INSTITUTO LINUX

LINUX COLLEGE

03.05

REDES LINUX

CONFIGURACIÓN DE FTP

EXPERTO EN ADMINISTRACIÓN DE REDES LINUX

-. MÓDULO 03 · CAPÍTULO 05.1 .-

LINUX FOUNDATION AUTHORIZED TRAINING PARTNER

Linux Professional Institute

UTN FACULTAD REGIONAL DELTA

Universidad Nacional de San Luis

debian

Capítulo 05.1 - **CONFIGURACIÓN DE FTP**

ÍNDICE

Capítulo 05.1 - **CONFIGURACIÓN DE FTP**

Servicio FTP (File Transfer Protocol)

El **protocolo FTP (File Transfer Protocol)** o **Protocolo de Transferencia de Archivos** es un protocolo de red utilizado para la transferencia de archivos entre las computadoras conectadas a una red TCP y se encuentra dentro de la capa de aplicación.

Como vamos a ver, este protocolo utiliza el puerto 21 y dependiendo de cierta configuración puede utilizar el puerto 20 u otros puertos.

Los orígenes del FTP se remontan a 1971, cuando los ingenieros del MIT y otras instituciones académicas buscaban un método eficaz para la transferencia de archivos.

Básicamente, se encuentra diseñado en torno a una **arquitectura del tipo cliente-servidor**; es decir, que el equipo o computadora cliente se debe conectar primero a un servidor para descargar (download) o subir archivos (upload).

Antes de continuar, debemos saber que el FTP es considerado como uno de los protocolos inseguros de por si.

Que quiere decir esto, que a diferencia de SSH, **la transferencia de usuario y contraseña se transmite en texto plano**. O sea, que los paquetes que viajan por la red **se transmiten sin ningún tipo de cifrado** y es susceptible de ser visto por cualquier otro equipo por el que navegue ese tráfico. Esto es algo que debemos tener en cuenta y que muchas veces se encapsula este protocolo en otro más seguro, como por ejemplo SSH.

Introducción a VSFTPD

Very Secure FTP Daemon (**VSFTPD**) es un software utilizado para implementar servidores de archivos a través del protocolo FTP. .

Se distingue principalmente porque sus valores por defecto son muy seguros y por su sencillez en la configuración, comparado con otras alternativas como Wu-ftpd.

Actualmente, se presume que VSFTPD es quizá el servidor FTP más seguro del mundo.

Instalación del software

Instalamos los paquetes del Servidor de VFTP escribiendo en la consola el siguiente comando:

```
# apt-get install vsftpd
```

Archivo de configuración

Los archivos de configuración con los que trabajaremos son dos:

- /etc/vsftpd.user_list, la lista que definirá usuarios a enjaular o no a enjaular, dependiendo de la configuración.
- /etc/vsftpd/vsftpd.conf, el archivo de configuración principal.

Parámetros importantes del archivo de configuración

· **listen:** este parámetro hace que corran como standalone. Esto hace que corra como un proceso independiente.

· **anonymous_enable:** este parámetro se utiliza para definir si se permitirán los accesos anónimos al servidor. Estableceremos los valores YES o NO de acuerdo a lo que queramos permitir.

· **local_enable:** Estev parámetro es particularmente atractivo si se combina con la función de jaula ya que establece si se van a permitir los accesos autenticados de los usuarios locales del sistema. Definiremos este valor por YES o NO de acuerdo a lo que queramos habilitar según el servidor que estemos configurando.

· **write_enable:** este parámetro establece si se permite el mandato “write” (escritura) en el servidor. El valor establecido por defecto es YES. Pero podemos modificarlo si deseamos crear un servidor que sólo permite bajar archivos.

· **ftpd_banner:** este parámetro sirve para establecer el mensaje de bienvenida que será mostrado cada vez que un usuario acceda a nuestro servidor. Podemos definir cualquier frase breve que nos parezca conveniente y que de poca información sobre nuestro servidor.

· **anon_upload_enable:** Esto es la capacidad de que los usuarios conectados como anónimos puedan subir archivos.

