

**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

Soto Ramos Manuel Alejandro

**Administración de Servicios en Red**

**Manual de Instalación y Configuración - Servidor FTP/FTPS**

Jiménez Maruri Pedro  
Romero Serrano Luis

Valencia Rodriguez Fernando Quetzalcoatl

Grupo: 4CV4

# 

# **Índice**

[**Índice**](#_hioritr8619r) **1**

[**Requerimientos**](#_dspt8prqwzu4) **3**

[**Sistema**](#_6pmexxzco2qk) **3**

[**Licencia**](#_xg0d2pvpsll8) **3**

[**Instalación**](#_p97jo3mur612) **3**

[Configurar iptables](#_4x69w8i5r635) 4

[**Configuración**](#_qnn9wnxtxee5) **4**

[Nombre del Servidor](#_7w2ods3u15dq) 5

[Configuración de puerto de operación](#_vvx3efgxi2vz) 5

[Rango de Puertos Pasivos](#_fl8fbbbx13p9) 5

[Definir archivos de autentificación](#_ttcfil65f35x) 5

[Crear Usuario](#_yh2logcmv6pz) 6

[Crear Usuario en Sistema](#_o1bskubwnvne) 6

[Crear Usuario Virtual](#_pa0iukyzsc8r) 6

[Permitir Usuarios Anónimos](#_ftfu97vdbqbl) 7

[Jaulas](#_143uibfdxzow) 8

[Bloquear por IP](#_n53m8zpb8xmt) 10

[<Límite de LOGIN>, y Listas Blancas versus Listas Negras](#_wrfcl7w2ydjh) 10

[Grandes listas negras](#_sx3shdgoast2) 11

[Uso de las clases](#_modi64rfs3be) 12

[Bloquear por Usuario](#_jt9uzatg2zhy) 13

[Bloquear usuario por archivo ftpusers](#_ynudk31xssd1) 14

[**Certificados**](#_kvod04wmuxyb) **14**

[Instalar OpenSSL](#_wg7t8q2pdh2f) 14

[Crear el certificado SSL para TLS](#_84y9air6ez47) 14

[Habilitar TLS en ProFTPd](#_t790k4v5ymre) 15

[**Bitacoras**](#_27grx137zaw5) **16**

[Archivos de registro](#_lvjlsdkv5wnn) 16

[Registro de transferencias](#_wpn68o6ywmfv) 17

[Registro del sistema](#_9c0gs1p7c6mo) 17

[ExtendedLog](#_s78jb6qo8669) 17

[Análisis de registros](#_rv9ii4mrsinm) 17

[Rotación de registros](#_d4ocfok0usrf) 17

[Registro SQL](#_crfzrl7r026f) 17

[Registro Trace](#_6sbakaqtk4p7) 18

[Directivas de configuración](#_7ewyvb5a72jl) 18

[**Referencias**](#_i0zlsi60th0i) **19**

# 

# **Requerimientos**

Los requerimientos mínimos para que funcione ProFTPd son:

**Hardware**:

* Procesador Xeon a 3 Ghz,
* Memoria RAM 4GB,
* Espacio en disco 20GB

**Software**:

sistema operativo Unix u otro de los especificados en la parte de abajo, en caso de contar con Window en algún emulador que corra los sistemas operativo que a continuación se mencionan.

# **Sistema**

ProFTPd funciona sobre distintos sistemas operativos y emuladores, a continuación se enlistan algunos ejemplos.

* AIX
* BSD/SO
* Linux
* Mac OS X
* Gento
* Solaris

# **Licencia**

Este servidor cuenta con una licencia “*General Public License (GPL)*”, que es una licencia de derechos de autor usada en software libre y código abierto que garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.

# **Instalación**

ProFTPd está disponible en los repositorios de software por defecto de Ubuntu, por lo que lo instalaremos utilizando las herramientas convencionales de gestión de paquetes.

Comenzaremos por actualizar el índice de paquetes locales para que refleje los últimos cambios. Después, podemos instalar el paquete proftpd:

sudo apt-get update  
 sudo apt-get install proftpd

Después de confirmar la instalación, apt-get instalará proFTPd y todas las dependencias necesarias.

