

HOWTO Proftpd

De Gentoo Linux Wiki

Este artículo es parte de las series de **HOWTO**.

Kernel & Hardware • Red y Servicios • Portage • Relacionado con el Sistema • Servidor X • Juegos • Misceláneos

Tabla de contenidos

- 1 Introduccion
- 2 Instalacion
- 3 Configuracion
 - 3.1 Fichero de configuracion proftpd.conf
 - 3.2 Runlevel
 - 3.3 Iptables
- 4 Uso del servidor Proftpd
 - 4.1 Arranque/Parada/Rearranque
 - 4.2 Creacion/Eliminacion de cuentas
 - 4.3 Control sobre los usuarios
 - 4.4 Cierre por mantenimiento
- 5 Documentacion extra
- 6 Créditos

Introduccion

Existen varios servidores de ftp para linux, la seleccion del proftpd ha sido debido a que nunca he tenido problemas con él y su funcionamiento siempre ha sido el correcto. Debemos de tener en cuenta que el ftp realiza las conexiones mediante texto plano y por tanto nuestras cuentas son enviadas en texto plano por la red y por tanto la seguridad de estas es minima, por lo que, para informacion muy delicada seria conveniente que usen sftp(openssh). Una vez comentado esto comenzaremos con la instalacion de este magnifico servidor ftp.

Instalacion

El proftpd se encuentra para la gran mayoría de plataformas, por lo que casi seguros que dispondremos de este gran servidor en nuestro "portage":

```
-----  
# emerge proftpd  
-----
```

Configuracion

Fichero de configuracion proftpd.conf

Los ficheros de configuración del proftpd se encuentran en **/etc/proftpd/**, como veremos tendremos creado un fichero llamado proftpd.conf.sample, el cual es un fichero de ejemplo para ayudarnos a configurar nuestro servidor. Nos vamos a centrar en el fichero principal de configuración **proftpd.conf**, el cuál no estará creado, por lo que deberemos crearlo:

```
-----  
# touch /etc/proftpd/proftpd.conf  
-----
```

Una vez creado comenzaremos la edición del fichero, este se divide en 3 partes. La primera trata de la configuración del servidor, la segunda los directorios donde tendremos nuestro servidor ftp físico y por ultimo la creación de un servidor anonymous(esta ultima parte se puede omitir si no es nuestro deseo):

- 1ªParte:

Indicaremos el nombre de nuestro servidor FTP

```
ServerName      "Nombre_del_servidor"
```

Tipo de arranque del demonio, si deseamos standalone o inetd:

```
ServerType      standalone
```

Mensaje de entrada que mostrara nuestro servidor, antes de la autentificacion, Tener en cuenta que en este tipo de opciones, si eligimos off no deberemos de poner acontinuacion el texto que queremos mostrar("mensaje"), esto ocurre con todas las opciones parecidas a esta.

```
DeferWelcome    off/on "mensaje"
```

Nombre identificativo que aparecerá cuando accedamos al ftp

```
ServerIdent     off/on "nombre"
```

Toma las opciones predeterminadas de un servidor ftp

```
DefaultServer   off/on
```

Si queremos que muestre la dirección física de nuestro enlace dinamico. Si dicho enlace esta fuera del directorio destinado al ftp, su acceso no sera posible.

```
ShowSymlinks    off/on
```

Permite continuar una descarga o subida de un fichero, en la cual la transferencia no se completo. A esta opción se le conoce como "resume"

```
AllowRetrieveRestart  yes/no
```

Permite la continuación de una escritura que no se realizo completamente, va unida a la opción anterior

```
AllowStoreRestart    yes/no
```

Permite la falsificacion del grupo de nuestros ficheros, para el cliente el grupo de los ficheros siempre sera el indicado en esta opcion

```
DirFakeGroup       on nombre_grupo_falso
```

Misma funcion que la opcion anterior pero para falsificar el usuario

```
DirFakeUser        on nombre_usuario_falso
```

Falsifica los permisos de nuestros ficheros indicando que todos los ficheros tienen los permisos que le otorguemos, sigue la nomenclatura numerica del chmod, 1 es ejecucion, 2 es escritura y 4 lectura para el 2bit(usuario), 3 bit(grupo) y 4bit(otros usuarios que no sean el propio dueño del fichero ni un usuario del grupo). El primer bit es un bit especial que normalmente no se usa, esta relacionado con el bit pegajoso(valor 1) y valores referentes al "sudo", dejar en 0. El valor X representa un numero dependiendo de nuestra eleccion.

```
DirFakeMode      0XXX
```

Numero maximo de segundos que puede estar un cliente sin realizar subida ni bajada de informacion en el ftp. Una vez sobrepasado el limite cierra la conexion con el cliente.

```
TimeoutNoTransfer      X
```