· **listen_ipv6:** esto es si deseamos escuchar en IPV6 es importante que si no tenemos IPV6 desactivemos este feature ya que sino vsftpd no iniciará.

Conectándonos por primera vez

Primero en el servidor debemos de reiniciar el servicio de VSFTPD:

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

Desde el cliente ejecutamos el cliente FTP por consola:

```
apt-get install ftp
```

Y luego intentamos conectarnos:

```
ftp localhost
Name (localhost:adrabenche): carreralinux
ftp>
```

Límites por defecto

Cuando nos conectemos, nos vamos a encontrar con un inconveniente por defecto que es que los usuarios pueden navegar por nuestro sistema de archivos.

Por más que tengamos seteado los permisos de accesos a otros archivos de manera efectiva, es preferible no tener a usuarios accediendo a lugares donde no es necesario que accedan:

```
ftp> passive
ftp> pwd
257 “/home/carreralinux”
```

Como pueden ver, ingresamos algo llamado **passive** que veremos más adelante; pero como pueden ver, aparece el **path** correspondiente.

Estableciendo límites para los usuarios

De modo predefinido los usuarios del sistema que se autentiquen tendrán acceso a otros directorios del sistema fuera de su directorio personal.

Si lo que queremos es limitar a los usuarios a solo poder utilizar su directorio personal, tendremos que utilizar el parámetro **chroot_local_user.**

Este parámetro habilitará la función de chroot() y los parámetros chroot_list_enable y chroot_list_file para establecer el archivo con la lista de usuarios que quedarán excluidos de la función chroot().

Vamos a aclarar un poco más respecto de este tema. Para lograr esta limitación de usuarios nos vamos a basar en dos herramientas **jailing** y **chroot**.

Jailing es restringir un usuario o proceso a un directorio/s específicos y luego **chroot** es cambiar el directorio raíz de un proceso.

Por lo que la limitación de usuarios podríamos definirla como r**estringir el proceso de VSFTPD al directorio del usuari**o, mediante la modificación del directorio raíz del mismo.

Para realizar esto, debemos modificar lo siguiente en la configuración.

Si seteamos los parámetros de la forma que sigue:

```
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list
```

Y luego creamos el archivo **chroot_list**:

```
#  touch /etc/vsftpd.chroot_list
```

Para finalmente reiniciar el servicio:

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

Cada vez que un usuario local se autentique en el servidor FTP, solo tendrá acceso a su propio directorio y lo que este contenga. No nos tenemos que olvidar de crear **/etc/vsftpd/vsftpd.chroot_list**, ya que de otro modo no arrancará VSFTPD.

Por último, si quisiéramos que un usuario accediera sin **jail ni chroot**, simplemente debemos agregarlo al **/etc/vsftpd/vsftpd.chroot_list** y luego reiniciar el servicio.

Corrigiendo el error de chroot

Al momento de reiniciar el servicio, cuando intentemos acceder desde un cliente nos arrojará el siguiente error:

```
500 OOPS: vsftpd: refusing to run with writable root inside chroot()
```

Nos está indicando que no va a escribir si el directorio home del FTP tiene permisos de escritura.

Esto es algo propio de las últimas versiones, por lo que deberemos crear una carpeta dentro del home con permisos de escritura:

```
# mkdir /home/carreralinux/ftp
# chown carreralinux:carreralinux /home/carreralinux/ftp
# chmod 770 /home/carreralinux/ftp
```

Y debemos quitarle los permisos de escritura a la carpeta home:

```
# chmod 555 /home/carreralinux
```

Ahora si, desde el cliente intentemos ingresar nuevamente desde el cliente:

```
ftp> pwd
257 “/”
```

Operaciones básicas

Una vez que ingresamos, lo primero que tenemos que ejecutar es el **comando hash**. Este comando **imprimirá en la pantalla por cada 1024 bytes subido o descargado**. Esto será un testigo que nos indicará si se está moviendo datos. Por ejemplo, para archivos muy grandes, si no tenemos algún testigo que nos indique si se están moviendo datos no podremos saber si es que la conexión se colgó o está funcionando:

```
ftp> passive
ftp> hash
ftp> cd ftp
ftp> put large.file
local: large.file remote: large.file
200 PORT command successful. Consider using PASV.
###...
```

