

Enjaulado SFTP

Descripción

El enjaulado es una técnica mediante la cuál se proporciona al usuario una acceso limitado al sistema de ficheros linux. Normalmente se realiza el enjaulado dentro de la *home* de cada usuario y el acceso se limita a ésta, de forma que el directorio raíz para un usuario enjaulado / se corresponde con el directorio *home* real /home/xxxxxx del sistema de archivos.

Como vimos en el apartado correspondiente a la securización del OpenSSH los usuarios no disponen de un acceso remoto real al sistema sino que se limita el acceso al servicio sftp a través de una dummy shell, por ello el enjaulado que realizaremos también estará limitado a este servicio.

Actuaciones

Antes de iniciar el proceso para adquirir e instalar nuestro certificado digital, debemos comprobar si el *Módulo SSL* de Apache está instalado y si no es así realizar la instalación a través de yum.

Para proceder a realizar los cambios vamos a requerir un compilador C adecuado, para ello se recomienda realizar la instalación de gcc o cpp mediante yum. El rpm de gcc depende de otros que también deben instalarse (cpp, glibc-devel, glibc-headers, libgomp)

También se requiere zlib-devel (yum install zlib-devel.x86_64) y OpenSSL Devel (yum install openssl-devel.x86_64), que a su vez requiere los paquetes siguientes: e2fsprogs-devel, y krb5-devel

1. Recompilando sftp-server

Verificamos la versión de OpenSSH que actualmente está presente en nuestro servidor., mediante el comando:

```
rpm -qa | grep openssh
> openssh-clients-4.3p2-18.fc6
> openssh-4.3p2-18.fc6
```

```
> openssh-server-4.3p2-18.fc6
```

Descargamos de la página oficial la versión correspondiente, mediante el comando *wget*.

```
wget  
ftp://ftp.rediris.es/mirror/OpenBSD/pub/OpenBSD/OpenSSH/portable/openssh-4.3p2.tar.gz
```

Extraemos y descomprimos los archivos que hemos descargado

```
tar -xzf openssh-4.3p2.tar.gz  
cd openssh-4.3p2
```

Configuramos openSSH, en nuestro caso solamente necesitamos sustituir el archivo binario de sftp-server (sftp-server.real) respetando el resto de la instalación del servidor OpenSSH, para ello se deben configurar los archivos de código con las actuales opciones.

La configuración típica de un sistema redhat o fedora es la siguiente

```
./configure --prefix=/usr --libexecdir=/usr/libexec/openssh  
--sysconfdir=/etc/ssh --mandir=/usr/share/man
```

En este punto debemos parchear los archivos de código de OpenSSH con el archivo de diferencias chroot (sftp-chroot.diff)

```
patch < ../sftp-chroot.diff  
patching file sftp-server.c  
Hunk #1 succeeded at 26 with fuzz 2 (offset -7 lines).  
Hunk #2 succeeded at 1039 with fuzz 1 (offset 13 lines).  
Hunk #3 succeeded at 1075 with fuzz 2 (offset 4 lines).
```

A continuación compilamos todo el código para generar los ejecutables

```
make
```

Este comando, a diferencia del comando `make install` no realiza la instalación, sino que deja los archivos binarios generados en el directorio en donde se está compilando..

Ahora solo nos queda reemplazar el nuevo sftp-server que hemos generado (con las opciones de *chroot*) por los actuales. Como anteriormente ya hemos modificado el archivo `/usr/libexec/openssh/sftp-server`, ahora solo hace falta copiar el archivo generado con otro nombre (por ejemplo `stop-server.chroot`)

y modificar el script en bach (contenido en sftp-server) para ejecutar el nuevo sftp.

```
cp sftp-server /usr/libexec/openssh/sftp-server.chroot
```

2. Enjaulando al Usuario

Ahora sólo queda proceder al enjaulado de los usuarios, este proceso se realiza señalándolo en ruta de la *home* del usuario del archivo `/etc/passwd`.

Este paso requiere haber incluido con posterioridad la shell en el archivo de shells reconocidas por el sistema, si no hemos realizado este paso posteriormente lo podemos realizar ahora con el siguiente comando:

```
echo "/bin/sftpsh" >> /etc/shells
```

El formato específico para el enjaulado es el siguiente:

```
/<chroot root>/.<home>
```

Chroot root indica el directorio de donde el usuario no pondrá escapar, en nuestro caso es la *home* del usuario `/home/aitor`, además debemos indicar del nuevo sistema de archivos virtual para este usuario cuál directorio se corresponde con la *home* de nuestro usuario, en nuestro caso la *home* del usuario y el directorio de usuario coinciden, por lo que indicamos `/`

La línea del archivo `/etc/passwd` para el usuario aitor, sería:

```
aitor:x:502:502::/home/aitor/./:/bin/sftpsh
```

3. Comprobaciones

Ahora solo nos queda comprobar que el enjaulado se ha realizado correctamente..

Veamos como el acceso por ssh está prohibido para este usuario, el comando siguiente no debería permitir el acceso:

```
ssh aitor@localhost
```

En cambio, el acceso por ftp debe estar permitido pero el resultado debe ser un acceso enjaulado.

```
sftp aitor@localhost
```

Al pedir el directorio actual mediante el comando `pwd`, éste debería mostrar que es la raíz del sistema de archivos `/`

```
Connecting to localhost...
```

```
aitor@localhost's password:  
sftp> pwd  
Remote working directory: /  
sftp>
```

¿Que es iddover.net?

Iddover Hosting es una iniciativa española en el sector de servidores dedicados. Nace en contraposición de los servicios actuales de hosting . Pretende ofrecer un hosting de alta calidad gestionable a través de un potente panel propio que se adecua a los requisitos de seguridad y que proporciona al cliente potentes herramientas al alcance de un solo click

Desde 69€ dispondrás de un servidor dedicado administrado auditado con todas las medidas de seguridad ya aplicadas.

<http://www.iddover.net>