

Compartición de datos con SAMBA

Samba es una solución software de interoperabilidad con Windows disponible en los sistemas Linux y Unix. El nombre de Samba viene del protocolo SMB: Server Message Block utilizado para la compartición de recursos en las redes Microsoft. Permite en particular compartir archivos e impresoras en los servidores Linux para clientes Windows. La suite software Samba también tiene un cliente que permite a las máquinas Linux conectarse a los recursos compartidos de un servidor Windows.

1. Configuración general

a. Los daemons samba

Samba se basa en dos daemons llamados **nmbd** y **smbd**. El daemon **nmbd** se encarga de anunciar los servicios y en general de todo el funcionamiento NetBIOS over IP. El daemon **smbd** se encarga de las comparticiones de archivos y de impresoras.

El script de gestión del servicio que generalmente está presente en las distribuciones inicia estos dos daemons en cada arranque.

b. Los archivos de configuración

Los daemons samba tienen su configuración en el archivo de configuración **smb.conf**, generalmente ubicado en el directorio **/etc/samba**.

El archivo de configuración está dividido en secciones estandarizadas, cada una precedida por un título entre corchetes. Los parámetros de funcionamiento se ubican en cada una de estas secciones escritos siguiendo la sintaxis **parámetro = valor**.

Formato resumido de smb.conf

```
[sección1]
parámetro1 = valor1
parámetro2 = valor2
[sección2]
parámetro3 = valor3
parámetro4 = valor4
```

Existe una herramienta muy útil llamada **testparm** que valida el formato de un archivo de configuración samba. También devuelve un informe puro (sin líneas de comentarios) de la configuración por la salida estándar. Naturalmente, esta salida se puede redirigir a un archivo y generar un **smb.conf** legible y de tamaño razonable. Cabe destacar que el comando **testparm** ignora todo parámetro del archivo de configuración si se ha configurado con su valor por defecto. Este comportamiento se puede modificar con la opción **-v**. Entonces todas las opciones aplicables se mostrarán.

Ejemplo de uso de testparm para generar un archivo smb.conf sencillo

Este método se utiliza a menudo para usar un archivo de configuración con muchos comentarios obteniendo un archivo real de dimensiones razonables.

```
alfa:/etc/samba# mv smb.conf big.smb.conf
alfa:/etc/samba# wc -l big.smb.conf
326 big.smb.conf
alfa:/etc/samba# testparm big.smb.conf > smb.conf
Load smb config files from big.smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

alfa:/etc/samba# wc -l smb.conf
31 smb.conf
```

```
alfa:/etc/samba# testparm -v big.smb.conf > todas-las-opciones.info.smb.conf
alfa:/etc/samba#
```

Las versiones preinstaladas de samba ofrecen siempre un archivo **smb.conf** preconfigurado. Aunque este archivo puede considerarse una buena base de partida, su tamaño (326 líneas para Debian) presenta el riesgo de impresionar a los iniciados. Seguramente sería mejor generar un archivo sólo con los elementos explícitamente necesarios.

c. Configuración global

En su configuración más simple, una implementación samba incluye un servidor que alberga uno o más recursos. Algunos parámetros relacionados con el funcionamiento global y la identidad de este servidor se encuentran en una sección llamada **global** del archivo **smb.conf**.

En los ejemplos siguientes, nos pondremos en la situación de un servidor simple, fuera de un dominio Windows, que tiene comparticiones para clientes Windows.

Elementos comunes de la sección [global] en smb.conf

```
workgroup = grupo_de_trabajo
server string = comentario
log file = /ruta/log.%m
max log size = log_maxi
security = user (por defecto)
encrypt passwords = true (por defecto)
```

Sección [global] del archivo smb.conf	
<i>grupo_de_trabajo</i>	El nombre del grupo de trabajo del servidor. Hay que tener en cuenta que este parámetro también proporciona el nombre del dominio cuando está trabajando en un dominio.
<i>comentario</i>	Comentario asociado al servidor. Visible por ejemplo en el Entorno de Red de las máquinas Windows.
<i>log.%m</i>	Definición del formato estándar de los archivos de registro.
<i>log_maxi</i>	Definición del tamaño máximo de los archivos de registro.
<i>user</i>	Opcional y que es un parámetro por defecto. Parámetro de seguridad que obliga a autenticarse con una cuenta de usuario.
<i>encrypt passwords</i>	Opcional y que es un parámetro por defecto. Necesario para todos los clientes modernos que presentan de forma nativa contraseñas encriptadas (desde NT4SP3).

