

U6 Recursos compartidos

Parte II. Linux. Samba

Implantación de Sistemas Operativos

Índice

6.1 Introducción.

6.2 Instalación de un servidor Samba.

6.3 Configuración de Samba.

6.4 Ejemplo práctico.

6.1 Introducción.

- La mayoría de empresas utilizan la topología de estrella en sus sistemas informáticos. Los ordenadores de los empleados se conectan a un servidor central.
- Vamos a ver el caso de un servidor con sistema operativo Linux, mientras que los ordenadores de los empleados tendrán el sistema operativo Windows o Linux.
- Para conectar ordenadores Windows con un servidor de dominio Linux y permitir que accedan a sus carpetas o recursos, se utiliza el protocolo Samba. También trataremos como acceder al servidor con un terminal remoto mediante los protocolos SSH.
- **SAMBA:** es una implementación para sistemas operativos Linux de una serie de protocolos utilizados por Windows para compartir archivos, servicio de nombres NETBIOS para identificar máquinas en una red local, autenticación en active directory, servicios de dominio, etc.

6.2 Instalación de un servidor Samba.

- La configuración de SAMBA se edita a mano desde el archivo **/etc/samba/smb.conf**. También desde otras herramientas gráficas como Webmin, SWAT, o las que incorporan escritorios como Gnome o KDE. Nosotros lo haremos a mano.
- La configuración más básica de SAMBA implica unirse a un grupo de trabajo (sin servidor de dominio) para compartir archivos e impresoras en red. Puede configurarse para entrar en un dominio, o para ser controlador de dominio. Todo esto interactuando sin problemas con máquinas Windows.
- Para usar SAMBA debemos instalar los paquetes **samba, samba-common, smbclient, cifs-utils**.
 - **sudo apt install samba samba-common smbclient cifs-utils**

6.2 Instalación de un servidor Samba.

- **samba:** servidor para compartir carpetas
- **samba common:** librería para samba
- **smbclient:** cliente para conectarnos a carpetas compartidas desde esta máquina virtual (es posible que no se use)
- **cifs-utils:** estándar de Windows para compartición de carpetas.
- Las carpetas que compartamos con samba serán accesibles desde máquinas Linux y máquinas Windows. Desde Linux, gracias a smbclient, podríamos acceder a carpetas compartidas en Windows.
- Las cifs-utils son necesarias. Este paquete tiene una serie de comandos que vamos a usar. Por ejemplo el “testparm” que nos permite comprobar si nuestro archivo de configuración de samba es correcto y además nos lo imprime por pantalla para que veamos la configuración que tenemos en ese momento.

6.2 Instalación de un servidor Samba.

- Una vez instalados los paquetes debemos modificar manualmente el fichero `/etc/samba/smb.conf`
- Antes de modificar el fichero es conveniente hacer una copia de seguridad de la configuración de SAMBA:
 - `sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.bak`
- Cada vez que cambiéis la configuración de SAMBA, debéis reiniciar el servicio (la d de demonio):
 - `sudo systemctl restart smbd`

6.3 Configuración de un servidor Samba.

- Editaremos el archivo de configuración con un editor como nano por ejemplo:
 - `sudo nano /etc/samba/smb.conf`
- Los bloques de configuración del archivo (1 por cada recurso compartido), están delimitados con el nombre entre corchetes en el archivo de configuración, siendo [global] (global settings) el que contiene la configuración general de SAMBA. Y share definitions la parte donde están los recursos compartidos.

