Qué

Se ha de instalar y configurar un servidor FTP que implemente medidas de seguridad y criptografía, ya sea por SSH (SFTP) o TLS (FTPS). Además, se tienen que reponder algunas preguntar teóricas.

Los clientes han de poder acceder al servidor a través de usuarios locales del servidor, usuarios virtuales exclusivos del servicio FTP, y de un usuario anónimo.

Para qué

El objetivo es crear un servidor FTP en el que la información y contraseñas no viajen por la red en forma de texto plano.

Cómo

Instalación

El servidor que se va a instalar es `proftpd`, pues no sólo es uno de los más usados, sino que también tiene todas las características necesarias para llevar a cabo este ejercicio: SFTP, FTPS, Usuarios virtuales (tiene hasta una herramienta para crear archivos de usuarios y grupos) y usuarios anónimos.

La instalación en Ubuntu se hace a través del gestor de paquetes, ejecutando el comando `sudo apt install proftpd-basic -y`.

Además, convendría desactivar el servidor SFTP que viene incorporado con `openssh-server`, para evitar que los usuarios entren a ese servidor SFTP, que se dejará sin configurar. Para hacerlo se tiene que comentar la línea del fichero `/etc/ssh/sshd_config` en la que pone `Subsystem sftp /usr/lib/openssh/sftp-server`

El siguiente paso hacer una copia de seguridad de los archivos de configuración. El directorio en el que se encuentran es `/etc/proftpd`, y los archivos a copiar son `proftpd.conf` y `modules.conf`.

```
sudo cp /etc/proftpd/proftpd.conf{,.bck}
sudo cp /etc/proftpd/modules.conf{,.bck}
```

Tras tenerlo todo instalado y preparado, se reinicia el servicio y se comprueba si está activo:

```
sudo service proftpd restart
sudo service proftpd status
```

Parámetros generales de configuración

- `Include /etc/proftpd/modules.conf`: Importa los módulos de proftpd.
- `ServerType standalone`: Indica que el servidor ha de actuar por su cuenta, en vez ser iniciado y estar subyugado por un servidor inetd[1].
- `DefaultServer on`: Define la configuración actual como la que se usará en caso de que no se encuentre un VirtualHost que corresponda a la IP y puerto al que se le ha hecho la petición.
- `ShowSymlinks on`: Muestra los enlaces simbólicos al ejecutar comandos como `ls`.
- `MaxInstances 30`: Cantidad máxima de conexiones abiertas que puede tener el servidor.
- `User proftpd`: El usuario que usará proftpd.
- `Group nogroup`: El grupo que usará proftpd.

[1] https://en.wikipedia.org/wiki/Inetd

- `Umask 022 022`: Los permisos que se le reducen a los permisos que tenga el usuario y grupo en el sistema, en el formato que tiene `chmod`, invertido. Es este caso se quitan permisos de escritura al grupo y al usuario que tengan.
- `TransferLog /var/log/proftpd/xferlog`: Ruta al fichero de registro sobre las tranferencias de ficheros.
- `SystemLog /var/log/proftpd/proftpd.log`: Ruta al fichero de registro general.
- `DefaultRoot ~`: Enjaula a los usuarios en su carpeta personal.
- `AllowOverwrite on`: Permite que se puedan sobreescribir ficheros ya existentes.

¿Es posible autentificarse sin contraseña en FTPS o SFTP?

En SFTP si es posible, ya que usa los mismos sistemas de autentificación que SSH, incluida la autentificación por clave pública. Para ello primero hay que habilitar el módulo de SFTP en modules.conf. Después se tienen que especificar ciertos parámetros de configuración desde proftpd.conf.

En FTPS se puede permitir el acceso a todos los clientes que se conecten por TLS o SSL sin pedir usuario ni contraseña.

Autentificación por par de claves en SFTP

Primero, se tiene que abrir /etc/proftpd/modules.conf y comprobar que la línea `LoadModule mod_sftp.c` existe y está descomentada. Luego, si existe el paquete `proftpd-mod-crypto`, se ha de instalar (Esto varía según la versión de Ubuntu, en 20.04 no debería ser un problema).

