



BULMA

Bisoños Usuarios de Linux de Mallorca y Alrededores | Bergantells Usuaris de Linux de Mallorca i Afegitons



Configuración de un servidor de FTP - Módulo 4.1.1

Por **Daniel Rodriguez**, *DaniRC* (<http://bulma.net/~danirc/>)

Creado el 30/05/2002 21:58 y modificado por última vez el 30/05/2002 21:58

Estoy preparando unos cursos de Linux, y necesitaba material didáctico. La verdad es que a veces es difícil explicar las cosas que uno sabe tan claramente como desea.

Como los chicos de Mandrake lo han hecho muy bien, estoy traduciendo sus manuales, con algun aporte personal. De momento os dejo este módulo, pero los otros iran cayendo uno a uno a intervalos de 2 o 3 días cada uno.

No son como un curso de verdad, porque no daré soporte de los artículos, pero para muchos, puede ser un buen principio!.

Esperando que os guste, aqui va la primera entrega.

PD: He intentado contactar con Mandrake para hacerles llegar la traducción, pero el proyecto MandrakeCampus parece muerto.

Nivel: Avanzado

Documento original: www.mandrakecampus.com

Traducción al castellano: Daniel Rodríguez.

Documento realizado con: OpenOffice 1.0

Licencia: LGPL

Tema 4 - Servicios del Servidor

Subtema 4.1 – Servidor de Ficheros

Modulo 4.1.1 – Configuracion de un servidor FTP

Indice

1. Introduccion

2. Abriendo un FTP anonimo
3. Creación de cuentas especificas de FTP
4. Ejecucion de comandos desde FTP
5. Introduccion al ProFTPD
6. Configuracion de ProFTPD
7. Configuracion de un FTP anónimo
8. VirtualHost
9. Creacion de cuentas de FTP
10. Limitar el ancho de banda
11. Mas informacion

Presentacion general

El protocolo FTP es:

- Usado para publicar en Internet
- No es seguro
- No tiene capacidades de filtrado
- Usa un sistema de autentificacion basado en ficheros de passwords

El protocolo FTP permite transferir ficheros de una maquina a otra. Existen principalmente dos tipos de acceso: el acceso al directorio personal del usuario y el acceso dentro de una jerarquia de ficheros destinada al FTP

Existen varios servidores que pueden realizar esta tarea. Veremos dos de ellos, a sabiendas que las opciones que proponen se encuentran tambien en el resto de los servidores existentes.

Después de haber estudiado los ficheros de configuracion de estos dos ejemplos, no deberia ser dificil llevar estos conceptos a otros servidores.

Veremos en primer lugar el wu.ftpd, instalado por defecto en la mayoria de distribuciones y el proftpd, el mas practico y utilizado hoy por hoy.

Autorizar el acceso a los usuarios

Lo mas prudente es confiar el arranque del demonio al inetd, puesto que esta opcion permite que implementemos reglas de filtrado usando el TCPwrapper.

La autentificación se realiza introduciendo el nombre de usuario y la contraseña: es decir, la validación se realiza del mismo modo en que se realiza un "login" en el sistema, verificando el fichero /etc/passwd (o el /etc/shadow)

Abrir un FTP anonimo

El FTP anonimo permite el acceso sin contraseña. Esta clase de FTP permite poner ficheros al alcance de todos los usuarios del sistema. No precisa de una autentificación precisa, por eso lo llamamos acceso anonimo, y no ofrece acceso mas que a una zona bien especifica del sistema.

Cuando se trata de un acceso FTP por parte de un usuario valido, este usuario puede recorrer los directorios, salir de su directorio personal y copiar ficheros presentes dentro de /etc o /var

En el caso de un FTP anonimo, el servidor utiliza una funcionalidad llamada "entorno protegido" ou chroot.