## **Configurar iptables**

Necesitamos abrir los puertos 20, 21 y el rango de puertos que está especificado en el fichero de configuración para las conexiones pasivas. Si el puerto de operación es diferente se debe poner en lugar del 21, ejemplo:

iptables -A INPUT -p tcp --dport 21 -j ACCEPT

iptables -A INPUT -p tcp --dport 20 -j ACCEPT

iptables -A INPUT -p tcp --dport 30000:50000 -j ACCEPT

Si el puerto de operación es diferente se debe poner en lugar del 21, por ejemplo si se quiere poner el puerto 131:

iptables -A INPUT -p tcp --dport 131 -j ACCEPT

iptables -A INPUT -p tcp --dport 20 -j ACCEPT

iptables -A INPUT -p tcp --dport 30000:50000 -j ACCEPT

Y ejecute un comando guardar para confirmar los cambios:

iptables-save

Podemos guardar la configuración actual de iptables en un archivo y recargarla si hacemos cambios y queremos restaurar estos ajustes más tarde:

iptables-save &gt; /path/to/file.txt

Estamos listos con la instalación del servidor FTP y deberíamos ser capaces de conectarnos con éxito.

# **Configuración**

El archivo de configuración principal de proFTPd se encuentra en **/etc/proftpd/proftpd.conf**. Desde aquí deberemos configurar la mayoría de los aspectos de nuestro servidor FTP.

Antes de hacer cualquier cambio en la configuración del servidor es bueno generar una copia de seguridad.

sudo cp /etc/proftpd/proftpd.conf /etc/proftpd/proftpd.conf.back

Lo primero que debemos hacer es establecer una carpeta como directorio principal para todos los usuarios del FTP. Para ello, abriremos con nuestro editor de texto preferido el archivo de configuración anterior:

sudo nano /etc/proftpd/proftpd.conf

**Nota:** Cada vez que cambie la configuración es necesario reiniciar el servidor. Se puede reiniciar introduciendo el siguiente comando en una terminal:

**sudo systemctl restart proftpd**

## **Nombre del Servidor**

Establecer un nombre de servidor (normalmente, el nombre de host del servidor se añade aquí):

**/etc/proftpd/proftpd.conf**

[…]

ServerName “ejemplo.com”

[...]

## **Configuración de puerto de operación**

Aquí se define el puerto principal de operación por el cual el servidor estará a la escucha de nuevas conecciones.

**/etc/proftpd/proftpd.conf**

[…]

# Port 21 is the standard FTP port.

Port 21

[...]

## **Rango de Puertos Pasivos**

Defina el rango de puertos para conexiones en modo pasivo (esta opción sólo existe en el fichero de configuración de los paquetes Debian/Ubuntu).

**/etc/proftpd/proftpd.conf**

[…]

PassivePorts 30000 50000

[...]

## **Definir archivos de autentificación**

Definir los archivos que utilizaremos para la autenticación. Los archivos AuthUserFile y AuthGroupFile se crearán más tarde, pero ahora podemos definirlos para completar la configuración.

**/etc/proftpd/proftpd.conf**

[…]

## a continuacion se activa las opciones para usuarios virtuales

AuthOrder mod\_auth\_file.c mod\_auth\_unix.c

AuthUserFile /etc/proftpd/ftpd.passwd

AuthGroupFile /etc/proftpd/ftpd.group

[...]

## **Crear Usuario**

### **Crear Usuario en Sistema**

ProFTPd detecta los usuarios en el sistema, por lo que se puede crear un usuario en sistema y proftpd lo agregara automáticamente. El proceso para crear usuarios es:

sudo adduser nombre\_de\_usuario

A continuación estableceremos la contraseña para el mismo:

sudo passwd nombre\_de\_usuario

A partir de ahora, “nombre\_de\_usuario” podrá conectarse al FTP y accederá, por defecto, a la carpeta especificada en DefaultRoot.