Ahora, dentro de una etiqueta `IfModule` con `mod_sftp.c` como atributo, se ponen los siguientes parámetros de configuración:

- `SFTPEngine on`: Habilita SFTP.
- `Port 2222`: Puede ponerse o no. Si se pone, sobreescribirá cualquier puerto definido anteriormente a no ser que éste esté dentro de un VirtualHost.
- `SFTPLog /var/log/proftpd/sftp.log`: La ruta de los registros relacionados específicamente con SFTP
- `SFTPHostKey /etc/proftpd/keys/ssh_host_ecdsa_key`: La ruta de la clave privada que identifica al servidor SFTP.
- `SFTPHostKey /etc/proftpd/keys/ssh_host_rsa_key`: Igual que la anterior, pero en este caso se usa otro algoritmo de encriptación. No es estrictamente necesario, pero en el caso de que el cliente no pueda entender la clave anterior.
- `SFTPAuthMethods publickey password`: Los métodos de autentificación que permite el servidor SFTP. El orden importa.
- `SFTPAuthorizedUserKeys file:/etc/proftpd/authorized_keys/%u`: La ruta al archivo en el que se guardan las claves públicas válidas.
  Se pone este valor para que estén centralizadas en un solo directorio (`%u` es el nombre del usuario con el que se intenta iniciar sesión) y las gestione un administrador. Se tiene que crear el directorio indicado y copiar las claves válidas para cada usuario dentro de él en un archivo con el nombre del usuario.
- `SFTPCompression delayed`: Habilita la compresión de los datos tranferidos.

El siguiente paso es crear el par de claves que sirven para identificar al servidor. Tienen que estar en un formato compatible con el RFC4716.

```
cd /etc/proftpd/
sudo mkdir keys
sudo ssh-keygen -m PEM -f keys/ssh_host_ecdsa_key -N '' -t ecdsa
sudo ssh-keygen -m PEM -f keys/ssh_host_rsa_key -N '' -t rsa
```

Por último, se tiene que obtener la clave pública del cliente. El problema es que la clave pública también ha de ser compatible con el RFC4716, pues no lo serán por defecto.

Para ello, desde el usuario deseado en el cliente, se tiene que ejecutar `ssh-keygen -e -m -f <clave_privada>|tee $USER`, donde `<clave_privada>` es la ruta a la clave privada y $USER se puede sustituir por el nombre del usuario en caso de que se esté ejecutando desde un usuario diferente.

Después este archivo se tiene que copiar al directorio del servidor especificado en `SFTPAuthorizedUserKeys`.

¿Es posible tener un servidor FTPS o SFTP y clientes FTP sin seguridad?

Con un servidor SFTP sólo puedes iniciar conexiones con clientes compatibles con SFTP, que son, obligatoriamente, todos seguros porque usan SSH.

Con un servidor FTPS depende del servidor y de la configuración. Muchos servidores FTPS, incluido proftpd tienen una opción que permite obligar o no a que las conexiones vayan por TLS (`TLSRequired`). Si se obliga a hacer esto es absolutamente necesario usar un cliente FTP que soporte TLS, y por ende, sea seguro.

¿Implementan SFTP o FTPS el nivel de sesión?

FTP, según el Punto 1.1.3 del RFC1122 pertenece a la capa de aplicación del modelo TCP/IP, que es el modelo que se usa en la práctica.

Esta capa se corresponde a las capas de aplicación, presentación y sesión del modelo OSI, por lo que FTP **podría** abarcar total o parcialmente todas estas capas, incluida la de sesión.

Aunque esto no implica que realmente FTP entre en la capa de sesión del modelo OSI, y el consenso general es que FTP *"plano"* no lo hace.

Como estos servicios no se desarrollan con el modelo OSI en mente, lo mejor para saber si encajan es definir que hace la capa de sesión para ver si lo hacen también estos servicios:

La capa de sesión se encarga de proporcionar servicios para establecer, mantener, ordenar y liberar conexiones entre dos entidades de presentación[2].