2. Compartición de directorios

Para compartir un directorio hay que añadir una sección en el archivo **smb.conf**.

Formato típico de una sección de compartición en smb.conf

```
[nombre_compartición]
comment = comentario
path = ruta
readonly = sólo_lectura
browseable = yes
```

Declaración de comparticiones en smb.conf.	
<i>nombre_compartición</i>	El nombre con el que se verá la compartición en las máquinas Windows.

<i>comentario</i>	Opcional. Definición del comentario asociado a la compartición.
<i>ruta</i>	Definición de la ruta del directorio que se desea compartir. yEl directorio debe existir en el sistema de archivos Linux.
<i>sólo_lectura</i>	Definición del acceso a la compartición en modo de sólo lectura o lectura/escritura. <i>sólo_lectura</i> tendrá el valor <i>yes</i> o <i>no</i> según se elija la configuración. Hay que tener en cuenta que este parámetro se aplica a la compartición y que el acceso queda sometido a los permisos del sistema de archivos Linux.
<i>browseable</i>	Gestión de la visibilidad de la compartición para los clientes.

Si consulta el conjunto de parámetros disponibles para el archivo **smb.conf**, puede quedar comprensiblemente impresionado por su gran cantidad. Hay que tener en cuenta que muchos parámetros funcionales pueden expresarse de varias formas. Tomemos por ejemplo el parámetro de acceso a una compartición en sólo lectura que hemos visto anteriormente. Todas las expresiones siguientes son equivalentes:

readonly = *yes*

readonly = *true*

writable = *no*

writable = *false*

writeable = *no*

writeable = *false*

3. Administración de credenciales

a. Algoritmos de hash y de almacenamiento de contraseñas

En la gran mayoría de sistemas operativos y aplicaciones las contraseñas no se almacenan sin encriptar. Las contraseñas de las cuentas están encriptadas y solamente se almacena la versión encriptada. La contraseña sin encriptar se olvida tan pronto como se encripta.

Cuando un usuario se conecta y teclea sus credenciales para identificarse, la contraseña se codifica de inmediato y esta versión recién encriptada de la contraseña se compara con la versión almacenada en la base de datos de cuentas de usuario del sistema. De este modo, la contraseña no se transmite nunca sin encriptar por la red.

Los algoritmos utilizados para encriptar la contraseña pertenecen a la familia de los algoritmos de hash. y Funcionan de un modo un poco particular, en el sentido de que permiten encriptar pero nunca desencriptar datos: tienen un único sentido y por este hecho se los considera de una tipología diferenciada dentro del mundo de la criptografía. y Este modo de funcionamiento justifica por qué, cuando un usuario pierde su contraseña, se le puede reasignar una nueva, pero no se le puede decir cuál era la que ha olvidado. La única información que se guarda es la versión encriptada de la contraseña y es hipotéticamente indescifrable.

Los algoritmos de hash más comunes se llaman MD4, MD5 y SHA1. Se utilizan para almacenar contraseñas, las operaciones de firma digital y los controles de integridad.

b. Autenticación con servidores Samba

Un servidor Linux con la suite software Samba instalada utiliza nativamente las cuentas del sistema para las autenticaciones Samba. De este modo, toda conexión por parte de un cliente se realiza con una cuenta de usuario albergada en el sistema Linux. Sin embargo, esta situación podría ser un problema. y El cliente de Windows presentará una contraseña encriptada por el algoritmo de hash nativo de los sistemas Windows MD4: Message Digest 4, mientras que las contraseñas de los sistemas Linux usan el algoritmo MD5: Message Digest 5. La contraseña encriptada proporcionada por el cliente Windows no será, por tanto, la misma que la que está almacenada en el archivo **/etc/shadow** del sistema Linux. Por consiguiente, la autenticación será imposible, aunque la contraseña sin cifrar sea la misma.