Esta encierra al usuario en una jerarquia especifica de directorios. Si damos acceso al usuario anonimo dentro de /home/ftpd, este directorio sera tomado como directorio raiz. Le sera imposible por tanto al usuario anonimo recorrer la jerarquia de directorios en sentido ascendente. Si marcamos como directorio raiz el /home/ftpd ... no se puede subir mas arriba de la raiz, por tanto no se tendra acceso a /home ni a cualquier otro directorio por encima de ese en la jerarquia del sistema.

Esto tambien significa que el programa no tendra acceso a los ejecutables del /bin, por ejemplo, y que por tanto debiera de tener una copia del mismo en el interior del directorio "raiz virtual"

Del mismo modo para ejecutar programas, debiera disponer de una copia accesible de todas las bibliotecas que le sean necesarias.

Crear una jerarquia de directorios

Cuando instalamos el paquete wu.ftpd en Linux-Mandrake, este crea un usuario ftp, y todos los ficheros necesarios dentro de /home/ftp. En función a los derechos otorgados a los diferentes directorios, es posible restringir el acceso en modo lectura a ciertas zonas.

Generalmente, colocaremos los ficheros en /pub, es decir en /home/ftp/pub para compartirlos de manera publica.

Autorizar los uploads (es decir, autorizar el envio de ficheros hacia el servidor)

Existen dos formas de autorizar la subida de un fichero al servidor. La primera forma consiste en otorgar derechos de escritura a un directorio que se llama /incoming. (/home/ftpd/incoming)

El usuario podra entonces enviar su fichero, pero no podra ver su contenido, ni recuperarlo mas tarde.

El otro metodo es bastante mas recomendado. Hay que editar el fichero de configuracion del ftpd, /etc/ftppass, y crear una entrada para autorizar los uploads.

```
upload /home/ftpd/ no
upload /home/ftpd/incoming yes toto titi 0440
```

Estas lineas no permiten la subida de ficheros al directorio raiz, pero si que autorizan la subida de ficheros dentro del directorio /incoming.

Los ficheros que sean colocados alli por el usuario, perteneceran al usuario toto, que el del grupo titi y los ficheros seran escritos con los permisos 0440

Evidentemente, es necesario que el directorio sea accesible para escritura.

Aviso : Tened cuidado porque cualquier persona puede llenar rapidamente la particion que contenga el directorio /incoming del servidor FTP a fuerza de enviar ficheros lo suficientemente grandes. Es por tanto recomendado colocar este directorio en una particion a parte, si es que estamos de verdad interesados en ofrecer este servicio a nuestros usuarios.

Creacion de cuentas de FTP

Presentacion

A veces es necesario ofrecer al usuario una mayor libertad para colocar sus ficheros en el servidor que la ofrecida por el acceso ftp anonimo. Pero tampoco deseamos ofrecer un acceso via terminal a todos los usuarios. Es entonces cuando aparece la necesidad de crear usuarios exclusivos del FTP

Creacion de usuario de FTP.

No existe con wu.ftpd, una solucion elegante para crear cuentas de FTP de manera simple: no dispone de ningun metodo de autentificacion propia. Es por tanto necesario crearles cuentas de usuario normales y luego limitarles el acceso y cerrarles la posibilidad de acceso al terminal.

Esto se hace en dos fases: cambiar el shell por defecto del usuario por algo de inutilizable que le explique no tiene acceso al sistema (por ejemplo un shell script de aviso, o bien un /bin/false) y editar el fichero /etc/shells para añadir este programa en la lista de los shells (wu.ftpd verifica la existencia del shell o terminal antes de autorizar el acceso)

Por ultimo hay que encerrar a los usuarios en un entorno protegido, para que no puedan desplazarse en la jerarquia de ficheros del servidor. Para ello añadimos en el fichero de configuracion /etc/ftppass el comando:

```
guestuser nombre_usuario
```

La ejecucion de comandos

- El servidor puede ejecutar comandos del cliente FTP
- Esto simplifica la navegacion por los directorios
- Esta opcion presenta problema de seguridad

Concepto

El protocolo FTP tiene los suficientes comandos como para permitir la navegacion por los directorios, asi como el para envio de ficheros y otras cosas. Pero esta bastante limitado si lo comparamos con la riqueza de comandos de Unix, por tanto es a veces adecuado ofrecer al usuario algunas herramientas extra para facilitarles el trabajo.