CONFIGURACION GLOBAL

Dentro de este bloque, que afecta a todos los recursos compartidos, podemos encontrar los siguientes parámetros de configuración. Primero buscamos si están, si están comentados, los descomentamos y si no están los añadimos:

6.3 Configuración de un servidor Samba.

CONFIGURACION GLOBAL

[global]

- **workgroup = WORKGROUP** → Nombre del grupo de trabajo al que pertenece la máquina (siempre que no haya un dominio) Si todos los equipos están en el mismo grupo de trabajo es más fácil que se vean. En Windows es igual (propiedades del equipo, configuración de acceso remoto, nombre del equipo). En este caso dejamos el que está que es el mismo que la máquina anfitriona.
- **server string = Samba Server %v** → descripción de mi servidor samba. Lo que ven los usuarios que se conectan en red. %v es la versión de samba. %h nombre del host.
- **netbios name = SAMBA-SERVER** → Nombre que tendrá la máquina en una red Windows (coincidirá con el nombre que le hayamos dado al equipo en la instalación por defecto).
- **guest account = nobody** → usuario del sistema que se utilizará por defecto cuando se acceda de forma anónima a un recurso compartido. En muchos sistemas existe un usuario llamado nobody para estas funciones, pero se puede usar cualquiera (también existe un grupo llamado nogroup).

6.3 Configuración de un servidor Samba.

CONFIGURACION GLOBAL

[global]

- **security = user** → Tipo de seguridad a nivel de autenticación de usuarios. Los valores pueden ser:
 - **user** → Recomendada por defecto, los usuarios deberán usar un nombre de usuario y contraseña del equipo.
 - **share** → No requiere usuario y contraseña. Inseguro.
 - **domain** → Sólo cuando se forme parte de un dominio. El usuario debe ser válido en nuestra máquina y en el controlador de dominio.
 - **server** → Valida el usuario con otro servidor, si esto falla, lo valida en la propia máquina.
- **map to guest = bad user** → Indica cuando un usuario se conectará como anónimo (guest account) teniendo en cuenta que vamos a pedir usuario y contraseña. No tiene sentido utilizarlo con la opción security = share por ejemplo. Las opciones son:

6.3 Configuración de un servidor Samba.

CONFIGURACION GLOBAL

[global]

- **Never** → Por defecto. No se permiten usuarios anónimos.
- **Bad User** → Cuando el nombre de usuario introducido no exista.
- **Bad Password** → Cuando el password sea incorrecto. Este método no es muy recomendable ya que el sistema no avisará al usuario de que ha introducido mal la contraseña y entrará como invitado.
- **unix password sync = yes** → mantiene sincronizadas las contraseñas del usuario para loguearse en el equipo y las de acceso a las carpetas compartidas de samba (que no tienen por qué ser las mismas). Si lo tenemos en yes, cuando cambiemos la contraseña del usuario SAMBA se cambiará también su contraseña para loguearse en el equipo.

6.3 Configuración de un servidor Samba.

CONFIGURACION COMPARTIDA (share definitions)

[homes]

Si quitamos el “;” (misma función que la almohadilla) lo que haremos es poder compartir las carpetas home a cada usuario del sistema. Eso querría decir que cada usuario podría acceder a su carpeta home como carpeta compartida.

[netlogon]

Tendríamos que descomentar esta la parte si samba fuera un controlador de dominio. En este caso no lo vamos a utilizar.

[profiles]

Guarda los perfiles de usuario en caso de que samba fuese controlador de dominio.

6.3 Configuración de un servidor Samba.

CONFIGURACION COMPARTIDA (share definitions)

[printers]

Para compartir las impresoras. Está descomentado, lo que significa que por defecto se comparten las impresoras.

[print\$]

Lo mismo, pero para clientes de Windows.

Y a partir de ahí es donde podríamos escribir para añadir nuestras carpetas compartidas.

[compartida] ponemos entre corchetes el nombre con el que se verá el recurso en red, y debajo la configuración.

6.3 Configuración de un servidor Samba.

- Los parámetros que vienen a continuación son **opcionales**, si los incluimos en la sección **GLOBAL**, afectarán a todos los usuarios del sistema y todas las carpetas. Sin embargo si estos parámetros los ponemos en la sección **SHARE** dentro de las carpetas que yo cree como recursos compartidos, solo afectarán a esas carpetas de forma individual.
- **invalid users** → Lista de usuarios (separados por espacio). Impide el login de los usuarios listados aquí en SAMBA. Si queremos especificar un grupo, en lugar de un usuario, pondremos una '@' delante del nombre del grupo. Esta lista es en modo permisivo. Si pongo a Pepe y Juan, todos los usuarios entrarán menos esos dos.
- **valid users** → Mismo formato que arriba. Si está vacía, cualquiera puede entrar (excepto si está en la lista de arriba). Si se pone algún usuario la cosa cambia y SÓLO esos usuarios pueden entrar. Modo restrictivo si tengo a Pepe y Juan, solo podrán acceder ellos dos.