Para que los clientes Windows se puedan autenticar en los sistemas Linux, hay que hacer que estos sistemas alberguen una versión de la contraseña encriptada en MD4 además de la contraseña nativa encriptada en MD5. y Estas dos contraseñas se administrarán de forma independiente y podrán ser incluso distintas.

c. Generación de contraseñas MD4

El comando específico **smbpasswd** permite crear una contraseña MD4 para una cuenta Linux existente. Esta contraseña se almacenará aparte, generalmente en el archivo **/etc/samba/smbpasswd**.

Sintaxis del comando smbpasswd para asignar una contraseña

```
smbpasswd -a nombre_cuenta
```

Comando smbpasswd: opciones y parámetros	
-a	Opcional. Necesario si la cuenta no dispone todavía de una contraseña samba.
nombre_cuenta	La cuenta Linux a la que hay que asignar la contraseña samba.

d. Sincronización con contraseñas Linux

Se puede solicitar que se sincronicen las contraseñas samba con las contraseñas del sistema Linux. Atención, tal y como se ha explicado anteriormente, las contraseñas se encriptan en ambos sistemas con algoritmos de hash distintos, que son por definición irreversibles. La sincronización sólo se puede realizar en el momento en que la contraseña se introduce sin encriptar cuando se utiliza el comando **smbpasswd**. Entonces cuando la contraseña se encripta dos veces con los dos algoritmos diferentes y se modifican las dos bases de datos de cuentas de usuario. Esta sincronización se activa mediante una directiva en el archivo **smb.conf**.

Activación de la sincronización de contraseñas en smb.conf

```
unix password sync = yes
```

e. Borrado o desactivación de una cuenta samba

Se puede necesitar interrumpir el acceso de una cuenta de un usuario a los recursos compartidos del servidor samba. El comando **smbpasswd** puede eliminar, desactivar o reactivar la cuenta samba, independientemente de la cuenta Linux asociada.

Comando smbpasswd para desactivar una cuenta samba

```
smbpasswd -d nombre_cuenta
```

Comando smbpasswd para reactivar una cuenta samba

```
smbpasswd -e nombre_cuenta
```

Comando smbpasswd para eliminar una cuenta samba

```
smbpasswd -x nombre_cuenta
```

Donde *nombre_cuenta* representa la cuenta de usuario samba que se desea modificar. Cabe decir que las operaciones en las cuentas samba no tienen efecto alguno en la cuenta Linux correspondiente.

4. El cliente Samba

El cliente samba permite acceder a una compartición de una máquina Windows o Samba desde un cliente Linux. Permite incluso a un cliente Linux conectarse a un servidor Samba Linux, pero el objetivo es más bien acceder a datos de una compartición Windows desde una máquina Linux. Los dos comandos principales del cliente samba son **smbclient** y **smbmount**.

a. Uso puntual de recursos compartidos con smbclient

Básicamente, se utiliza **smbclient** para obtener información de los recursos compartidos albergados por un servidor SMB.

Utilización de smbclient para obtener información acerca de un servidor smb

```
smbclient -L dirección_servidor -U nombre_usuario
```

smbclient para mostrar comparticiones: parámetros	
<i>dirección_servidor</i>	La dirección IP del servidor del que se quiere mostrar los recursos.
<i>nombre_usuario</i>	Indica el nombre del usuario que realiza la consulta al servidor. Tiene que ser una cuenta existente y válida en el servidor.

También se puede utilizar el comando **smbclient** de forma interactiva conectándose a un recurso compartido y accediendo a un shell que permita realizar operaciones con los archivos.

Utilizaciones de smbclient en modo interactivo

```
smbclient \\\dirección_servidor\\comparticiónU nombre_usuario  
smbclient //dirección_servidor/compartición -U nombre_usuario
```

Donde *compartición* representa el nombre de la compartición albergada por el servidor. Las múltiples contrabarras son necesarias aunque generen una sintaxis un tanto curiosa. De hecho, se trata de una ruta UNC: *Uniform Naming Convention*, utilizada para designar un recurso en los entornos Windows. Una ruta UNC se compone del nombre del servidor, precedido de dos contrabarras, seguido de la ruta al recurso, separando con una contrabarra cada nivel. Sin embargo, se da el caso que en los sistemas Linux la contrabarra es un carácter reservado que indica que el shell no debe interpretar el carácter siguiente. Para escribir una contrabarra de verdad, hay que anteponerle otra para indicarle al sistema que la segunda debe considerarse como una contrabarra normal. Una alternativa más ligera consiste en cambiar las contrabarras por barras normales. Ambas sintaxis están permitidas.