### **Crear Usuario Virtual**

Aunque agregar un usuario al sistema es la forma más fácil de agregar usuarios, esto contamina el sistema con muchos usuarios. Existe otra forma de agregar usuarios a proftpd sin que se tengan que agregar al sistema. Este método se llama creación de usuarios virtuales.

Primero, necesitamos crear los archivos de configuración que contendrán a nuestros usuarios y grupos. Podemos crear los archivos en un directorio aleatorio. En este ejemplo usaremos **/etc/proftpd/**. Los archivos de configuración se crean automáticamente cuando se intenta añadir un usuario a un archivo no existente.

sudo mkdir /etc/proftpd

cd /etc/proftpd

Para añadir un usuario en ftpd.conf utilice el siguiente comando:

ftpasswd --passwd --name=user\_name --home=/default/user/folder --shell=/bin/bash --uid=500 --gid=500

**Nota:** Tenga en cuenta que la carpeta que ha configurado como predeterminada para el usuario (**--home=/default/user/folder**) debe existir!

Una vez ejecutado, se le pedirá que introduzca la contraseña de este usuario dos veces.

Las opciones exactas que contiene el comando son:

* **--name** - el nombre de usuario de la cuenta ftp
* **--home** --home - su directorio por defecto
* **--shell** - la carpeta shell
* **--uid** - ID de usuario
* **--gid** - ID de grupo

**Nota:** Si usted está en el servidor ubuntu, entonces el valor de uid y gid debe ser 33 (id del servidor web), de lo contrario puede que tenga problemas al escribir los archivos.

Al ejecutar este comando, se generará automáticamente un archivo ftpd.passwd, en el que se agregará el usuario que hemos creado.

Después de eso necesitamos crear el grupo:

ftpasswd --group --gid=500 --name=group\_name

Ahora necesitamos agregar restricciones de acceso a los archivos de autentificación.

chmod 400 /etc/proftpd/ftpd.passwd /etc/proftpd/ftpd.group

chown proftpd.nogroup /etc/proftpd/ftpd.passwd /etc/proftpd/ftpd.group

### **Permitir Usuarios Anónimos**

Para permitir usuarios anónimos, sólo hay que descomentar la parte del archivo de configuración que dice:

**/etc/proftpd/proftpd.conf**

[…]

# A basic anonymous configuration, no upload directories.

# <Anonymous ~ftp> #Desde aqui

# [...]

# </Anonymous> #Hasta aca quitar el primer ‘ # ’

[...]

Con esto se habilita que los usuarios puedan ingresar con el usuario **anonymous** o **ftp**, sin que requiera contraseña.

Para enjaular a los usuarios anónimos a una carpeta tenemos que poner la ruta de la carpeta en la etiqueta Anonymous.

**/etc/proftpd/proftpd.conf**

[…]

<Anonymous L>

[...]

Los usuarios anónimos no podrán subir mas de esta carpeta.

Cuando a opción **RequireValidShell** está apagada, permite ingresar de forma automática como usuario anónimo a través de un navegador web. Esto lo hace no mostrando la caja de autentificación.

**/etc/proftpd/proftpd.conf**

<Anonymous>

[…]

RequireValidShell off

[...]

<Anonymous ~ftp>

Si no se requiere registrar el acceso de los usuarios anónimos, se puede agregar la siguiente línea a dentro de la configuración de los usuarios anónimos:

**/etc/proftpd/proftpd.conf**

<Anonymous>

[…]

WtmpLog off

[...]

<Anonymous ~ftp>

Para registrar las transferencias de usuario anónimos ponemos lo siguiente:

**/etc/proftpd/proftpd.conf**

<Anonymous>

[…]

# # Logging for the anonymous transfers

ExtendedLog /var/log/proftpd/access.log WRITE,READ default

ExtendedLog /var/log/proftpd/auth.log AUTH auth

[...]

