] [-I IP_destino] [-U usuario] [-W grupo] [-c
comando]
```

Donde *servicio* es el nombre del servidor y del recurso compartido, con el formato *servidor**recurso*, al cual pretendemos realizar la conexión. El resto de parámetros son opcionales:

-*L* *host*: permite obtener un listado de los recursos compartidos en un servidor. Si se introduce la orden

```
smbclient -L \\jlam
```

El resultado serían los recursos compartidos por la máquina con nombre JLAM. La salida de la orden debe ser parecida a la siguiente:

```
Added interface ip=192.168.100.100
bcast=192.168.100.255  nmask=255.255.255.0
Domain=[DIESIA] OS=[Unix]       Server=[Samba 2.0.5a]

Sharename       Type    Comment
homes           Disk    Home Directories
tmp             Disk    Temporary file space
IPC$            IPC     IPC Service (Samba Server)
epson           Printer epson ascii
epson-a4-raw    Printer epson auto raw

Server          Comment
JLAM            Servidor SAMBA jlam

Workgroup       Master
DIESIA  JLAM
```

-*clave*: es la clave de acceso al recurso. Si no se especifica clave, un *prompt* solicitará al usuario la introducción de la misma. Si el recurso no tiene clave se podría dejar en blanco.

-*N*: indicará que el recurso al que deseamos acceder no tiene clave. De esta forma no aparecerá el *prompt* solicitando la clave.

-*P*: mantenida por compatibilidad, indica que se trata de un recurso de impresión. En SAMBA 2.0 el servidor reconoce el tipo de dispositivo, por lo que no es necesario indicar que se trata de un servicio de impresión.

-*I IP_destino*: permite especificar la IP del servidor destino.
-*U usuario*: permite especificar el usuario con el que deseamos establecer la conexión.
-*W grupo*: permite establecer el grupo con el que pretendemos establecer la conexión.
-*c comando*: permite ejecutar un comando.

Una vez establecida la conexión, mediante *smbclient*, aparecerá un *prompt* (*smb/>*) desde el que podemos utilizar una serie de órdenes para obtener y enviar ficheros desde y hacia ese directorio compartido. Las órdenes más utilizadas son:

Comandos de smbclient

Comando	**Parámetros**	**Descripción**
? o help	[comando]	Genera un mensaje de ayuda cuando así se requiere, o de forma general si no se especifica ningún comando.
!	[comando de shell]	Ejecuta un comando de shell especificado o pone al usuario en un indicador de shell.
cd	[directorio]	Cambia al directorio especificado en la computadora del servidor (no en la computadora local). Si no se especifica ningún directorio, smbclient registrará el directorio actual de trabajo.
lcd	[directorio]	Cambia al directorio especificado en la computadora local. Si no se especifica ningún directorio, smbclient registrará el directorio actual de trabajo en la computadora local.
del	[archivos]	Los archivos especificados en el servidor se eliminarían si el usuario tiene permiso para hacerlo. Los archivos pueden contener caracteres comodín.
dir o ls	[archivos]	Relaciona en una lista los archivos especificados. También puede utilizarse el comando ls para obtener una lista de archivos.
exit o quit	none	Sale del programa smbclient.
get	[archivo remoto] [nombre local]	Recupera el archivo remoto especificado y lo almacena en el servidor local. Si se especifica un nombre local, el archivo copiado se guardará con este nombre de archivo en vez de con el nombre de archivo que tenía en el servidor remoto.