Una vez que este comando se ejecuta y después de haber introducido la contraseña del usuario, se entra en el shell específico **smbclient** que permite realizar operaciones con los archivos. Los principales usos serán por supuesto obtener o enviar archivos a la compartición. Se puede desplazarse por la estructura de directorios con el comando **cd**. Además, los dos comandos básicos son **get** para obtener archivos y **put** para enviar archivos a la compartición.

Ejemplo de uso de smbclient en modo interactivo

La utilidad **smbclient** presenta un conjunto de comandos parecido al de los clientes **yFTP**.

```
alfa:~# smbclient \\\192.168.0.1\data -U toto  
Enter toto's password:  
Domain=[WSEVRIDOR] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]  
smb: \> ls  
  
.                D            0  Wed Feb  3 19:28:33 2010  
..               D            0  Wed Feb  3 19:28:33 2010  
dos              D            0  Wed Feb  3 18:50:05 2010  
uno              D            0  Wed Feb  3 19:28:38 2010  
  
40915 blocks of size 262144. 34718 blocks available  
smb: \> cd uno  
smb: \uno\> ls  
  
.                D            0  Wed Feb  3 19:28:38 2010  
..               D            0  Wed Feb  3 19:28:38 2010  
archivo.txt      A           27  Wed Feb  3 19:15:49 2010  
truco.bmp        A            0  Wed Feb  3 18:46:44 2010  
  
40915 blocks of size 262144. 34718 blocks available  
smb: \uno\> get archivo.txt  
getting file \uno\archivo.txt of size 27 as fichier.txt (2,0 kb/s) (average 2,0 kb/s)  
smb: \uno\> exit  
alfa:~# ls  
archivo.txt  
alfa:~#
```

b. Montaje de una compartición smb con smbmount

Aunque **smbclient** permite realizar un acceso puntual a las comparticiones, existe un método más cómodo para usar directorios compartidos desde un cliente Linux: el montaje de una compartición en la estación de trabajo Linux.

El comando **smbmount** permite realizar el montaje de una compartición SMB en un directorio local tal y como se puede hacer con un sistema de archivos local o una compartición NFS.

Sintaxis del comando smbmount

```
smbmount \\\dirección_servidor\compartición punto_de_montaje -o user=nombre_usuario
```

```
smbmount //dirección_servidor/compartición punto_de_montaje -o user=nombre_usuario
```

smbmount: opciones y parámetros	
<i>dirección_servidor</i>	Dirección IP del servidor que tiene la compartición a la que se quiere acceder.
<i>compartición</i>	Nombre de la compartición albergada en el servidor.
<i>punto_de_montaje</i>	Directorio existente en el que se montará la compartición.
<i>nombre_usuario</i>	Nombre del usuario que realizará la petición al servidor. Tiene que ser una cuenta existente y válida en el servidor.

Existe una alternativa a esta sintaxis, que supone realizar el montaje mediante el comando **mount** llamando a **smbmount** como subprograma. Esta sintaxis presenta la ventaja de uniformar todas las operaciones de montaje y, por lo tanto, sólo tener que recordar una sintaxis genérica.

Sintaxis del comando mount para montar una compartición smb

```
mount -t smbfs -o username=nombre_usuario //dirección_servidor/compartición punto_de_montaje
```

La opción **-t smbfs** provoca la llamada al subprograma **smbmount** para realizar el montaje, pero a partir de una sintaxis casi estándar para realizar el montaje.

c. Montaje de una compartición CIFS

Para responder a las necesidades de apertura del protocolo, SMB se ha normalizado, ha evolucionado y se ahora se denomina CIFS: *Common Internet File System*. La suite software Samba ahora designa a su cliente y a los elementos software con este nombre. Como los hábitos resisten a morir, todavía persiste el uso del nombre de SMB.

Según las versiones de samba usadas, se puede utilizar smb, cifs o smb y cifs indiferentemente. La tendencia es la desaparición de smb en beneficio de cifs.

Sintaxis del comando mount para comparticiones cifs

```
mount -t cifs -o username=nombre_usuario //dirección_servidor/compartición punto_de_montaje
```



Se puede verificar desde el lado servidor cuáles son los clientes que están conectados. El comando **smbstatus** muestra las conexiones smb activas.