RL numero maximo de segundos que puede estar cliente-servidor sin recibir informacion de una transferencia ya sea subida o bajada. Es el tiempo limite que dejamos a una subida o bajada cuando se produce un corte y el servidor no recibe informacion, una vez sobrepasado el limite cancela la operacion de transferencia

```
TimeoutStalled      X
```

Numero maximo de segundos que puede estar un cliente en el ftp, sin realizar movimientos, ya sea desplazarse por el ftp como subir o bajar informacion. Una vez sobrepasado el limite cierra la conexion con el cliente.

```
TimeoutIdle      X
```

Pequeño plugin de filtro del propio proftpd

```
DenyFilter      \*.* /
```

Indicamos el puerto donde queremos que escuche nuestro servidor(el 21 es el predeterminado)

```
Port      X
```

Numero de conexiones al ftp que se pueden hacer a la vez(recomendable numero bajo)

```
MaxInstances      X
```

Indicamos el mensaje de acceso realizado correctamente:

```
AccessGrantMsg      "mensaje"
```

Indicamos el mensaje de acceso realizado incorrectamente:

```
AccessDenyMsg      "mensaje"
```

Especificamos la manera que queremos que se "logueen" un tipo de conexion:

```
LogFormat      nombreformatolog "forma de loguear"
```

Especificamos donde guardamos fisicamente los logs, ademas del TIPO que seleccionemos: WRITE(escritura), READ(lectura), AUTH(accesos) y ALL(todos los anteriores)

```
ExtendedLog      /directorio/fichero TIPO nombreformatolog
```

Existe un tipo de logeo que esta ya descrito con un LogFormat y TIPO predeterminado. Este tipo guarda todas las transferencias realizadas:

```
TransferLog      /directorio/fichero
```

Hacemos chroot en el directorio de nuestro ftp, es decir, no permitiremos que pueda subir a un directorio gerarquicamente superior a él

```
DefaultRoot     ~
```

Le indicamos donde esta el fichero donde tenemos guardados los usuarios que tienen acceso a nuestro ftp

```
AuthUserFile    "/etc/passwd"
```

Le indicamos donde esta el fichero donde tenemos guardados los grupos que tienen acceso a nuestro ftp

```
AuthGroupFile   "/etc/group"
```

Maximo numero de clientes que pueden estar a la vez en el servidor, ademas adjuntamos un mensaje cuando sobrepasamos el numero maximo

```
MaxClients      X "mensaje"
```

Maximo numero de clientes con el mismo ip, ademas adjuntamos un mensaje cuando sobrepasamos el numero maximo

```
MaxClientsPerHost X "mensaje"
```

Maximo numero de clientes con la misma cuenta de usuario

```
MaxClientsPerUser X "mensaje"
```

Decimos si los usuarios necesitan tener una shell valida o no, aqui importante poner en off

```
RequireValidShell off/on
```

■ 2ªParte:

En la segunda parte nos limitamos a indicar mediante etiquetas los directorios de nuestro ftp. Normalmente la estrategia es indicar un directorio y darle unas propiedades. Los directorios inferiores jerarquicamente tomara los valores del padre, si no especificamos otra etiqueta dandole unas características distintas a este.

```
<Directory /directorioftp>
    Opciones del directorio
</Directory>
```

Las opciones del directorio debemos de tener en cuenta, que para dicho directorio estan por encima de las opciones indicadas en el principio del fichero (1ª parte), es decir, en la 1ª parte hemos indicado algunas opciones que son las generales para todo el ftp, pero si queremos añadir alguna opción especial, incluso una opción que contradiga a una expuesta en el principio, unicamente con añadirla dentro de la etiqueta Directory. ese directorio automaticamente

tomara esas propiedades y no las indicadas en la 1ª parte. Podemos añadir las siguientes opciones:

-Limitar el TIPO de acceso al directorio, siendo TIPO la opción READ(acceso lectura), WRITE(permite modificaciones sobre directorios existentes), STORE(permite escribir nuevos datos) y por último LOGIN(quien puede acceder). Después debemos de indicarles 'para quienes' tiene ese acceso de 2 formas muy fáciles, AllowAll(acceso todos) y DenyAll(deniegas a todos). Si deseamos una especificación mayor sería con Allow/Deny from ip1,ip2,ip3. Debemos de tener en cuenta que esta etiqueta también podemos usarla en la 1ª parte para impedir que ciertas ip puedan conectarse al ftp. Por último indicar que la etiqueta Allow/DenyAll viene realmente de la opción Allow/Deny from All así como podemos indicarle con la opción Order primero,segundo el orden a seguir a la hora de leer que está aceptado o rechazado, por ejemplo Order deny,allow primero cargara las direcciones Deny y luego las de Allow:

```
<Limit TIPO>
```

para quienes

```
</Limit>
```

Hay que tener en cuenta, que si no indicamos nada, el toma como Limit READ WRITE STORE con el valor DenyAll y la opción de LOGIN como AllowAll, encarga esta tarea al motor de comprobación de contraseña de nuestro usuario.

