

Servidor Ftp proFTPD

José Domingo Muñoz Rodríguez <josedom24@gmail.com>
Alberto Molina Coballes <alberto.molina@hispalinux.es>

23 de septiembre de 2006

Resumen

En este documento vamos a explicar los pasos para instalar, configurar y poner en funcionamiento el servidor ftp proFTPD. La utilización del servidor ftp nos va permitir la gestión de forma remota de los sitios web creados anteriormente. Nos vamos a centrar en la utilización de usuarios virtuales cuya información vamos a guardar en una base de datos MySQL.

©José Domingo Muñoz Rodríguez y Alberto Molina Coballes. Algunos Derechos reservados.

Esta obra está bajo una licencia Attribution-ShareAlike 2.5 de Creative Commons. Para ver una copia de esta licencia, visite:
<http://creativecommons.org/licenses/by-sa/2.5/>

1. Introducción al servidor Ftp proFTPD

ProFTPD es un servidor FTP. Se promociona desde su página web como estable y seguro, cuando se configura correctamente. El servidor ProFTPD se promociona a sí mismo como un “Software servidor FTP altamente configurable con licencia GPL” (*Highly configurable GPL-licensed FTP server software*).

2. Instalación de MySQL

Como medida de seguridad vamos a utilizar usuarios virtuales para el acceso a los ficheros mediante el protocolo ftp. Normalmente el servidor proFTPD utiliza los usuarios reales del sistema para el acceso a los archivos, pero esto puede dar problemas de seguridad. Utilizando usuarios virtuales evitamos, entre otras cosas, que si un usuario malicioso logra una cuenta FTP, tenga menos posibilidades de entrar en nuestro sistema.

Normalmente la información de los usuarios virtuales se guarda en un fichero de texto plano, nosotros vamos a explicar la configuración del servidor para guardar la información de los usuarios virtuales en una base de datos construida en MySQL. Para instalar el servidor de base de datos MySQL utilizamos el comando apt-get:

```
apt-get install mysql-server
```

Durante el proceso de instalación se bajarán e instalarán todos los paquetes para que funcione MySQL. Una vez terminado la instalación, lo primero que tenemos que hacer es cambiar la contraseña al usuario root de MySQL:

```
mysqladmin -u root password nueva_password
```

3. Instalación de proFTPD

Para instalar el servidor proFTPD, utilizamos el comando apt-get de la siguiente forma:

```
apt-get install proftpd
```

Durante la instalación debemos escoger la forma en que se va ejecutar este servicio, si lo hará bajo inetd o en forma independiente (como un proceso más). La diferencia entre ambos métodos es la velocidad de ejecución que se obtendrá del servicio y la carga que se le dará al equipo. Utilizando inetd el equipo iniciará un proceso nuevo por cada petición ftp que llegue al equipo, mientras que ejecutándose en forma independiente, el proceso ya está inicializado por lo que no hay necesidad ejecutar un nuevo programa (simplemente se clona el proceso, generando una nueva instancia del mismo para atender al cliente ftp que inició el requerimiento). Nosotros escogeremos la opción *independiente*.

4. Creación de la base de datos

ProFTPD, de forma normal, valida sobre el `/etc/passwd` y `/etc/shadow`, por tanto necesita una tabla que simule los datos contenidos en estos ficheros.

También utiliza el `/etc/group`, por lo que necesitamos otra tabla del mismo modo:

Tabla de usuarios:

- `username`: Nombre del usuario
- `password`: Contraseña del usuario
- `uid`: Identificador del usuario
- `gid`: Identificador del grupo al que pertenece el usuario
- `homedir`: Directorio donde tendrá acceso el usuario
- `activa`: Este campo lo usamos para activar y desactivar cuentas sin borrar el usuario¹

Tabla de grupos:

- `groupname`: Nombre del grupo
- `gid`: Identificador del grupo
- `members`: Nombres de los usuarios que pertenecen a este grupo

Como se puede ver es todo muy similar al `/etc/passwd` y `/etc/group`. Ahora vamos a ver las instrucciones SQL para crear estas tablas.