Desde el punto de vista del cliente, la sintaxis es "direccion exec comando". El servidor entonces ejecutara el comando si este esta disponible dentro de la limitada jerarquia del servidor de FTP.

Ejemplos practicos

Podemos por ejemplo, cuando el servidor es muy grande, realizar de manera periodica un indice de los ficheros que contiene el servidor. Podemos para ello usar el comando locate, y colocar una version de este ejecutable en la jerarquia de ficheros del FTP, o bien hacer un ls -lR y colocar este resultado en la raiz del directorio, si ademas dejasemos acceso a una version de grep accesible via exec, esta maniobra permitiria a los usuarios del ftp buscar mejor los contenidos del directorio.

Presentacion de ProFTPD

wu.ftpd padece una concepcion un tanto simplista del mundo real y por tanto no responde a todas las necesidades actuales. Sobre todo en nuestros dias en que las comunicaciones por red han cambiado enormemente. Fue concebido como un pequeño servidor de intercambio de ficheros, conocio un tiempo de apogeo en el que muchos programadores aportaron rutinas importantes, pero no suele responder a las necesidades de un hospedaje web actual

ProFTPD esta mejor pensado. A pesar de su relativa juventud (y por tanto un tiempo de correccion de errores y detección de bugs inferior), ProFTPD ha sido creado para ofrecer de forma simple la mayor parte de las opciones que con wu.ftpd eran demasiado engorrosas.

Proceso de instalacion.

Actualmente la distribucion de Linux-Mandrake asi como muchas otras ya traen este servidor incorporado como servidor por defecto de FTP. Si no fuera asi, deberias buscar los ficheros .rpm o bajarlos los fuentes de <http://www.proftpd.net>.

Si la distribucion trae los paquetes, estos generalmente son dos. Un paquete para el programa y su documentacion y otro con los ficheros de configuracion para que lo lancemos en modo stand alone o bien a traves del inetd

Si lo que queremos es arrancar el ProFTPD desde el inetd -recomendado- instalaremos los paquetes:

```
proftpd-core-1.x-1.i586.rpm  
proftpd-inetd-1.x-1.i586.rpm
```

Si lo que queremos es arrancar directamente el ProFTPD usaremos el paquete:

```
proftpd-standalone-1.x-1.i586.rpm en lugar del paquete del inetd.
```

Configuracion del ProFTPD.

Una configuracion imbricada.

El fichero de configuracion proftpd.com (habitualmente estara en /etc) contiene una serie de instrucciones que deben ser tomadas en cuenta como instrucciones imbricadas.

Algunos comandos se aplican a la totalidad del servidor, pero es posible concretar la configuracion directorio por directorio.

Los comandos generales

El modo de arranque.

La directiva **ServerType** standalone permite lanzar el demonio directamente, mientras que si cambiamos standalone por inetd nos permitira lanzar el meta-demonio.

Cuidado! Si lo que deseamos es cambiar un servidor de FTP tal que wu.ftpd por ProFTPD hay que pensar en: Comentar la linea correspondiente al servidor de FTP en el inetd.conf
Reemplazar la orden de arranque del wu.ftpd por la correspondiente para el ProFTPD

La identidad del servidor

El servidor de FTP debe disponer de los permisos suficientes para escribir los ficheros en su jerarquia. Las directivas User y Group nos permitiran cambiar la identidad del demonio. Pero **CUIDADO**, esta muy desaconsejado dar permisos en exceso al demonio de FTP por motivos de seguridad del servidor.