-La siguiente opción del directorio es la de indicarle si se permite sobrescribir ficheros existentes, debemos de tener acceso de WRITE y STORE para ello:

```
AllowOverwrite off/on
```

El valor predeterminado es off.

-Por último la opción que podemos añadir es indicarle nuestra máscara de permisos(funciona al contrario que la forma típica de dar permisos del chroot)

```
Umask XXX
```

■ 3ªParte:

El servidor anónimo es una etiqueta parecida a Directory pero con unas propiedades muy específicas. Estas son la de permitir el acceso al ftp sin password con el user predeterminado: anonymous. Dicho servicio lo creamos añadiendo la siguiente etiqueta:

```
<Anonymous directorio/anonimo>
  User ftp
  Group nogroup
  UserAlias anonymous ftp
  Opciones especificas
  <Directory directorio/anonimo>
    propiedades del directorio, ver parte 2ª
  </Directory>
</Anonymous>
```

Las opciones específicas están explicadas anteriormente, por ello listare las opciones que tiene este ftp tan especial:

```
AccessGrantMsg
AccessGrantMsg "mensaje"
RequireValidShell off/on
MaxClients X
MaxClientsPerHost X
MaxClientsPerUser X
```

Debemos de saber que la etiqueta anonymous no se considera un usuario, por lo que imaginarnos que deseamos en estos momentos, tener únicamente acceso anónimo y no permitir acceso privado, pues en vez de comentar la 2ª parte o borrarla podríamos poner la siguiente etiqueta limit para que solo pueda conectarse anónimamente (anonymous) indicando LOGIN a DenyALL:

```
<Limit LOGIN>
    DenyAll
</Limit>
```

Lo ultimo en este punto, es comentar que en los mensajes así como en la creación de logs, como veremos en el ejemplo siguiente podemos añadirle variables que tiene información que podemos usar a la hora de mostrar mensaje, dichas variables están asignadas con la terminología %letra, por ejemplo:

```
%m: numero de usuarios conectados al ftp
%u: nombre del usuario que se ha logueado
%t: fecha/hora a la que se logueo el usuario
```

Un ejemplo para un servidor proftpd que permite acceso anónimo (/ftp/anonimo) y acceso privado (/ftp/) con un directorio específico para subir información al servidor (/ftp/subir):

Archivo: /etc/proftpd/proftpd.conf

```
ServerName                "FTP Server"
ServerType                standalone
DeferWelcome              off
ServerIdent               on "Servidor ftp"

DefaultServer             on
ShowSymlinks              off
AllowOverwrite            off

AllowRetrieveRestart      yes
AllowStoreRestart         yes

DirFakeGroup              on
DirFakeUser               on ftp
DirFakeMode               0777

TimeoutNoTransfer         600
TimeoutStalled            100
TimeoutIdle               200

DenyFilter                \.*./

Port                      21

MaxInstances              2

AccessGrantMsg             "-- Bienvenido %u, tu acceso ha sido realizado correctamente =="
AccessDenyMsg              "Estas intentando acceder a un servidor privado, tus datos de conexion"

LogFormat                 default "%h %l %u %t \"%r\" %s %b"
LogFormat                 auth    "%v [%P] %h %t \"%r\" %s"
LogFormat                 write   "%h %l %u %t \"%r\" %s %b"
TransferLog               /var/log/proftpd/transfer
ExtendedLog               /var/log/proftpd/proftpd.down_up_log WRITE,READ write
ExtendedLog               /var/log/proftpd/proftpd.auth_log AUTH auth
ExtendedLog               /var/log/proftpd/proftpd.paranoid_log ALL default

DefaultRoot               ~
AuthUserFile               "/etc/passwd"
AuthGroupFile              "/etc/group"

DefaultRoot ~

MaxClients                 6 "Perdona, max %m usuarios -- Prueba después"
MaxClientsPerHost          2 "Solo 2 conexiones por HOST"
MaxClientsPerUser          2 "Solo 2 conexiones por usuario"

RequireValidShell          off

<Limit LOGIN>
    Order deny,allow
    Deny from 192.16.0.1, 192.16.0.5
```

```

    Allow    from all
</limit>
<Directory /ftp>
    Umask           022
    AllowOverwrite  off

    <Limit READ>
        AllowAll
    </Limit>
</Directory>

<Directory /ftp/subir>
    Umask           022
    AllowOverwrite  on

    <Limit READ WRITE>
        DenyAll
    </Limit>

    <Limit STOR>
        AllowAll
    </Limit>
</Directory>

<Anonymous /ftp/anonimo>
    AccessGrantMsg  "--= Bienvenido al FTP anonimo =-"
    User            ftp
    Group           nogroup

    UserAlias       anonymous ftp

    RequireValidShell  off

    MaxClients      5
    MaxClientsPerHost 1
    MaxClientsPerUser 5

    <Directory /ftp/anonimo>
        Umask           077 077

        AllowOverwrite  off
    </Directory>
</Anonymous>