Listar archivos

```
ftp> ls
```

Descargar

```
ftp> get large.file
```

Como regla básica, si a los comandos le agregamos "**!**" **ejecuta el mismo comando de manera local**:

```
ftp> !pwd
/tmp
ftp> !ls
Administrador_de_Redes_Clase5_1.odt
```

Control del ancho de banda

Si queremos **limitar la tasa de transferencia de bytes por segundo para los usuarios anónimos** utilizaremos el **parámetro anon_max_rate**. Es sumamente útil en servidores FTP de acceso público.

En el siguiente ejemplo limitaremos la tasa de transferencia a 5 Kb por segundo para los usuarios anónimos:

```
anon_max_rate=5120
```

Con el **parámetro local_max_rate** haremos lo mismo que **anon_max_rate**, pero será válido para los usuarios locales del servidor.

Veamos cómo tendríamos que definir este parámetro:

```
local_max_rate=5120
```

Con el parámetro max_clients establecemos el número máximo de clientes que podrán acceder simultáneamente al servidor FTP.

En la línea que sigue limitaremos el acceso a 5 clientes simultáneos:

```
max_clients=5
```

Con el parámetro que sigue establecemos el número máximo de conexiones que se pueden realizar desde una misma dirección IP.

Tengamos en cuenta que algunas redes acceden a través de un servidor proxy o puerta de enlace y debido a esto podrían quedar bloqueados innecesariamente algunos accesos.

En ejemplo que sigue limitaremos el número de conexiones a 5:

```
max_per_ip=5
```

Levantando el servicio

La versión incluida en esta distribución puede inicializarse, detenerse o reinicializarse a través de un script similar a los del resto del sistema.

En este caso lo haremos con el comando:

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

O bien

```
# systemctl start vsftpd
```

¿Qué es el modo pasivo?

Uno de los términos que solemos escuchar cuando hablamos de problemas con el FTP es el de **passive mode**. Veremos aquí de que se trata, y por qué es la solución a muchos problemas de conectividad.

Como sabemos en toda transferencia FTP interviene un programa servidor y un programa cliente. El programa servidor se ejecuta dónde están almacenados los archivos que se quieren bajar (o donde se almacenarán los que deseamos subir) y el programa cliente es el programa FTP que ejecutamos desde la máquina local, para subir o bajar los archivos.

En este proceso de comunicación entre cliente y servidor, el cliente puede actuar en modo activo o en modo pasivo.

Una conexión FTP en modo pasivo usa dos puertos, es decir abre dos canales:
1.- El **puerto de comandos**, **command port** o **control port** (normalmente puerto 21) por donde se transferirán las órdenes.
2.- El otro es el **puerto de datos** (**data port**), por donde circulan los datos que integran los archivos (normalmente el puerto 20, pero puede ser cualquiera por debajo del 1024).

¿Qué es el modo activo?

Cuando usamos FTP en modo activo (también considerado modo normal) se establecen **dos conexiones distintas**.

En primer lugar se establece una conexión para la transmisión de comandos (desde cualquier puerto de nuestro equipo inferior a 1024 hacia el puerto 21 del servidor) y por esa misma conexión, mediante un comando PORT se indica al servidor cuál es el puerto (distinto al 21) de nuestro equipo que estará a la escucha de los datos.

Entonces, si bajamos algún archivo, es el servidor el que inicia la transmisión de datos, desde su puerto 20 al puerto que le hemos indicado. Se llama modo activo precisamente por esto último que hemos dicho, porque **la transmisión de datos es iniciada como proceso distinto desde el servidor, hacia el puerto que le hemos indicado**.

Diferencias y problemas

En modo pasivo es siempre el programa cliente el que inicia la conexión con el servidor.