Autorizar el reemplazo de ficheros

El fichero de configuracion por defecto contiene una directiva tipica, que permite autorizar o prohibir la

sobreescritura de ficheros (cuando un usuario ponga un fichero en el servidor, si el servidor contiene un fichero con el mismo nombre, este puede tener el derecho de reemplazar el existente por el nuevo)

```
<DIRECTORY /*>  
AllowOverwrite on #Permitir Sobreescritura ON=Permitir/OFF=No permitir  
</DIRECTORY>
```

Este ejemplo representa la estructura típica de la configuración de ProFTPD. El comando Directory permite restringir la aplicación de directivas a un cierto directorio. Las directivas se aplican desde <DIRECTORY ...> hasta el </DIRECTORY> Se trata de un lenguaje etiquetado lo mismo que el HTML.

En este caso en particular, la totalidad de los directorios dentro del directorio raíz son afectados, y la sobreescritura está autorizada.

Limitar el acceso a ciertos usuarios

El comando **Limit LOGIN**

El comando Limit permite limitar el acceso al servidor. Se trata de un comando muy versátil, puesto que permite limitar un gran número de parámetros: la conexión, el envío de ficheros, la descarga de ficheros, el acceso a ciertos directorios, ... Y todo ello utilizando una sintaxis uniforme.

Si lo que deseamos es prohibir la conexión de ciertos usuarios, podremos colocar, en la configuración del servidor:

```
<Limit LOGIN>  
DenyUser usuario1,usuario2...  
</Limit>
```

Si lo que deseamos es permitir la conexión de solo ciertos usuarios, podremos utilizar AllowUser :

```
<Limit LOGIN>  
AllowUser usuario1,usuario2...  
DenyUser all  
</LIMIT>
```

En el interior de una directiva LIMIT, ProFTPD examina en primer lugar las autorizaciones explícitas, y después las prohibiciones explícitas. Si una conexión no responde a ninguno de los 2 criterios es autorizada. Es posible cambiar este comportamiento usando la orden Order deny,allow. Si esta orden está presente, el servidor primero examinará los comandos Deny y luego los comandos Allow, y prohibirá toda conexión que no corresponda a estos patrones.

Nota : Es posible utilizar los comandos AllowAll, AllowGroup, DenyAll, DenyGroup para simplificar la gestión de las reglas de acceso.

Filtrar por dirección IP

La opción Deny dirección (o bien Allow dirección) permite filtrar en función de la dirección IP del cliente. Se utiliza dentro de una directiva Limit. El valor pasado puede tomar los siguientes valores:

- all (prohibido para todas las direcciones) ;
- none (no prohibir nada) ;
- adresse-ip (prohibido para esta IP) ;
- classe-ip. (prohibido para esta clase IP).

De este modo, los comandos:

```
<Limit LOGIN>  
Allow 10.0.0.7 192.168.2.  
Deny all  
</LIMIT>
```

permiten autorizar las conexiones desde la maquina 10.0.0.7 y desde todas las maquinas 192.168.2.x

Nota : Es posible, pero se desaconseja utilizar filtros relativos a nombres de maquinas o de dominios. Pero esto puede traer problemas porque los nombres son sensibles a problemas tecnicos diversos y ataques de todo tipo sobre el servidor de nombres.

Configurar un FTP anonimo

La directiva **Anonymous /path/acceso**

Este comando permite definir las modalidades de acceso a un servidor FTP anonimo. Contiene todas las directivas que se aplican en caso de conexión externa. Hay que facilitarle el path (camino) de acceso al directorio protegido. Podemos inspirarnos en el ejemplo anonymous.conf proporcionado junto a la distribucion para configurar el servidor

Las restricciones de acceso

ProFTPD se lanza con privilegios de root, puede por tanto hacerse pasar por cualquier usuario. Es por ello que es posible definir con que derechos debe operar en cada directorio particular.

Si lo que deseamos es recuperar la configuracion del servidor wu.ftpd, podemos elegir utilizar los derechos del usuario ftp. Lo que permitiria de forma facil, crear una zona en /home/ftp/ con derechos de escritura.