```

Los directorios deben de estar creados con los pertinentes permisos, mirando el ejemplo anterior, los directorios creados y los permisos que deberíamos haberle otorgado serian los siguientes:

```

# mkdir /ftp
# mkdir /ftp/anonimo
# mkdir /ftp/subir
# chmod 755 /ftp
# chmod 755 /ftp/anonimo
# chmod 777 /ftp/anonimo/subir

```

Runlevel

Para añadir el demonio proftpd en el runlevel y por tanto para que arranque cuando iniciemos nuestra maquina, unicamente deberemos de introducir el siguiente comando:

```

# rc-update add proftpd default

```

Iptables

Si disponemos de un fichero de iptables deberíamos de añadir las siguientes reglas a nuestro script:

Archivos: mi script iptables.sh

Archivo: mi.script iptables.sn

```

...
$Iptables -N FLOODFTP
$Iptables -A FLOODFTP -m limit --limit 4/s --limit-burst 5 -j RETURN
$Iptables -A FLOODFTP -j LOG --log-level info --log-prefix 'DoS FTP: '
$Iptables -A FLOODFTP -j DROP
$Iptables -A INPUT -p tcp --dport 21 -f -j DROP
$Iptables -A INPUT -p tcp --dport 21 -m state --state NEW -j FLOODFTP
$Iptables -A INPUT -p tcp --dport 21 -m state --state NEW -j LOG --log-level info --log-prefix 'FTP: '
$Iptables -A INPUT -p tcp --dport 21 -m state --state NEW -j ACCEPT
...

```

Uso del servidor Proftpd

Arranque/Parada/Rearranque

Para arrancar nuestro ftp deberemos de hacer lo siguiente:

```
# /etc/init.d/proftpd start
```

Para reiniciarlo:

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

Para pararlo:

```
# /etc/init.d/proftpd stop
```

Creacion/Eliminacion de cuentas

La creacion de usuarios la haremos como si fuera una creacion de un usuario cualquiera, indicandole que nuestro directorio es el del ftp y nuestro shell es una shell falsa, como por ejemplo /bin/false:

```
# adduser -d /directorioftp -s /bin/false nombre_usuario && passwd nombre_usuario
```

Para eliminar un usuario seria la siguiente orden:

```
# userdel nombre_usuario
```

Control sobre los usuarios

Para saber en que momento cuantos, quienes y que operaciones estan realizando los usuarios en su servidor ftp, existen 2 pequeñas utilidades que permiten mostrarselo:

```
# ftpwho
# ftptop
```

Para expulsar a algun usuario del servidor ftp, unicamente deberemos de matar el proceso que identifica al usuario. Para ver el PID o numero identificativo del proceso correspondiente a ese usuario, podemos usar el 'ftpwho'(es el 1 valor, es un numero normalmente de 4 a 5 cifras) o usar el comando 'ps aux'en el cual el PID es el 2 valor de la tabla. Para matar el proceso deberemos de escribir lo siguiente:

```
# kill -9 PID
```

// -9 == SIGKILL, es recomendable mandar SIGTERM == -15, para que así sea el mismo proceso el que termine y así no tener que matarlo, siempre que se pueda

Cierre por mantenimiento

Si deseamos cerrar el ftp momentaneamente y queremos a su vez que nuestros usuarios sepan que el servidor esta cerrado momentaneamente existe el comando 'ftpsht tiempo "mensaje"', por ejemplo para cerrar el ftp y dejar un mensaje identificativo en el caso de mantenimiento del servidor seria de la siguiente forma:

```
# ftpshut now "El servidor se encuentra cerrado por motivos tecnicos"
```

Documentacion extra

Pagina principal de proftpd

Para levantar el servidor despues de haber realizado el mantenimiento. Teclea: ftpshut -R

Créditos

- Original creado por g0su
- Portado al wiki por g0su
- Revisión léxico-gramatical por D. González-Arjona (no se han usado las tildes para evitar conflictos con codificadores)