Al abrir una conexión FTP se abre primero una conexión de control (desde un puerto inferior a 1024 de la maquina local al puerto 21 del server).

Al pasar a modo pasivo (comando PASV), el cliente pide un puerto abierto al servidor (será otro puerto inferior al 1024 del server) y recibida la contestación, será el cliente el que establezca la conexión de datos al servidor a través de ese puerto.

En modo pasivo las conexiones son siempre abiertas por el equipo cliente, mientras que en modo activo se abren por el que envía los datos (el servidor si se trata de bajar archivos al equipo local, el cliente si se trata de subir archivos al servidor).

¿Y POR QUÉ SUPONE UN PROBLEMA EL MODO ACTIVO?

Como hemos visto, en el modo activo se abre una conexión para datos desde el servidor a la máquina cliente; esto es, una conexión de fuera a dentro.

Entonces, si la máquina cliente está protegida por un firewall, este filtra o bloquea la conexión entrante, al serle un proceso desconocido.

En modo pasivo es el cliente el que inicia ambas conexiones, de control y de datos, con lo cual el firewall no tiene ninguna conexión entrante que filtrar.

Si alguna vez tenemos problemas de conexión y sospechamos que es por un firewall, deberíamos buscar un programa cliente FTP que disponga de esta posibilidad (la inmensa mayoría).

Para terminar con el modo pasivo, parte de un log de conexión:

```
FTP > PASV
FTP < 227 Entering Passive Mode (222,222,222,22,196,39).
FTP > LIST
```

Como vemos el servidor reacciona al comando PASV entrando en modo pasivo e indicando una serie de números (entre paréntesis la segunda línea).

Los cuatro primeros bloques de números se corresponden con la IP del servidor. Los dos números restantes son el puerto que el servidor abre a la escucha para el canal de datos. El puerto está indicado mediante dos números de 8 bits.

El servicio de FTP es el que hay que monitorear más de cerca, ya que normalmente es la puerta de acceso a los problemas en nuestra red.

Porque mal configurado puedo ser usado como reservorio de programas, archivos y demás por parte de los usuarios de nuestra red, pero lo que aún es más peligroso es que sea usado por personas ajenas a nuestra red para fines non santos.

Si quisieramos **forzar los puertos del modo pasivo**, simplemente agregaremos estas lineas:

```
pasv_min_port=19990
pasv_max_port=20000
```

Usuario “anonymous”

Muchas veces solo necesitamos que se descarguen archivos. Por ejemplo, si deseamos que los usuarios descarguen documentación o aplicaciones por citar algunos ejemplos, nos alcanzará con dar un espacio de solo lectura.

En resumen, si nos conectamos con el usuario anonymous nos dejará en una carpeta en la que solo podremos subir archivos, pero no descargarlos.

Lo primero a tener en cuenta es la carpeta en la que se encontrará, por ejemplo en **/var/vsftpd/anonymous**:

```
kdir -p  /var/vsftpd/anonymous
```

Luego vamos a especificar que activamos al usuario anonymous y que ingresará a la carpeta correspondiente. Editamos los archivos de configuración donde dejaremos las siguientes líneas como indicamos debajo:

```
anonymous_enable=YES
anon_upload_enable=NO
anon_root=/var/vsftpd/anonymous
```

Donde:

- **anonymous_enable:** activamos al usuario anonymous.
- **anon_upload_enable:** denegamos la posibilidad de que un usuario suba archivos.
- **anon_root:** directorio raíz donde nos situaremos apenas ingresemos.

Para conectarnos:

```
# ftp localhost
Name (localhost:adrabenche): anonymous
Password:
```

Si queremos ver el directorio, obviamente nos mostrará el chroot:

```
ftp> pwd
257 “/”
```

Veamos que documentación hay:

```
ftp> ls
-rw-r--r--    1 1000     1000     52428800 Sep 07 13:55 large.file
```

Y descargamos:

```
ftp> hash
ftp> get  large.file
```

Si intentamos subir un archivo, vamos a ver que nos devolverá acceso denegado.