Esto es algo que SSH, y por ende SFTP, defnitivamente hace, como menciona el RFC de SSH.

De hecho, esto es algo que hacen tanto SSL como TLS[3], lo cual no sólo respalda la afirmación de que SFTP forma parte de la capa de sesión, sinó que también indica que FTPS forma parte de la capa de sesión de igual modo.

¿Implementan SFTP o FTPS el nivel de presentación?

Tanto el SFTP y el FTPS implementan el nivel de presentación, ya que transfieren archivos de forma segura entre nodos usando encriptación, para lo que se combinan con los protocolos SSL y/o TLS, que forman parte de esta capa.

Usuarios anónimos en proftpd

Los usuarios anónimos no existen en SFTP, lo más similar es crear un usuario virtual con una contraseña vacía, pero eso da problemas según el cliente desde el que se intente acceder. Por tanto, se tiene que usar

[2] Punto 7.3.2 de este documento de la ITU https://www.itu.int/rec/T-REC-X.200-199407-I/en
[3] https://www.websecurity.digicert.com/security-topics/how-does-ssl-handshake-work

FTP o FTPS, pero como el resto del servidor funciona por SFTP, se tiene crear un VirtualHost en el que se usará un puerto y configuración diferentes.

Autentificación sin contraseña en FTPS.

Se ha de comprobar que la línea `LoadModule mod_tls.c` exista y esté decomentada en el fichero `/etc/proftpd/modules.conf` y, al igual que con SFTP instalar el paquete `proftpd-mod-crypto` en caso de que exista en los repositorios.

Dentro del VirtualHost se ponen 3 cosas:

Configuración general

- `Port 21`: Puerto por el que actúa el servidor. No ha de repetirse.
- `DefaultRoot /srv/ftp`: Directorio raíz por defecto de los usuarios.

Además, se deniega el acceso a todos los usuarios, solo se podrá acceder de forma anónima.

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

Configuración de TLS Después, dentro de una etiqueta `IfModule` con `mod_tls.c` como atributo, se ponen los siguientes parámetros de configuración:

- `TLSEngine on`: Habilita FTPS.
- `TLSLog /var/log/proftpd/tls.log`: Ruta al fichero de registros.
- `TLSProtocol TLSv1 TLSv1.1 TLSv1.2 SSLv23`: Versiones de los protocolos TLS y SSL que admite.
- `TLSRSACertificateFile /etc/ssl/private/proftpd.crt`: Ruta al certificado.
- `TLSRSACertificateKeyFile /etc/ssl/private/proftpd.key`: Ruta a la clave del certificado.
- `TLSRequired on`: Obliga a los clientes a usar TLS.
- `TLSVerifyClient off`: Permite a los clientes entrar al servidor FTPS sin usuario o contraseña, siempre y cuando se conecten con TLS.
- `TLSOtions NoSessionReuseRequired`: Permite hacer `ls` como usuario anónimo en algunos sistemas que no lo admiten.

El siguiente paso es crear los certificados del servidor FTPS:

```
sudo openssl genrsa -out /etc/ssl/private/proftpd.key 4096

sudo openssl req -new -x509 -days 1460 -key /etc/ssl/private/proftpd.key \
-out /etc/ssl/certs/proftpd.crt
```

Y por último, se reinicia el servicio con `sudo service proftpd restart`.

Configuración de los usuarios anónimos Para habilitar los usuarios anónimos en proftpd se tienen que añadir una etiqueta de Anonymous que tendrá como atributo el directorio personal del usuario anónimo (por defecto es `~ftp`, que corresponde a `/srv/ftp`). Dentro de esa etiqueta se añade, antes que nada, una etiqueta para permitir el acceso al usuario anónimo, ya que se ha denegado el acceso a todos los usuarios previamente:

```
<Limit LOGIN>
    AllowAll
</Limit>
```

Despuén se añaden los siguientes parámetros de configuración:

- `User ftp`: El usuario del sistema como el que se iniciará el usuario anónimo.
- `Group nogroup`: El grupo del sisteema al que pertenecerá el usuario anónimo.
- `UserAlias anonymous ftp`: Los nombres del usuario anónimo.
- `DirFakeUser on ftp`: Cambios estéticos, ofusca el propietario de los ficheros.
- `DirFakeGroup on ftp`: Cambios estéticos, ofusca el grupo propietario de los ficheros.
- `RequireValidShell off`: Permite iniciar sesión a usuarios con un shell que no esté en /etc/shells, hace falta cuando se usan usuarios con shell /bin/false, un shell que únicamente devuelve el valor false y sale.
- `MaxClients 10`: No es necesario. Limita los usuarios anónimos que pueden estar conectados simultáneamente.
- `DisplayLogin welcome.msg`: Muestra el contenido de welcome.msg al iniciar sesión.
- `DisplayChdir .message`: Muestra el conenido de .message al cambiar de directorio.

Además de lo anterior, para denegar los permisos de lectura independientemente del directorio o archivo, se añade está etiqueta:

```
<Directory *>
    <Limit WRITE>
      DenyAll
    </Limit>
</Directory>
```

Enjaular a los usuarios en su carpeta personal en proftpd

Simplemente hay que poner establecer escribir `DefaultRoot ~` en /etc/proftpd/prodftpd.conf. Esto hace que al conectarse por FTP, FTPS o SFTP el directorio personal del usuario haga las veces de raíz. Se puede comprobar si funciona o no usando el comando `pwd`.

Creación de usuarios virtuales en proftpd.

Para crear usuarios virtuales se usa el comando `ftpasswd`, que forma parte del paquete `proftpd-basic`.

Es conveniente a la hora de gestionar permisos usar un grupo del sistema que compartirán todos los usuarios virtuales, para que tengan los mismos permisos generales.

```
sudo groupadd -g 2001 ftpgroup
```

Luego se tiene que hacer una carpeta y darle permisos al usuario virtual que se va a crear. El sistema operativo saca los nombres del fichero `/etc/passwd`, así que no podemos usar el nombre del usuario virtual para asignar permisos. Pero en realidad, el kernel de Linux no usa los nombres de usuario para asignar permisos, sinó la ID de usuario. Por tanto, se pueden definir los permisos de esta manera:

```
sudo mkdir /home/user
sudo chown 2001:ftpgroup -R /home/user
```

Ahora se tienen que hacer los propios usuarios. Para ello se ejecutan los siguientes comandos:

```
sudo ftpasswd --passwd --file=/etc/proftpd/ftpd.passwd --name=user --uid=2001 --gid=2001 --
home=/home --shell=/bin/false
sudo ftpasswd --group --file=/etc/proftpd/ftpd.passwd --name=users --gid=2001 --member=user
```

Así se crean dos archivos en el directorio de configuración de proftpd: Uno que contiene usuarios y otro que contiene grupos, ambos con el mismo formato que tienen los archivos /etc/passwd y /etc/group,

respectivamente. Es importante destacar que la ID del grupo debe coincidir con la del grupo del sistema para que compartan permisos, y que la ID del usuario y la del grupo coincidan con las que tienen permisos en la carpeta del usuario.

Para modificarlos, se tiene que volver a usar el comando `ftpasswd` o cambiar los permisos antes y después de modificarlo manualmente, pues ni tienen ni pueden tener permisos de escritura para ningún usuario.

Finalmente, se tiene que reiniciar el servicio tras especificar en la configuración qué archivos ha de usar `proftpd` para los usuarios y grupos virtuales, añadiendo estas líneas al fichero de configuración (En este caso, ha sido añadido dentro de la etiqueta `<IfModule mod_sftp.c>`):

```
AuthUserFile /etc/proftpd/ftpd.passwd
AuthGroupFile /etc/proftpd/ftpd.group
```

Conclusión

El mayor problema que he tenido con esta práctica es que hay puntos en los que nos quedamos atascados y no tenemos los conocimientos técnicos requeridos para salir de esa situación.

El líneas generales, está práctica no me produce sentimientos especialmente negativos ni positivos, sólo indiferencia.