<Anonymous ~ftp>

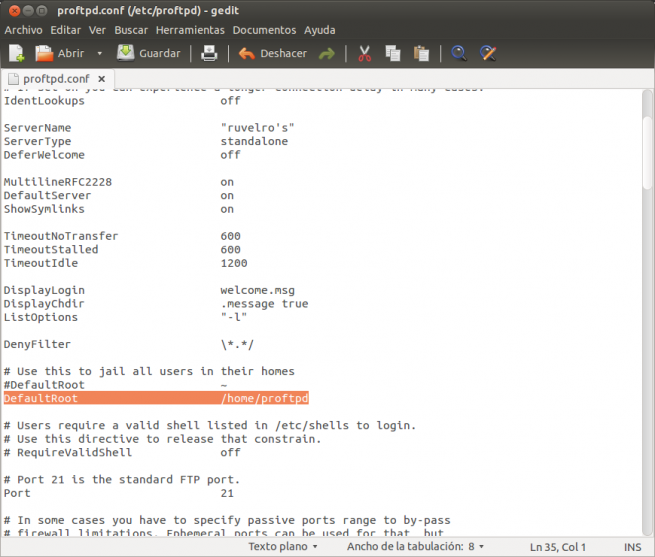
## **Jaulas**

Ahora buscaremos la línea comentada “DefaultRoot” y la descomentamos borrando la almohadilla #. Esto nos va a permitir que cuando cada usuario acceda a su cuenta del FTP, estos accederán directamente a su carpeta “home”.

Para esto crearemos la carpeta pública

sudo mkdir /home/proftpd

Si queremos que todos los usuarios que inicien sesión accedan por defecto a una misma carpeta, debemos cambiar el parámetro DefaultRoot y añadir la ruta a la que queramos que accedan. Por ejemplo, en nuestro caso hemos creado una carpeta llamada “proftpd” dentro de /home/ y queremos que todos los usuarios por defecto accedan a ella. El archivo de configuración quedará de la siguiente manera:



También podemos hacer que todos los usuarios accedan por defecto a una carpeta (como en el paso anterior) salvo un usuario que queremos que acceda a otra o, simplemente, tenga permiso de administrador y pueda acceder a todo el disco.

Para ello debemos modificar los siguientes aspectos:

El parámetro DefaultRoot tiene la siguiente estructura:

DefaultRoot [directorio] [grupo de usuarios que tendrán ese directorio por defecto] [Grupo de usuarios a los que no se les aplicará ese directorio]

Es decir, vamos a poner el ejemplo de tener 2 usuarios, A y B. A únicamente podrá acceder a su directorio mientras que B podrá acceder a todo el disco duro. Debemos añadir las siguientes líneas:

**/etc/proftpd/proftpd.conf**

[…]

DefaultRoot /home/ftp A

DefaultRoot / B

[...]

## **Bloquear por IP**

Muchos sitios que ejecutan proftpd tienen la necesidad de limitar/rechazar conexiones de direcciones erróneas conocidas, o de aceptar conexiones sólo de direcciones buenas conocidas. ProFTPD soporta varios mecanismos diferentes para implementar ACLs de conexión FTP.

Sin embargo, hay una cosa importante a tener en cuenta: ProFTPD no es un cortafuegos. Nunca olvide esto. No se puede evitar que las conexiones TCP lleguen a proftpd en primer lugar utilizando proftpd. Esto significa que para muchos sitios con necesidades de conexión ACL, las capacidades de firewall de su sistema pueden ser un mejor mecanismo para implementar ACLs.

### **<Límite de LOGIN>, y Listas Blancas versus Listas Negras**

La forma más fácil de aceptar/rechazar conexiones es usar la sintaxis <Limit> de ProFTPD, particularmente <Limit LOGIN>.

Por ejemplo, puede simplemente rechazar todas las conexiones a su instancia de proftpd usando:

**/etc/proftpd/proftpd.conf**

[…]

<Limit LOGIN>

DenyAll

</Limit>

[...]