mget	[archivos]	Copia todos los archivos especificados -incluidos los que coincidan con algún carácter comodín- en la computadora local.
md o mkdir	[directorio]	Crea el directorio especificado en la computadora remota.
rd o rmdir	[directorio]	Elimina el directorio especificado de la computadora remota.
put	[archivo]	Copia el archivo especificado de la computadora local al servidor.
mput	[archivo]	Copia todos los archivos especificados de la computadora local al servidor.

print	[archivo]	Imprime el archivo en cuestión en la computadora remota.
queue	none	Muestra todos los trabajos en espera de ser imprimidos en el servidor remoto.

4.2. INSTALACIÓN DE LOS CLIENTES.

Consideramos DOS, Windows para Trabajo en grupo y Windows 95 como clientes importantes.

En primer lugar se debe mencionar que los clientes solo pueden ser accesibles vía TCP/IP. NetBEUI y NetBIOS vía IPX no están disponibles por el momento. Puesto que TCP/IP se está haciendo más y más popular incluso bajo Novell y Microsoft es posible que no vaya a cambiar en un futuro próximo.

No vamos a tratar MS-DOS ni Windows 3.1. Nos centraremos en Windows NT y Windows 95.

4.2.1. Windows para trabajo en grupo (NT)

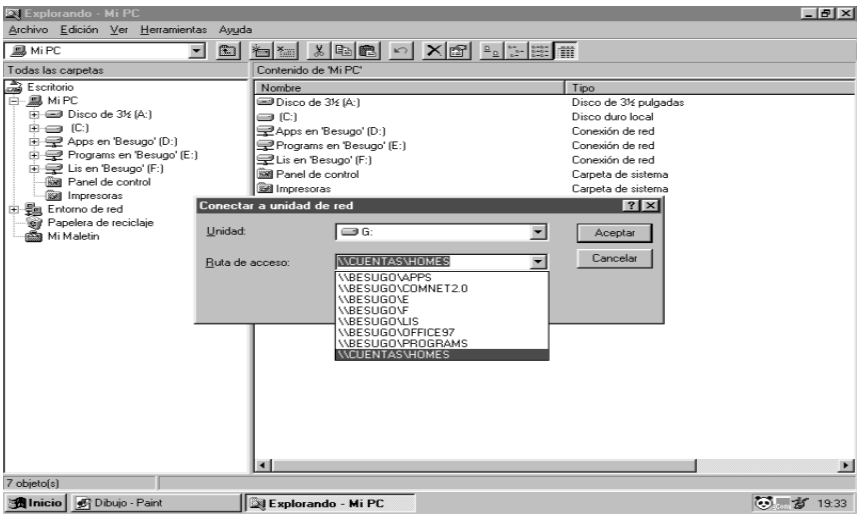
Windows para trabajo en grupo soporta SMP directamente (built-in). Este se llama ACliente para redes Microsoft≅ y normalmente corre sobre NetBEUI o IPX. La conexión TCP/IP tiene que ser instalada adicionalmente. La pila de protocolos de TCP/IP está disponible libremente en ftp://ftp.microsoft.com/bussys/clients/WFW. TCP32B.EXE es un fichero auto-extraíble (self-extractinf archive) que ha de ser descomprimido a un disco.

Ahora para su instalación. Primero se ha de entrar >not listed or actualised protocol= en >drivers= (network setup) el cual está en el disco.

Después los contenidos del disco se copian los parámetros TCP/IP que han de ser fijados con >Settings=. Aquí se necesita (al igual que con Linux) la dirección IP de la máquina, la máscara de red y tal vez un gateway existente. Más opciones se pueden seleccionar bajo >Extended=. Por lo tanto se debe seleccionar TCP/IP como protocolo estándar o quitar IPX/SPX y NetBEUI. Esto reduce la cantidad de memoria que se usa y acelera el arranque de Windows. Si hay otras máquinas que sigan este protocolo no se debe de quitar, por supuesto!.

Después de la instalación es obligatorio reiniciar. Por lo tanto se puede acceder a la máquina Linux vía Samba al igual que a otras máquinas.

Para usar una impresora sobre el servidor Samba se debería instalar el driver general postscript de impresión para Windows para trabajo en grupo y conectarla a la cola de Linux, la cual incluye posibilidades de reconocimiento apsfilter.



4.2.2. Windows 95.

Windows 95 tiene el protocolo TCP/IP soportado (built-in). Al igual que para Windows para trabajo en grupo no lo tiene instalado por defecto. Para añadirlo, seleccionamos >Add= (en >System=), >Protocols= TCP/IP de

Microsoft. El resto es como para Windows para trabajo en grupo.

4.3. EJEMPLO.

Veamos un ejemplo. Si disponemos de una máquina Windows, llamada JLAM, que tiene un directorio como recurso compartido, llamado TMP, el cuál no tiene clave de acceso, podríamos acceder a él con:

```
smbclient \\\\jlam\\tmp -N
```

4.3.1. Configuración SMB.CONF

Los recursos compartidos por una máquina Linux deben indicarse en el fichero *smb.conf*. Para ello bastará incluir una nueva sección para el recurso que deseamos compartir y asignarle los valores de los parámetros según nuestras necesidades. Por ejemplo, para compartir el directorio */tmp*, como TMP, con cualquier usuario, permitiendo incluso escribir en él, será necesario añadir al fichero *smb.conf* las siguientes líneas:

```
[TMP]
    comment = Directorio Temporal
    path = /tmp
    guest ok = yes
    writeable = yes
```

Si, además, deseamos compartir el directorio */home/grupo*, con el nombre de COMP, con todos pero que sólo puedan escribir los usuarios del grupo *diesia* y el usuario *jpajon*, es necesario incluir las líneas:

```
[COMP]
    comment = Directorio del Grupo
    path = /home/grupo
    guest ok = yes
    writeable = yes
    writeable list = @diesia, jpajon
```

Una vez el recurso ha sido compartido en Linux, para poder utilizarlo desde Windows, bastará conectar como si de otro recurso Windows se tratase. La figura muestra como conectar a una unidad de red desde Windows 95.

4.3.2. Uso de impresoras en Windows desde Linux

Si deseamos utilizar una impresora instalada en una máquina Windows desde Linux necesitaremos añadir una impresora en Linux que utilice *smbclient* para establecer la conexión y enviar el fichero. Esta breve descripción del proceso, quizás nos ha dejado con las mismas dudas. Profundicemos en ello. Linux utiliza el fichero */etc/printcap* para configuración de las impresoras. Este fichero cuenta con una línea por cada impresora, donde entre otras cosas se especifican el nombre de la impresora, dispositivo físico (*/dev/lpt1*), directorio de *spool*, tamaño máximo del fichero, fichero de cuentas y filtro de impresión. Un ejemplo podría ser:

```
#/etc/printcap
lp      :cm=HPDeskjet 600:\            # Nombre
        :lp=/dev/lp1:\                 # Dispositivo
        :sd=/var/spool/lpd/lp:\# Directorio de spool
        :mx#0:\                # Tamaño máximo del fichero
        :if:=/usr/bin/filtro:          # Filtro de impresión
```

Para utilizar una impresora remota bastará incluir una entrada que especifique las características de la impresora, añadir una entrada para el fichero de cuentas y utilizar como filtro un *script* que mediante la utilidad *smbclient* realice la conexión y el envío del fichero. Podría ser algo parecido a:

```
#/etc/printcap
# Configuración para impresora remota con SAMBA
smb     :cm=HPDeskjet 820:\            # Nombre
        :lp=/dev/null:\                # Dispositivo
        :sd=/var/spool/lpd/smb:\               # Directorio de spool
        :af=/var/spool/lpd/smb/acct:\  # Fichero de cuenta
        :mx#0:\                        # Tamaño máximo del fichero
        :if:=/usr/bin/smbprint         # Filtro de impresión
```

Como puede apreciarse el script que utiliza *smbclient* se ha denominado *smbprint*. Esto no ha sido tomado al azar, sino que éste es un *script* que generalmente se usa para este fin y puedes encontrar una copia del mismo en el directorio de los fuentes de SAMBA: */usr/local/samba-2.0.5/docs/examples/printing*.

4.3.3 Uso de impresoras en Linux desde Windows

Para compartir una impresora instalada en Linux desde una máquina Windows, al igual que al compartir directorios, será necesario especificarlo en el fichero de configuración *smb.conf*. Este fichero tiene una sección especial para impresoras [*printers*]. Antes de entrar en esa sección será necesario especificar en la sección [*global*] los valores para los parámetros *printing* (sistema de impresión), *printcap name* (ubicación del fichero *printcap*), *load printers* (posibilidad de cargar impresoras). Los valores de estos parámetros podrían ser:

```
[global]
```

```
...
    printing = bsd
    printcap name = /etc/printcap
    load printers = yes
```

Ahora será necesario configurar la sección [*printers*]. En ella se tendrán que especificar los valores de los parámetros para todas las impresoras. Para definir parámetros para impresoras específicas se utilizan secciones propias para cada impresora. Un ejemplo de cómo podría quedar la sección [*printers*] podría ser el siguiente:

```
[printers]
    comment = Todas las impresoras
    security = server
    path = /var/spool/lpd/lp
    browseable = no
    printable = yes
    guest ok = yes
    writeable = no
    create mode = 0700
```

Una posible sección para una impresora compartida podría ser como se muestra en la siguiente sección:

```
[dj600]
    security = server
    path = /var/spool/lpd/lp
    printer name = lp
    writeable = yes
    guest ok = yes
    printable = yes
    print command = lpr -r -h -P %p %s
```

Una vez que Linux está compartiendo una impresora para instalarla desde Windows se seguirán los mismo pasos que para cualquier otra impresora remota.

5. PRUEBAS

5.1. Prueba del archivo smb.conf

Una vez creado el archivo de configuración, conviene probarlo para asegurarse de que está correcto con el programa testparm. testparm es un programa sencillo de prueba que permite comprobar la corrección interna del archivo de configuración /etc/smb.conf. Si el programa no detecta ningún fallo, puede utilizar el archivo de configuración con la certeza de que smbd cargará sin problemas el archivo de configuración.

PRECAUCIÓN:

El uso de testparm NO garantiza que los servicios especificados en el archivo de configuración estarán disponibles o se ejecutarán satisfactoriamente.

testparm tiene asignada la siguiente línea de comandos: testparm [*configfile* [hostname *hostip*]]
donde *configfile* indica la ubicación del archivo smb.conf si éste no se encuentra en su ubicación predeterminada (/etc/smb.conf). El parámetro opcional hostname hostIP ordena a testparm que compruebe si el sistema central tiene acceso a los servicios prestados en el archivo smb.conf.

El ejemplo siguiente ofrece un resultado de muestra de la ejecución de testparm. Si el archivo smb.conf contiene algún error, el programa lo detectará generando el correspondiente mensaje de error.