Podemos tambien restringir el acceso en terminos de usuarios conectados. El comando MaxClients fue pensado para ello. Esta restriccion permite limitar la carga del servicio en el procesador. Sin embargo no permite limitar el ancho de banda consumido por el servicio, puesto que el ancho de banda esta compartido con el resto del servidor.

Limitar los comandos : Limit

El comando Limit, permite definir de forma muy fina y elegante los permisos. Es posible limitar cada comando del protocolo FTP:

CWD permite cambiar de directorio. Limitando su uso podemos limitar el acceso a ciertos directorios ;
DIRS es una palabra clave que agrupa todos los comandos que permiten la navegacion por el directorio.
Limita por tanto el conocer el contenido de un directorio en particular;
MKD y RMD permiten respectivamente crear y borrar directorios ;
RETR permite recuperar un fichero del servidor

STOR permite enviar un fichero al servidor

DELE permite borrar un fichero.

Las palabras clave READ y WRITE engloban varios comandos, y limitan respectivamente el acceso a la lectura y a la escritura. De esta forma, si deseamos que los usuarios anonimos no puedan hacer nada mas que leer los ficheros, podriamos usar la directiva :

```
<LIMIT WRITE>
DenyAll
</LIMIT>
```

Creacion de un directorio publico para el almacenamiento de ficheros por parte de los usuarios.

La orden Directory /path/de/acceso permite definir propiedades particulares para una jerarquia de ficheros dada. Hereda las propiedades del entorno, de la misma forma que Anonymous, por ejemplo, hereda el entorno « configuracion global del servidor de FTP ».

Bastara con crear un directorio -incoming, por ejemplo- en el que el acceso para escritura estara permitido.
<DIRECTORY incoming/*>

```
<LIMIT STOR>
```

```
AllowAll
```

```
</LIMIT>
```

```
</DIRECTORY>
```

Los VirtualHosts

La configuracion cambia en funcion de la IP

No es extraño que un servidor disponga de varias direcciones IP. Podemos imaginarnos un servidor FTP en el punto de union entre la red local e Internet: este dispone de una direccion IP publica y otra privada.

Se puede configurar ProFTP para que no responda del mismo modo a las peticiones recibidas en funcion de la IP que realiza la peticion. Atencion, hablamos de direcciones IP y no de nombres de dominio.

La directiva VirtualHost

Por defecto, ProFTPD espera las conexiones en el puerto FTP de todas las direcciones IP del servidor. Gracias a la directiva VirtualHost, es posible especificar elementos de configuracion en funcion de la direccion utilizada. La sintaxis es la siguiente:

```
<VIRTUALHOST direccion.IP>
comando1
comando2
...
</VIRTUALHOST>
```

De este modo es posible hospedar varios sitios FTP en la misma maquina, separando a cada uno por la direccion IP de procedencia de la peticion. Cuidado, la configuracion global del servidor se aplica a todas las direcciones disponibles, las directivas VirtualHost solo definen subconjuntos de la configuracion global.

Es posible también autorizar las conexiones unicamente a direcciones especificas. Hay que proceder en 2 tiempos: prohibir toda conexión al servidor en la configuracion general y despues crear VirtualHost para cada direccion IP a la que deseamos que el servidor atienda.

Para impedir toda conexión al servidor, le podemos decir en la configuración general que escuche el puerto 0, gracias al comando `Port 0`.

Después habrá que indicar en cada uno de los `VirtualHost`, el comando `Port 21` (que es el puerto estándar para FTP).

Con el fin de evitar que el `ProFTPD` no escuche en el puerto 21 de todas las direcciones, es necesario colocar además, en la configuración del servidor el comando `SocketBindTight yes`.

Creación de cuentas de FTP

Crear cuentas de shell, y prohibir la conexión.