Un ejemplo más práctico sería:

**/etc/proftpd/proftpd.conf**

[…]

<Limit LOGIN>

# Estas son direcciones confiables

Allow from 1.2.3.4 5.6.7.8

Allow from trusted-domain.com

# Todos los demás son bloqueados

DenyAll

</Limit>

La configuración anterior demuestra el concepto de lista blanca: las direcciones de confianza conocidas (es decir, "blancas") se enumeran y permiten explícitamente, y todas las demás se rechazan. Por el contrario, puede utilizar una lista negra, en la que las direcciones conocidas malas/maliciosas/no deseadas (es decir, "negras") se enumeran y se rechazan explícitamente, y todas las demás están permitidas:

**/etc/proftpd/proftpd.conf**

[…]

<Limit LOGIN>

Order deny,allow

# These are known bad addresses

Deny from 1.2.3.4 5.6.7.8

Deny from evil-domain.com

# Everyone else is allowed

AllowAll

</Limit>

[...]

Es útil saber acerca de las listas blancas en comparación con las listas negras, ya que puede afectar el tamaño de sus ACL. Los sitios que ejecutan servidores FTP disponibles públicamente pero que necesitan eliminar a los clientes malos tienden a utilizar listas negras; con el tiempo, estas listas negras pueden llegar a ser bastante grandes. Los sitios que saben quién usará su servidor FTP y sólo quieren que esas personas puedan conectarse usarán listas blancas; estas listas blancas tienden a ser bastante pequeñas o cortas.

#### **Grandes listas negras**

Supongamos que está ejecutando un servidor FTP público y que está utilizando una sección <Limit LOGIN> para crear una lista negra. Su sitio puede incluso tener algún script que detecte clientes malos y los añada automáticamente a su lista negra, por ejemplo:

**/etc/proftpd/proftpd.conf**

[…]

<Limit LOGIN>

Order deny,allow

# -- Add blacklisted addresses here --

Deny from ...

Deny from ...

# Many more Deny from lines

# Allow everyone not explicitly blacklisted

AllowAll

</Limit>

Con el tiempo, el número de direcciones IP en la lista negra comienza a ser grande y difícil de manejar. Usted podría tener miles de direcciones de este tipo; se da cuenta de que proftpd comienza a actuar de forma extraña y comienza a disminuir la velocidad.

Cuando esto sucede, usted debe comenzar a buscar maneras de hacer su lista más corta. Si su lista está compuesta de entradas para cada dirección IP individual, debería pensar en usar expresiones glob y máscaras de red para reducir el número de entradas. Por ejemplo, en lugar de usar algo como esto:

Deny from 1.2.3.4

Deny from 1.2.3.5

Deny from 1.2.3.6

Deny from 1.2.3.7

Deny from 1.2.3.8

Deny from 1.2.3.9

Podría (a partir de proftpd-1.3.4rc1) usar un rango:

Deny 1.2.3.[4-9]

O incluso utilizar un comodín **glob** como este, lo que reduce aún más el número de entradas en sus listas de direcciones:

Deny 1.2.3.\*

O puedes usar máscaras de red como:

Deny 1.2.3.0/8

#### **Uso de las clases**

Ahora supongamos que en este punto usted ha usado rangos globales, comodines y máscaras de red para hacer su ACL más corto, pero todavía es grande, y proftpd es todavía lento. Es hora de usar las clases de conexión de ProFTPD.

Para hacer esto, usted reescribiría su gran sección <Limit LOGIN> para que se vea algo como esto:

**/etc/proftpd/proftpd.conf**

[…]

<Class blacklist>

# Put all of your blacklisted address/range lines here, as From lines

From ...

From ...

From ...

</Class>

<Limit LOGIN>

Order deny,allow

# Rejected blacklisted clients

DenyClass blacklist

# Allow everyone else

AllowAll

</Limit>

¿Por qué el uso de clases como ésta hace que las cosas sean más rápidas? Cuando se utilizan clases, la etiqueta (es decir, el nombre de la clase en la sección <Class>, como "blacklist" en el ejemplo anterior) se determina cuando el cliente se conecta por primera vez a proftpd, antes de que se envíe cualquier comando. Es entonces cuando se comprueban todas las máscaras de red y similares, y sólo en ese momento. Después de eso, todas las ACL basadas en la configuración se evalúan sólo usando esa etiqueta.