```
# testparm
Load smb config files from /etc/smb.conf
Processing section A[homes]≅
Processing section A[printers]≅
Loaded services file OK.
Press enter to see a dump of your service definitions
```

Al pulsar la tecla <Intro>, testparm empezará a evaluar cada una de las secciones definidas en el archivo de configuración.

5.2. TESTEAR LOS DEMONIOS DE SAMBA.

Una manera de comprobar que todo funciona correctamente es utilizar el programa *smbclient* para examinar lo que el servidor ofrece a la red. Si todo está configurado correctamente se debería de poder ver algo similar a 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 lo mejor es ver el log: /usr/local/samba/var/log.smb

5.3. Diagnósticos del servidor samba.

Para un nivel más avanzado de pruebas, se recomienda echar un vistazo al archivo `diagnosis.txt` de la documentación de samba, en el que se plantean hasta 10 tests distintos para nuestro sistema. Para cada administrador, sería muy recomendable que analizase este documento y se programase un script que le ayude a realizar una prueba a su sistema de manera sencilla.

6. GESTIÓN DIARIA

Para una correcta administración del servicio la mejor manera es inspeccionar periódicamente los logs de samba. Se puede configurar la exhaustividad de estos ficheros así como su nivel de depuración con ayuda del archivo smb.conf: entradas: debug timestamp, debug level, log file, log level, syslog, syslog only, timestamp logs, ...

Las nuevas versiones de Samba tienen la tendencia de incluir mejoras para facilitar la administración del servicio. Así en la versión 2.0 es de destacar la incorporación del Samba Web Administration Tool (SWAT) (una utilidad basada en un navegador para configurar samba).

SWAT have sencillo establecer un servidor y cambiar su configuración, sin tener que dar una configuración basada en un fichero de texto. SWAT da una interfaz gráfica a los recursos que Samba comparte con sus clientes.

En tiempo de compilación GNU autoconf se usa para compilar inicialmente y preparar SWAT, para que pueda ser utilizado lo antes posible.

6.1. Los logs de Samba.

La primera línea de ataque a la hora de problemas o manteniendo, debería ser chequear los ficheros log. Los ficheros log de Samba pueden ayudar a diagnosticar la gran mayoría de los problemas a los que los administradores de Samba están habituados a encontrarse. Como ya comentábamos antes, se puede configurar al servidor para que nos muestre unos logs más o menos completos según queramos. También hay variables de sustitución que nos ayudan a separar los logs de una determinada máquina, share o combinación deseada.

Por defecto los logs se encuentran en samba_directory/var/smbd.log y samba_directory/var/nmbd.log, donde samba_directory es el directorio donde Samba está instalado (típicamente, /usr/local/samba). Se puede cambiar la localización y el nombre de los logs utilizando la opción de configuración *log file* en el *smb.conf*. Esta opción acepta todo tipo de variables de sustitución, así que fácilmente se puede conseguir que el servidor separa un log para cada cliente conectado, simplemente especificando lo siguiente en la sección [global] de smb.conf:

```
log file = %m.log
```

Alternativamente, se puede especificar un directorio para los logs con el flag -l en la línea de comandos, por ejemplo:

```
smbd -l /usr/local/var/samba
```

Otro truco interesante es que cada servidor tenga un log para cada servicio (share) que ofrece, especialmente si sospechamos que un servicio en particular es el causante del problema. Usamos la variable %s en la sección [global] de smb.conf:

```
log file = %s.log
```

6.2. TEST UTILITIES.

Como ya comentamos en el apartado de las pruebas, el fichero DIAGNOSIS.TXT del directorio /docs/textdocs del kit de distribución Samba.

6.3. UTILIDADES UNIX.

Algunas veces es útil usar una herramienta exterior a Samba para examinar que es lo que está ocurriendo en el servidor. Dos herramientas de diagnóstico pueden ser *trace* (strace en Linux) y *tcpdump*.

7. BIBLIOGRAFÍA.

Si alguien desea saber más sobre samba puede consultar las siguientes fuentes:

-Documentación de la aplicación (contenida en el directorio de ésta y accesible también a través de las páginas del manual).

-LINUX EDICIÓN ESPECIAL 40 Edición. Jack Tackett, Jr. & Steve Burnett. Prentice-Hall 1999. Capítulo 15. (Pag. 350). AUso de Samba. Por Jack Tackett≅

-S.u.S.E. Linux 5.2. Installation, Configuration & First Steps. Marzo 1998.

-RUNNING LINUX. 20 edición.

-≅Creación, configuración y manejo de máquinas virtuales≅, artículo de J. Manuel Canelada y A. José Novillo. (jmcanelada@retemail.es y anovillo@retemail.es) de la revista Linux Actual Año 2, número 12, pags. 17-22.

-Revista SOLO LINUX. Num 6. Pag. 41. José Luis Álvarez Macías (alvarez@uhu.es): SAMBA permite compartir recursos entre Linux y Windows.

-Revista SOLO PROGRAMADORES LINUX. Año III. Num. 14. Samuel Lobo Poyo, profesor del área de Ingeniería de sistemas y automática de la universidad Calos III de Madrid.

-El web site oficial: http://samba.org.