Si algún usuario está afectado por las dos listas se le niega el acceso. Aplica lo más restrictivo. Lo lógico es tener solo una de las listas.

6.3 Configuración de un servidor Samba.

- **admin users** → Usuarios que tendrán todos los privilegios (como si fueran root) en los recursos compartidos (cuidado con esta opción). Esta opción solo se usará en carpetas, no la pondremos en la configuración global.
- **read list** → Lista de usuarios que tendrán permisos de sólo lectura en los recursos compartidos independientemente del atributo read only que haya en el recurso compartido. Si un usuario en local puede leer y escribir en la carpeta, pero lo metemos en la read list, entonces solo podrá leer.
- **write list** → Usuarios que tendrán permiso de lectura y escritura en los recursos compartidos. Esta lista tiene preferencia sobre la de solo lectura, por lo que quien esté afectado por ambas, seguirá pudiendo escribir.

Si el usuario no está en ninguna lista, hará lo que indique el recurso en local. Si la opción por defecto de esa carpeta es solo lectura y quiero que ese usuario escriba, lo pondré en la write list.

6.3 Configuración de un servidor Samba.

- **hosts allow** → lista de máquinas que podrán conectarse (vacía significa que todas). Puede usarse el nombre de la máquina, su ip, o una ip de red (ip/máscara). Se puede especificar después qué máquinas (de esas redes) se excluyen poniendo la palabra EXCEPT seguida de las máquinas.
 - `hosts allow = 192.168.0.0/24 EXCEPT 192.168.0.27`
- **hosts deny** → máquinas que tienen prohibido el acceso. Si se va a usar una política de prohibir a todos el acceso excepto a los que estén en hosts allow, se debe usar la palabra ALL o la dirección 0.0.0.0/0.
 - Tiene prioridad la lista hosts allow sobre esta.
 - Si queremos usar política permisiva (permitir a todos excepto los que estén en hosts deny, se debe dejar la lista de hosts allow vacía)

Ejemplo:

`hosts allow = 192.168.0.0/24 EXCEPT 192.168.0.27`

`host deny = ALL`

6.3 Configuración de un servidor Samba.

COMPARTIR CARPETAS:

- Para cada recurso compartido crearemos una sección dentro del fichero de configuración de SAMBA, con el nombre del recurso compartido entre corchetes (este nombre es el que aparecerá cuando accedamos al recurso compartido, independientemente de como se llame la carpeta en local). Después del nombre y hasta que empiece el siguiente recurso compartido, pondremos las opciones de compartición de ese recurso.
- Las listas **invalid users**, **valid users**, **admin users**, **read list**, **write list**, **hosts allow** y **hosts deny** funcionan exactamente igual que en la configuración global, pero esta vez sólo afectan al recurso compartido (y tendrán preferencia sobre las globales).

6.3 Configuración de un servidor Samba.

COMPARTIR CARPETAS:

[nombre carpeta compartida]

- **comment** → Descripción del recurso compartido.
- **path** → ruta (absoluta) al directorio/carpeta compartido.
- **read only** → valores yes o no. Si activamos esta opción (yes) se compartirá en modo sólo lectura por defecto (excepto para usuarios que estén en la write list). (read only = yes es lo mismo que writable = no)
- **writable** → Equivalente a read only pero contrario en cuanto al significado (yes indica que se puede leer y escribir y no, solo lectura). No tiene sentido usar ambas a la vez. (writable = yes es lo mismo que read only = no)

6.3 Configuración de un servidor Samba.

COMPARTIR CARPETAS:

- **guest ok** → (yes, no). Para admitir o no invitados. Permite que usuarios anónimos puedan acceder (con la cuenta definida en guest account, nosotros pusimos nobody, que es la cuenta por defecto de invitado de linux). Si la seguridad es diferente de security = share (por defecto el valor es user), necesitamos tener habilitada la opción map to guest (en nuestro caso “bad user”) en globales para que alguien pueda loguearse de forma anónima.
- **browseable** → (yes, no). Para que aparezca la carpeta en el listado de recursos compartidos, cuando me conecte a la máquina con el servidor samba. Si le pongo que no, la carpeta existirá pero no la veré y a menos que sepa que existe y el nombre que tiene, no sabré que está.
- **available** → (yes,no). No se pone porque por defecto es que si, pero si se pone y le ponemos no, la carpeta dejará de estar compartida.

6.3 Configuración de un servidor Samba.

COMPARTIR CARPETAS:

Ejemplo de configuración de carpeta compartida:

[share]

writeable = yes

path = /home/share/samba-share/

admin users = sambauser

valid users = sambauser usuario1

read list = usuario1

available = yes

6.3 Configuración de un servidor Samba.

AÑADIR USUARIOS EN SAMBA:

- Podemos añadir un usuario existente en el sistema a SAMBA asignándole una contraseña con el comando:
 - **smbpasswd -a usuario**
- La contraseña no tiene por qué ser la misma que la asignada para la autenticación en el sistema (/etc/shadow), aunque si se activa la opción `unix password sync = yes`, sí que se mantendrá la misma contraseña para SAMBA y login de usuario.

6.4 Ejemplo práctico.

EJEMPLO DE CARPETA CON ACCESO ANÓNIMO:

- Una vez instalado SAMBA con los paquetes adecuados, editamos el fichero de configuración:
 - `sudo nano /etc/samba/smb.conf`
- Añadiremos los nuevos recursos compartidos al final del archivo. Podemos salir guardando con Ctrl-x → s → enter.
- Pondremos estos parámetros configuración en la sección [global]. Busca primero a ver si aparecen y modifica (o descomenta quitando la almohadilla de delante '#'). Si no aparecen entonces añade la opción (dentro de [global]). En nano se puede buscar con Ctrl+w.

6.4 Ejemplo práctico.

EJEMPLO DE CARPETA CON ACCESO ANÓNIMO:

- Modificamos en la sección [global] los parámetros que vimos (mirar antes lo que hay por defecto por si la opción es correcta):

```
[global]
workgroup = WORKGROUP
server string = Samba Server %v
netbios name = servidor_samba
security = user
map to guest = bad user
dns proxy = no (puede que no aparezca)
```

6.4 Ejemplo práctico.

EJEMPLO DE CARPETA CON ACCESO ANÓNIMO:

- Para configurar un acceso anónimo a una de las carpetas deberemos añadir la siguiente sección al final del fichero de configuración:

[Anonymous]

comment = carpeta compartida (descripción recurso compartido: opcional)

path = /samba/anonymous (ruta absoluta carpeta compartida)

browsable = yes (para que aparezca en la lista de recursos compartidos)

writable = yes (carpeta con permiso de escritura, siempre que el usuario pueda)

guest ok = yes (permite invitados)

~~read only = no~~ (esta opción no hace falta, porque ya está writable = yes)

available = yes (no es obligatorio porque por defecto es que si)

force user = nobody (si no lo ponemos, entraremos con nuestro usuario)

La última opción indica que aunque entre otro usuario (previo login), se le asignará el usuario nobody en esa carpeta (usuario de la cuenta de invitado por defecto).

6.4 Ejemplo práctico.

EJEMPLO DE CARPETA CON ACCESO ANÓNIMO:

- Creamos la carpeta que queremos compartir:
 - `sudo mkdir -p /samba/anonymous`

En estos momentos la carpeta tiene como usuario y grupo a root, y además root es el único que puede escribir en ella. Aunque yo le dijera en samba que otro usuario puede escribir, este usuario no podría hacerlo porque los permisos de Linux no lo permitirían. En Windows pasa lo mismo, para poder acceder a una carpeta como lectura y escritura tanto los permisos locales como los permisos en red deben permitirlo. Siempre aplica la configuración más restrictiva.