Cuando no utiliza clases, proftpd analiza cada configuración, cada máscara de red/dirección para cada comando. Por eso las cosas se ralentizan tanto.

## **Bloquear por Usuario**

Podemos controlar los usuarios que pueden y no pueden iniciar sesión en nuestro servidor de una forma muy sencilla. Para ello, abrimos el archivo de configuración y añadiremos lo siguiente al final de este:

**/etc/proftpd/proftpd.conf**

[…]

<Limit LOGIN>

AllowUser nombre\_de\_usuario

DenyAll

</Limit>

[…]

Otros parámetros que podemos utilizar aquí son:

* AllowUser: Nos permite permitir un usuario específico.
* DenyUser: Bloqueamos el acceso a un usuario específico.
* DenyAll: Bloqueamos el acceso a todos los usuarios salvo a los que especifiquemos con AllowUser.
* AllowAll: Permite que los usuarios como “anonymous” se conecten al FTP.

Por ejemplo, si queremos permitir las conexiones anónimas a nuestro FTP debemos añadir dentro del grupo anterior el parámetro AllowAll, quedando de la siguiente manera:

**/etc/proftpd/proftpd.conf**

[…]

<Limit LOGIN>

AllowUser nombre\_de\_usuario

AllowAll

</Limit>

[…]

### **Bloquear usuario por archivo ftpusers**

Para denegar a los usuarios el acceso al servidor FTP, añada sus nombres en el archivo /etc/ftpusers y guárdelo.

Todos los usuarios de ese archivo no tendrán acceso al servidor FTP.

# **Certificados**

FTP sin TLS es un protocolo inseguro porque todas las contraseñas y todos los datos se transfieren en texto claro. Al utilizar TLS, toda la comunicación puede ser encriptada, lo que hace que el FTP sea mucho más seguro.

## **Instalar OpenSSL**

TLS necesita OpenSSL; para instalar OpenSSL, simplemente ejecutamos:

apt-get -y install openssl

Por razones de seguridad, debería añadir la siguiente línea a **/etc/proftpd/proftpd.conf**:

sudo nano /etc/proftpd/proftpd.conf

**/etc/proftpd/proftpd.conf**

[…]

ServerIdent on "FTP Server ready."

[...]

Esto habilita un mensaje ServerIdent que no contiene ninguna información sobre el software, la versión o el sistema operativo del servidor FTP utilizado, de modo que un posible atacante no obtenga estos detalles en bandeja de plata.

## **Crear el certificado SSL para TLS**

Para poder utilizar TLS, debemos crear un certificado SSL. Lo creo en **/etc/proftpd/ssl**, por lo tanto, primero creo ese directorio:

sudo mkdir /etc/proftpd/ssl

Después, podemos generar el certificado SSL de la siguiente manera:

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/proftpd/ssl/proftpd.key -out /etc/proftpd/ssl/proftpd.crt

Se le harán una serie de preguntas. Antes de repasar eso, echemos un vistazo a lo que está sucediendo en el comando que estamos emitiendo:

* **openssl:** es la herramienta básica de la línea de comandos para crear y gestionar certificados, claves y otros archivos de OpenSSL.
* **req:** Este subcomando específica que queremos utilizar la gestión de solicitud de firma de certificado (CSR) X.509. El "X.509" es un estándar de infraestructura de clave pública al que se adhieren SSL y TLS para la gestión de claves y certificados. Queremos crear un nuevo cert X.509, así que estamos usando este subcomando.
* **-x509:** Esto modifica aún más el subcomando anterior al indicar a la utilidad que queremos hacer un certificado autofirmado en lugar de generar una solicitud de firma de certificado, como ocurriría normalmente.
* **-nodes:** Esto le indica a OpenSSL que omita la opción de asegurar nuestro certificado con una frase de contraseña. Necesitamos que Apache sea capaz de leer el archivo, sin intervención del usuario, cuando el servidor se inicie. Una frase de contraseña evitaría que esto ocurriera porque tendríamos que introducirla después de cada reinicio.
* **-days 365:** Esta opción establece el período de tiempo durante el cual el certificado se considerará válido. Lo fijamos para un año aquí.
* **-newkey rsa:2048:** Esto especifica que queremos generar un nuevo certificado y una nueva clave al mismo tiempo. No creamos la clave necesaria para firmar el certificado en un paso anterior, por lo que debemos crearla junto con el certificado. La porción rsa:2048 le dice que haga una clave RSA de 2048 bits de longitud.
* **-keyout:** Esta línea le indica a OpenSSL dónde colocar el archivo de clave privada generado que estamos creando.
* **-out:** Indica a OpenSSL dónde colocar el certificado que estamos creando.