El primer método consiste en crear cuentas de usuario y impedir el acceso creando un intérprete de comandos (shell – terminal) inválido. Como se vio cuando hablabamos del `wu.ftp`

El comando `AuthUserFile`

Otra solución, más versátil y elegante, consiste en crear servidores virtuales y definir para cada uno de ellos un fichero de contraseñas específico.

Usaremos para ello el comando `AuthUserFile /path/fichero`, que permite definir un fichero de contraseñas específico de cada servidor virtual. Si usamos `shadow passwords`, el método más simple consiste en: Crear un usuario. Esto crea una entrada en el fichero de configuración `/etc/passwd` y otra entrada en `/etc/shadow`

Podemos dar a todos los usuarios el mismo UID y el mismo GID, los del usuario de `ftp` por ejemplo.

Copiar las entradas correspondientes de `/etc/passwd` en `/etc/proftpd.virtualhost`, fichero que utilizaremos de cara a la autenticación de los usuarios.

Ahora copiamos el `password` cifrado, que encontraremos en `/etc/shadow` y lo colocaremos en el lugar del * del fichero `/etc/proftpd.virtualhost`

Por último borramos los usuarios de los ficheros `/etc/passwd` y `/etc/shadow` del sistema.

De esta forma hemos obtenido un fichero de autenticación utilizable por el `ProFTPD`. Cada usuario dispone entonces de un acceso individual.

Para evitar que accedan a directorios de otros usuarios (puesto que les hemos puesto a todos la misma UID), podemos usar el comando `DefaultRoot ~`, que hará de su directorio personal un directorio protegido. Es decir, hará que su directorio personal sea para ellos como el directorio raíz. Como ya explicamos anteriormente hablando del `wu.ftpd`.

Limitación del ancho de banda

TCP/IP no lo permite a nivel de protocolo.

Un servicio FTP publico muy frecuentado puede consumir una gran cantidad de ancho de banda. Es natural que deseemos limitarlo para garantizar que las otras aplicaciones del sistema pueda desenvolverse de forma natural. Por desgracia el protocolo TCP/IP no contempla la regulacion del ancho de banda: esta es simplemente repartida entre todas las conexiones.

Es posible limitar el numero de conexiones con la directiva MaxClients, pero es una medida inutil ante el abuso por parte de los usuarios, puesto que basta un solo usuario para ocupar todo el ancho de banda disponible.

Sin embargo el servidor tiene un "truco" puede ser limitado en cuanto a capacidad de lectura escritura.

ProFTPD tiene un mecanismo de limitacion de ancho de banda: solo puede leer un cierto numero de bytes por segundo y por conexión, evitando asi todo riesgo de saturacion. Usamos para ello el comando RateReadBPS n, donde n es la tasa de transferencia maxima en bytes/seg.

Del mismo modo existe el comando RateWriteBPS n que permite limitar la velocidad de escritura de los ficheros.

Favorecer las transferencias pequeñas

Es posible configurar con mas precision el uso del ancho de banda. Podemos desear en efecto, permitir la transferencia rapida de ficheros pequeños, limitando a la vez el ancho de banda usado para la transferencia de ficheros grandes.

Los comandos RateReadFreeBytes y RateWriteFreeBytes nos permiten definir las condiciones bajo las cuales no se aplican las limitaciones de ancho de banda.

Para profundizar:

Referencias bibliograficas

<http://www.wu-ftp.org> Site oficial del servidor wu.ftpd.

<http://www.landfield.com/wu-ftp> Un site-portal sobre wu.ftpd.

<http://www.proftpd.net/docs/index.html> Lista de las directivas de configuracion de ProFTPD, ejemplos de configuracion, un manual excelente ...

<ftp://ftp.univ-lyon1.fr/pub/rfc/9xx/959.gz> Descripcion del protocolo FTP.

E-mail del autor: danirc@bulma.net

Podrás encontrar este artículo e información adicional en:

<http://bulmalug.net/body.phtml?nIdNoticia=1344>