Lo primero, creamos la base de datos que vamos a utilizar. Accedemos como `root`:

```
debian:/home/jose\# mysql -u root -p
password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9 to server version: 5.0.24-Debian_1-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database ftpd;
Query OK, 1 row affected (0.05 sec)
```

Ahora generamos un usuario que acceda a la base de datos, para no utilizar el usuario `root`:

```
mysql> GRANT ALL ON ftpd.{*} TO proftpd@localhost \\\
IDENTIFIED BY "nueva_contraseña";

Query OK, 0 rows affected (0.09 sec)
```

Es decir, hemos creado un nuevo usuario (`proftpd`) que se va a conectar desde `localhost`, tiene permiso en todas las tablas de la base de datos que hemos creado anteriormente y cuya contraseña es “`nueva_contraseña`”. Este es el usuario que utilizaremos para crear las tablas anteriormente descritas:

¹Este campo no es necesario, pero lo vamos a utilizar para tener una forma sencilla de activar o desactivar los usuarios ftp. Se puede poner el valor que queramos, nosotros vamos a utilizar el 1 para indicar usuario activo u 0 para desactivar el usuario.

```
debian:/home/jose\# mysql -u proftpd -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9 to server version: 5.0.24-Debian_1-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| ftpd              |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> use ftpd;
Database changed
```

```
mysql> show tables;
Empty set (0.00 sec)
```

Hemos entrado con el usuario creado anteriormente, hemos visualizados las bases de datos, hemos indicado que la base de datos que vamos a utilizar es ftpd y hemos mostrado las tablas, que como podemos observar no hay ninguna. Estamos preparados para crear las tablas:

```
CREATE TABLE usuarios (
username char(12) NOT NULL,
password char(100) NOT NULL,
uid int(6) default NULL,
gid int(6) default NULL,
homedir char(50) default NULL,
shell char(20) default NULL,
activa tinyint(4) default '1' )
TYPE=MyISAM;
```

```
CREATE TABLE grupos (
groupname varchar(20) NOT NULL,
gid int(6) NOT NULL default '65534',
members longtext )
```

```
TYPE=MyISAM;
```

Veamos las tablas que hemos creado y su estructura:

```
mysql> show tables;
+-----+
| Tables_in_ftpd |
+-----+
| grupos          |
| usuarios       |
```

```
+-----+
2 rows in set (0.00 sec)
```

```
mysql> describe usuarios;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null    | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| username   | char(12)      | NO      |      |         |      |
| password   | char(100)     | NO      |      |         |      |
| uid        | int(6)        | YES     |      | NULL    |      |
| gid        | int(6)        | YES     |      | NULL    |      |
| homedir    | char(50)      | YES     |      | NULL    |      |
| shell      | char(20)      | YES     |      | NULL    |      |
| activa     | tinyint(4)    | YES     |      | 1       |      |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.06 sec)
```

```
mysql> describe grupos;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null    | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| groupname  | varchar(20)   | NO      |      |         |      |
| gid        | int(6)        | NO      |      | 65534   |      |
| members    | longtext      | YES     |      | NULL    |      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

5. Configuración del servidor ProFTPD

La configuración del servidor proFTPD está en el fichero `/etc/proftpd.conf`, la sintaxis que utiliza es parecida a la utilizada en apache, por lo que el usuario acostumbrado a configurar apache le resultará familiar. La configuración que viene por defecto en el fichero nos viene bien, lo único que vamos a hacer es introducir nuevas directivas para que el servidor proFTPD valide los usuarios utilizando la información guardada en nuestra base de datos MySQL. Las directivas que vamos a introducir en el fichero de configuración son las siguientes:

- *AuthPAM off*: Para deshabilitar la autenticación PAM, es decir para que no autentique a los usuarios reales.
- *RequireValidShell off*: Requiere una Shell válida para iniciar sesión. Las shell válidas son las que están en `/etc/shells`.
- *DefaultRoot ~*: Para que cuando un usuario entra en el servidor ftp, no pueda salir de su home a directorios superiores.

Y por último nos queda las directivas de configuración para que proFTPD autentique los usuarios virtuales cuya información vamos a guardar en las tablas de nuestra base de datos. Antes de ver las directivas vamos a explicar cada una de ellas:

- *SQLBackend*: Con esta directiva indicamos el servidor de base de datos que vamos a utilizar. En nuestro caso mysql.

- *SQLAuthTypes*: En tipos de identificación tenemos (entre otras) PlainText (texto plano), Crypt (como en el /etc/shadow) o Backend (utiliza el sistema propio de la base de datos). Aquí depende de con qué vayamos a cifrar las contraseñas en la base de datos. Si lo haces con CRYPT() usa Crypt, si usas PASSWORD() usa Backend.
- *SQLAuthenticate*: Con esta directiva decimos quien tiene que identificarse en MySQL. Podemos seleccionar grupos o usuarios, yo me limito a decir que todos: on.
- *SQLConnectInfo*: Se indica los datos de la conexión a la base de datos. base_de_datos@host usuario contraseña
- *SQLDefaultGID* y *SQLDefaultUID*: Indican, respectivamente, los GID y UID que se usan por defecto en caso de no poner nada en la tabla.
- *SQLMinUserGID* y *SQLMinUserUID*: Indican el UID y el GID mínimo que han de tener los usuarios para poder loguear.
- *SQLGroupInfo*: Se indica el nombre de la tabla que posee la información de los grupos, así como los nombres de los campos.
- *SQLUserInfo*: Se indica el nombre de la tabla que posee la información de los usuarios, así como los nombres de los campos².
- *SQLUserWhereClause*: Indica una condición a la hora de seleccionar los campos. Nosotros la usamos para controlar si una cuenta está activa o no³.
- *SQLLogFile*: Se indica el archivo de log donde se guarda la información generada por el servidor al acceder a la base de datos.

Veamos las directivas que hay que poner en el fichero de configuración⁴:

```
SqlBackend mysql
SQLAuthTypes Backend
SQLAuthenticate on
SQLConnectInfo ftpd@localhost proftpd nueva_password
SQLDefaultGID 65534
SQLDefaultUID 65534
SQLMinUserGID 500
SQLMinUserUID 500
SQLGroupInfo grupos groupname gid members
SQLUserInfo usuarios username password uid gid homedir shell
SQLUserWhereClause "activa=1"
SQLLogFile /var/log/proftpd_mysql.log
```

²Esto es necesario para que el servidor pueda construir la instrucción SQL que le permite obtener la información del usuario. En la lista de campo no hay que poner el campo activa.

³Con la información de esta directiva y de la anterior(SQLUserInfo) el servidor puede crear la instrucción sql que le permite acceder a los datos del usuario:

```
SELECT username, password, uid, gid, homedir, shell FROM usuarios WHERE (username='nombre') and ((activa=1)) LIMIT 1
```

⁴Yo uso SQLAuthType Backend porque cifro las contraseñas con PASSWORD(). Además, ponemos el SQLDefaultUID/GID a 65534 porque son los que indican el usuario nouser y el grupo nogroup (echad un ojo al /etc/passwd y al /etc/group para comprobarlo).

Bueno ya hemos terminado la configuración, para continuar tenemos que reiniciar el servicio:

```
/etc/init.d/proftpd restart
```

6. Creación de usuarios virtuales

En este apartado vamos a seguir el ejemplo que planteamos en el capítulo del servidor Apache, donde teníamos dos sitios virtuales (`www.pagina1.com` y `www.pagina2.com`) en dos directorios distintos (`/srv/www/pagina1` y `/srv/www/pagina2`). En este capítulo continuamos con el ejemplo creando dos usuarios virtuales (`user1` y `user2`) que van a tener acceso por FTP a cada uno de los espacios web para la gestión remoto de las páginas webs.

Lo primero que vamos a hacer es añadir la información necesaria de los usuarios en las tablas de la base de datos. Vamos a crear un grupo virtual `ftpusers` al que le vamos a hacer corresponder un identificador de grupo ficticio, por ejemplo el 6000, además vamos a crear dos usuarios: `user1` y `user2`, con contraseñas `passuser1` y `passuser2`, con identificadores de usuarios 5000 y 5001, estos dos usuarios van a pertenecer al grupo `ftpusers` por lo que su campo `gid` va a ser el 6000, además el campo `homedir` será para cada uno el correspondiente a los dos directorios de los sitios web virtuales (`/srv/www/pagina1` y `/srv/www/pagina2`) y por último el campo `shell` será `/bin/false`, indicando que no van a poder logearse en el sistema como usuario reales. Para ello tenemos que ejecutar las siguientes instrucciones SQL:

```
INSERT INTO `grupos` VALUES ('ftpusers', 6000, 'user1,user2');
```

```
INSERT INTO `usuarios` VALUES ('user1', PASSWORD('passuser1'),  
5000, 6000, '/srv/www/pagina1', '/bin/false', 1);
```

```
INSERT INTO `usuarios` VALUES ('user2', PASSWORD('passuser2'),  
5001, 6000, '/srv/www/pagina2', '/bin/false', 1);
```

Para comprobar que la inserción ha sido correcta, podemos ejecutar el siguiente comando:

```
mysql> select * from usuarios;
```

```
+-----+-----+-----+-----+-----+-----+-----+
| username | password          | uid  | gid  | homedir          | shell          | activa |
+-----+-----+-----+-----+-----+-----+-----+
| user1    | 5226b4b67c3db6de | 5000 | 6000 | /srv/www/pagina1 | /bin/false    | 1      |
| user2    | 5226b2b77c3db4df | 5001 | 6000 | /srv/www/pagina2 | /bin/false    | 1      |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

Podemos comprobar como el campo `password` ha sido cifrado. Por otro lado, modificando el campo `activa` a 0 podemos desactivar el usuario de una manera muy fácil.

El último paso que tenemos que hacer es poner como propietario de los directorios donde tenemos nuestra páginas webs a los dos usuarios respectivamente, como root tenemos que ejecutar lo siguiente:

```
debian:~\# cd /srv/www
debian:/srv/www\# chown -R 5000:6000 pagina1
debian:/srv/www\# chown -R 5001:6000 pagina2
```

```
debian:/srv/www\# ls -al
total 16
drwxr-xr-x 4 www-data www-data 4096 2006-08-17 16:10 .
drwxr-xr-x 3 root root 4096 2006-08-17 16:10 ..
drwxr-xr-x 2 5000 6000 4096 2006-08-17 16:33 pagina1
drwxr-xr-x 2 5001 6000 4096 2006-08-17 16:10 pagina2
```

Se puede observar como en ningun momento hemos empleado los nombres de los usuarios o el nombre del grupo, es más si listamos el directorio /srv/www veremos que los directorios pagina1 y pagina2 pertenecen al usuario 5000 y 5001 del grupo 6000, esos usuarios no son usuario reales del sistema.

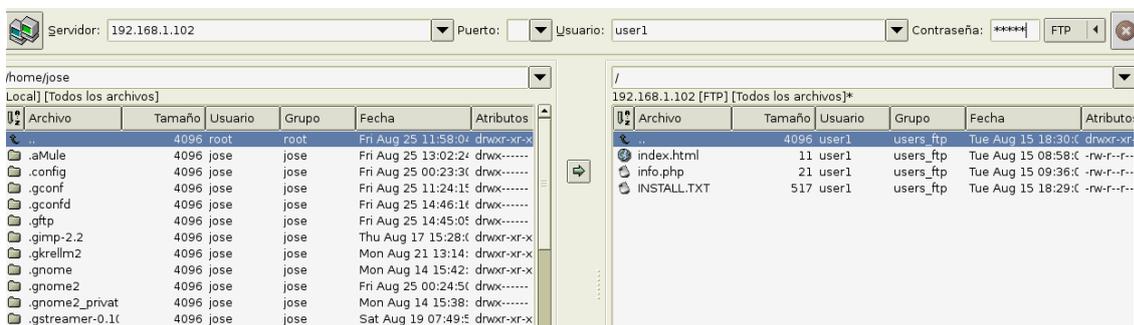
7. Acceso remoto usando un cliente FTP

Bueno para probar que el sistema está funcionando correctamente podemos hacer uso del cliente ftp de Linux:

```
debian:~\# ftp localhost

Connected to localhost.localdomain.
220 ProFTPD 1.2.10 Server (Debian) [127.0.0.1]
Name (localhost:jose): user1
331 Password required for user1.
Password:
230 User user1 logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Si utilizamos un cliente ftp gráfico, por ejemplo gftp, podemos apreciar como ahora si se indica el nombre de usuario y de grupo del propietario de los ficheros.



8. Conclusión

En este documento se ha abordado la instalación y configuración de un servidor FTP. El servidor elegido ha sido el proFTPD que nos permite la utilización de usuarios virtuales. La información de estos usuarios ha sido guardada en una base de datos MySQL.

Referencias

- [1] Documentación del servidor proFTPD:
<http://www.proftpd.org/docs/>
- [2] Documentación del módulo mod_sql:
http://www.proftpd.org/docs/directives/linked/config_ref_mod_sql.html
- [3] ProFTPD module mod_sql:
http://www.castaglia.org/proftpd/modules/mod_sql.html
- [4] Configuración de ProFTPD+PostFix validados en MySQL:
http://max.bandaancho.st/docs/old_postfix_proftpd.html
- [5] ProFTPD con MySQL:
<http://alf.dyndns.ws/numero1/articulo03.php>