Después de crear la llave y el certificado, asegure los archivos de certificado generados.

sudo chmod 600 /etc/proftpd/ssl/proftpd.\*

## **Habilitar TLS en ProFTPd**

Para habilitar TLS en ProFTPd, abra **/etc/proftpd/proftpd.conf**.

sudo nano /etc/proftpd/proftpd.conf

y descomentar la línea *Include /etc/proftpd/tls.conf*:

**/etc/proftpd/proftpd.conf**

[...]

#

# This is used for FTPS connections

#

Include /etc/proftpd/tls.conf

[...]

Luego abra **/etc/proftpd/tls.conf** y haga que se vea como sigue:

sudo nano /etc/proftpd/tls.conf

**/etc/proftpd/tls.conf**

[...]

<IfModule mod\_tls.c>

TLSEngine on

TLSLog /var/log/proftpd/tls.log

TLSProtocol TLSv1.2

TLSCipherSuite AES128+EECDH:AES128+EDH

TLSOptions NoCertRequest AllowClientRenegotiations

TLSRSACertificateFile /etc/proftpd/ssl/proftpd.crt

TLSRSACertificateKeyFile /etc/proftpd/ssl/proftpd.key

TLSVerifyClient off

TLSRequired on

RequireValidShell no

</IfModule>

[...]

Si utiliza TLSRequired on, sólo se permiten las conexiones TLS (esto bloquea a los usuarios con clientes FTP antiguos que no tengan soporte para TLS); al comentar esa línea o utilizar TLSRequired off, se permiten tanto las conexiones TLS como las no TLS, dependiendo de lo que soporte el cliente FTP.

A continuación, reinicie ProFTPd:

sudo systemctl restart proftpd.service

Eso es todo. Ahora puede intentar conectarse utilizando su cliente FTP; sin embargo, debe configurar su cliente FTP para que utilice TLS (esto es imprescindible si utiliza TLSRequired) - vea el siguiente capítulo sobre cómo hacerlo con FileZilla.

Si tiene problemas con TLS, puede consultar el archivo de registro de TLS **/var/log/proftpd/tls.log**.

# **Bitacoras**

ProFTPd registra los accesos y transferencias en los archivos que se encuentra en la carpeta **/var/log/proftpd/**

## **Archivos de registro**

Hay tres tipos principales de registros que un demonio proftpd puede generar: TransferLog, SystemLog y ExtendedLog.

### **Registro de transferencias**

Un TransferLog es el registro más común que se mantiene, registrando las transferencias de archivos. Está escrito en el formato xferlog(5).

### **Registro del sistema**

Si el administrador del sitio quiere que proftpd registre sus mensajes en un archivo en lugar de pasar por syslogd, la directiva de configuración de SystemLog es la que debe usarse. Sólo se mantiene un archivo de este tipo para todo el demonio. Consulte la directiva ServerLog para mantener un registro similar por host. Tenga en cuenta que la directiva DebugLevel sólo se aplica a los archivos SystemLog; no afecta materialmente a los mensajes de registro basados en syslog.

### **ExtendedLog**

La directiva ExtendedLog se utiliza para crear archivos de registro de un formato muy flexible y configurable, y para tener un control granular sobre lo que se registra y cuándo. El formato de un ExtendedLog se describe usando la directiva LogFormat. Se pueden configurar varios ExtendedLogs, cada uno con un LogFormat diferente.