6.4 Ejemplo práctico.

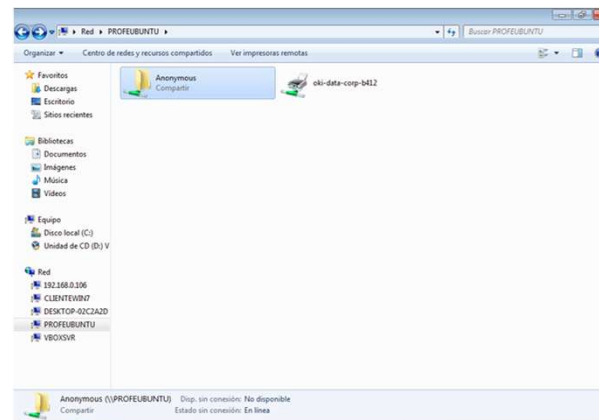
EJEMPLO DE CARPETA CON ACCESO ANÓNIMO:

- Y le asignamos el propietario **nobody** y el grupo **nogroup**, que tienen las distribuciones de Linux para el acceso de invitados o usuarios anónimos. Además de conceder a todo el mundo todos los permisos en local (que podremos limitar en red via SAMBA)
 - **chown -R nobody:nogroup /samba/anonymous**
 - **sudo chmod 777 /samba/Anonymous**
- Recordad que hay que reiniciar el servicio de SAMBA después de modificar el fichero de configuración, si no lo habéis hecho ya.
 - **sudo systemctl restart smb**

6.4 Ejemplo práctico.

EJEMPLO DE CARPETA CON ACCESO ANÓNIMO:

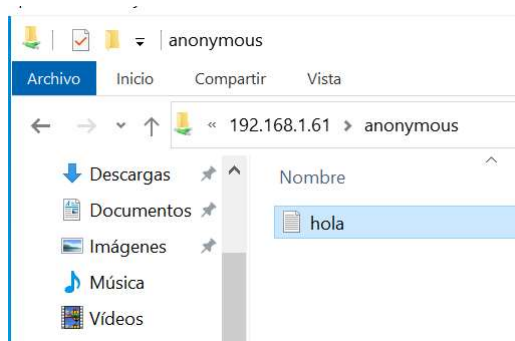
- Para acceder a la máquina con SAMBA, en el explorador de archivos de Windows escribid la ruta \\IP (Siendo IP la dirección de red de la máquina Linux). Se puede consultar la IP de la máquina Linux con el comando **ifconfig** o **ip a**.
- También se puede usar el nombre netbios que se puso en la configuración de SAMBA, pero puede tardar mucho, mejor la IP.
- Comprobaremos que está la carpeta que hemos creado y compartido en Linux.



6.4 Ejemplo práctico.

EJEMPLO DE CARPETA CON ACCESO ANÓNIMO:

- Crearemos un archivo desde Windows, dentro de la carpeta y comprobamos que se ha creado en Linux.
- Cosas a tener en cuenta:
 - No nos ha pedido usuario ni contraseña para acceder, pero como la carpeta admite invitados, hemos podido entrar.
 - Hemos podido crear un archivo porque en los permisos de la carpeta (local) y en el archivo de samba (red) teníamos habilitada la opción de escritura.



```
marina@marina-VirtualBox:/samba$ ls -l anonymous/  
total 0  
-rwxr--r-- 1 marina marina 0 abr 16 10:46 hola.txt
```

6.4 Ejemplo práctico.

EJEMPLO DE CARPETA CON ACCESO DE USUARIO:

- Para limitar el acceso a las carpetas por usuarios o por grupos de usuarios debemos de hacer uso de los grupos y de los usuarios del sistema. En el siguiente ejemplo vamos a crear un grupo llamado usertest con un usuario test1, al cual le asignaremos permisos de lectura y escritura en la carpeta test.
- Lo primero que debemos hacer es crear el grupo usertest, luego creamos el usuario test1 y lo añadimos al grupo usertest, para terminar, creamos una contraseña samba al usuario test1.
 - **sudo addgroup usertest**
 - **sudo useradd test1 -G usertest**
 - **sudo smbpasswd -a test1**

6.4 Ejemplo práctico.

EJEMPLO DE CARPETA CON ACCESO DE USUARIO:

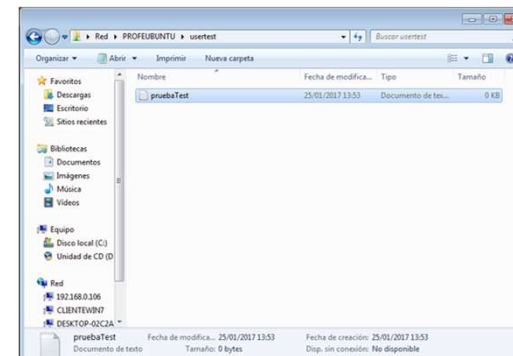
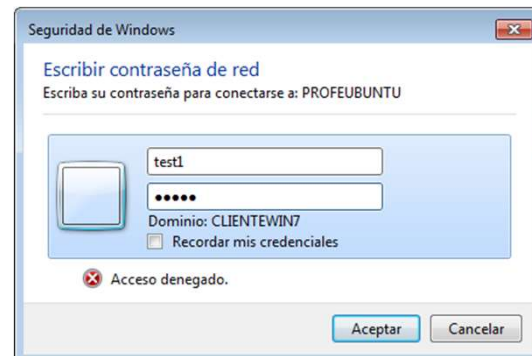
- Ahora creamos la carpeta test que vamos a compartir con todo el grupo. Creamos la carpeta y asignamos los permisos correspondientes.
 - `sudo mkdir -p /samba/test`
 - `sudo chmod -R 0770 /samba/test`
 - `sudo chown root:usertest /samba/test`
- Y ya por último añadimos una nueva sección en el archivo smb.conf

```
[usertest]
path = /samba/test
valid users = @usertest (@ porque es un grupo)
guest ok = no (no se permiten invitados, solo usuarios)
writable = yes
browseable = yes
```

6.4 Ejemplo práctico.

EJEMPLO DE CARPETA CON ACCESO DE USUARIO:

- Reiniciamos el servicio de samba y con el siguiente comando comprobamos que no hay errores de sintaxis en el fichero smb.conf
 - **testparm**
- Ahora ya podemos acceder desde nuestro cliente Windows al servidor Ubuntu. Al acceder a la carpeta test nos pedirá que nos identifiquemos. Si nos identificamos correctamente accederemos a la carpeta test. Crearemos un archivo y comprobaremos que también se ha creado en Linux.



6.4 Ejemplo práctico.

CERRAR SESIÓN:

Cerrar sesión SAMBA desde el servidor (Linux):

- Si queremos acceder desde otro perfil en el cliente, sin tener que cerrar sesión, desde el servidor SAMBA, para ver las conexiones remotas a nuestros recursos compartidos usamos el siguiente comando:
 - **sudo net status shares**
- Podemos desconectar a los usuarios de una carpeta remota (para que vuelva a pedir usuario y contraseña) con:
 - **smbcontrol all close-share NOMBRE_COMPARTIDO**

6.4 Ejemplo práctico.

CERRAR SESIÓN:

Cerrar sesión SAMBA desde el cliente (Windows):

- Con el comando **net use** vemos los recursos compartidos a los que estamos conectados.
- Para borrar una entrada de la tabla (el recurso del que nos queremos desconectar). Introducimos el comando:
 - **net use /delete \\IP SERVIDOR\nombre carpeta compartida**

Puede darse el caso de que no funcione y tengamos que reiniciar la máquina Windows para poder entrar con otro usuario