## **Análisis de registros**

Hay una variedad de analizadores de registros disponibles; estos son sólo algunos:

* Webalizer: http://www.webalizer.org/
* Análisis HTTP: http://www.netstore.de/Supply/http-analyze/
* Informe Magic: http://www.reportmagic.org/
* FTPWeblog: http://www.nihongo.org/snowhare/utilities/ftpweblog/

## **Rotación de registros**

Incluso en un servidor moderadamente ocupado, la cantidad de información almacenada en los archivos de registro es muy grande. En consecuencia, será necesario rotar periódicamente los archivos de registro moviendo o eliminando los registros existentes. Esto no se puede hacer mientras el servidor está ejecutándose, porque el demonio continuará escribiendo en el antiguo archivo de registro mientras mantenga abierto el archivo. En su lugar, el servidor debe reiniciarse después de mover o eliminar los archivos de registro para que pueda abrir nuevos archivos de registro.

## **Registro SQL**

El módulo mod\_sql también permite algunas capacidades de registro potentes y complejas.

## **Registro Trace**

“Tracing" es una nueva forma de registro, introducida en la serie 1.3.1 ProFTPD.

El registro Trace tiene el concepto de múltiples canales de registro, cada uno de los cuales tiene su propio nivel de registro. Las capas y las APIs dentro de la fuente ProFTPD tenderán a utilizar sus propios canales. No hay límite en el número de canales de registro diferentes que se pueden admitir.

Dentro de un canal, la suposición es que los niveles de registro más bajos indican mayor prioridad o importancia. O para verlo de otra manera, cuanto más alto es el nivel de registro para el canal, más ruidoso puede ser ese canal de registro. El nivel de registro más bajo es 1; no hay límite de nivel de registro superior.

¿Por qué tener esta nueva forma de registro, además de todos los otros tipos de registros (por ejemplo, SystemLog, ExtendedLog, etc.) que un demonio proftpd puede producir actualmente? El habitual SystemLog de registro de depuración de proftpd, a un alto nivel de DebugLevel, se estaba volviendo ilegible; es difícil encontrar los trozos de conocimiento entre los demás mensajes de ese archivo. Al separar los mensajes de registro en canales y niveles, el rastreo le da al administrador un control mucho más fino sobre el registro, dándoles una forma de enfocar el registro más estrechamente, de modo que sólo se registra el área de código de interés.

La compatibilidad con el rastreo está activada de forma predeterminada. Utilice la opción --disable-trace configure, al compilar ProFTPD, para desactivar todo el soporte de rastreo. Recomiendo que los sitios de producción de alto tráfico, que no necesitan un registro de depuración en esta granularidad, usen la opción --disable-trace.

### **Directivas de configuración**

Hay dos nuevas directivas de configuración para el trazado: TraceLog y Trace. Tenga en cuenta que para que el trazado sea efectivo, estas dos directivas, si se utilizan, deben aparecer al principio del archivo proftpd.conf, antes que cualquier otra directiva.

La directiva TraceLog especifica un nombre de archivo para escribir los mensajes de registro de rastreo. Por ejemplo:

**/etc/proftpd/proftpd.conf**

[...]

TraceLog /var/ftpd/trace.log

[...]

Sin esta directiva, no se producirá ningún registro de trazas.

Una vez que haya configurado su TraceLog, usará la directiva Trace para controlar la verbosidad de ese registro:

**/etc/proftpd/proftpd.conf**

[...]

Trace channel1:level1 channel2:level2 ...

[...]

Esta directiva le permite establecer cada canal de registro y su nivel de forma diferente, por ejemplo:

**/etc/proftpd/proftpd.conf**

[...]

Trace command:5 response:8 timer:2 config:9

[...]

También hay soporte para una palabra clave especial "DEFAULT":

**/etc/proftpd/proftpd.conf**

[...]

Trace DEFAULT:10

[...]

# **Referencias**

http://www.uk.proftpd.org/docs/howto/Logging.html