GUÍA RÁPIDA DE CONFIGURACIÓN DE HTACCESS

Autor	VenCERT
Fecha última edición	20/08/2012
Versión y revisión	2.0

Control de versiones

Versión	Áreas Modificadas	Descripción del cambio	Autor	Fecha
1	Todas	Version Inicial	VenCERT	23/02/2015

Índice

INTRODUCCIÒN

Un archivo .htaccess (hypertext access), también conocido como *archivo de configuración distribuida*, es un archivo especial, popularizado por el Servidor HTTP Apache que permite definir diferentes directivas de configuración para cada directorio (con sus respectivos subdirectorios) sin necesidad de editar el archivo de configuración principal de Apache.

Este archivo se suelen emplear para:

- Redireccionar tráfico
- Reescritura de URL
- Compresión de ficheros para reducir el tiempo de carga
- Bloquear acceso a directorios

El presente manual contiene información sobre el procedimiento Y las diferentes utilidades que nos puede ofrecer este archivo de configuración

PROCEDIMIENTO

Modificar el archivo /etc/apache2/site-enable/default o donde se encuentre configurado el sitio,

<Directory "/var/www/administrator"> **(indica la ruta absoluta del directorio a blindar)**
Options None
AllowOverride None
Order allow,deny
Allow from all
AuthName "Acceso a la Administracion del portal" **(es el nombre aparece en el cuadro de autenticacion)**
AuthType Basic
AuthUserFile /etc/apache2/.htpasswd **(es el archivo donde se encuentra los usuarios y los password la ubicacion puede ser en el lugar de preferencia)**.
require valid-user
</Directory>

Al finalizar borrar comentarios y guardar

para crear el archivo donde se guardarán las contraseñas se realizá lo siguiente:

- htpasswd -c /etc/apache2/.htpasswd nombreusuario1

luego de esto se le requerirá el passwd del usuario nuevo para agregar otro usuario ejecute lo siguiente

- htpasswd /etc/apache2/.htpasswd nombreusuario2

luego de esto se le requerirá el passwd del usuario nuevo2, Recuerde que la opcion "-c" es solo para el primer usuario para crear el archivo.

UTILIDADES DE .HTACCESS

➔ AUMENTAR LA VELOCIDAD DE NUESTRA WEB:

Una de las funciones más importantes que nos puede ofrecer es que nos permite utilizar la caché para aumentar considerablemente el rendimiento de carga de nuestra web.

```
# 1 Año
<FilesMatch "\.(ico|pdf|flv)$">
Header set Cache-Control "max-age=29030400, public"

# 1 Semana
<FilesMatch "\.(jpg|jpeg|png|gif|swf)$">
Header set Cache-Control "max-age=604800, public"

# 1 Minuto
<FilesMatch "\.(html|htm|php)$">
Header set Cache-Control "max-age=60, private, proxy-revalidate"
```

➔ *EVITAR EL HOT-LINKING EN TU WEB:*

El **hot-linking** o **hotlink** es una técnica que consiste en aprovecharse de las imágenes almacenadas en tu web. De esta forma pueden enlazar las imágenes de tu web y hacer que sea tu servidor el que corra con la carga de mostrarlas. Para evitarlo podemos usar el siguiente código

```
RewriteEngine On
#Sustituimos ?miweb\.com/ con la url de nuestra página
RewriteCond %{HTTP_REFERER} !^http://(.+\.)?miweb\.com/ [NC]
RewriteCond %{HTTP_REFERER} !^$
#Sustituimos/img/noHotLink.jpg con la ruta de la imagen que queramos
prohibir
RewriteRule .*\.(jpe?g|gif|bmp|png)$ /img/noHotLink.jpg [L]
```

➔ *CONTROL DE ACCESO A CARPETAS:*

Gracias a esta funcionalidad, podremos disponer de un control absoluto sobre el acceso a nuestros ficheros, un ejemplo:

```
#Denegar el acceso a cualquier carpeta deny from all
#permitir el acceso a una IP específica
deny from all
allow from 190.18.19.20
#bloquear el acceso a un archivo en concreto
Order allow,deny
Deny from all

#Acceso rangos de IPs
allow from 192.168.0.0/50
```

➔ *BLOQUEAR DIRECCIONES IP CON UNA LISTA NEGRA:*

Puede evitar las peticiones de IPs con intenciones turbias incluyéndolas en una lista negra. Este es el código

```
order allow, deny
allow from all
deny from 111.111.111.1
deny from 000.000.000.0
```

➔ *PÁGINAS DE ERRORES PERSONALIZADOS:*

Esta función es muy útil para evitar que sean los servidores los que manejen los errores más comunes 404, 403, etc. De esta forma puedes presentarlos con una página de confianza al usuario.

```
ErrorDocument 403 /error/403.html
ErrorDocument 404 /error/404.html
ErrorDocument 500 /error/500.html
```

➔ *AHORRO DE ANCHO DE BANDA:*

Para ahorra el ancho de banda también podemos habilitar la compresión de datos con este código:

```
php_value zlib.output_compression 16386
```

➔ *BLOQUEAR ROBOTS:*

Para evitar que spiders innecesarios consuman recursos de tu servidor, podemos usar el siguiente código

```
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} ^BlackWidow [OR]
RewriteCond %{HTTP_USER_AGENT} ^Bot\ mailto:craftbot@yahoo.com
[OR]
RewriteCond %{HTTP_USER_AGENT} ^ChinaClaw [OR]
RewriteCond %{HTTP_USER_AGENT} ^Custo [OR]
RewriteCond %{HTTP_USER_AGENT} ^DISCo [OR]
RewriteCond %{HTTP_USER_AGENT} ^Download\ Demon [OR]
RewriteCond %{HTTP_USER_AGENT} ^eCatch [OR]
RewriteCond %{HTTP_USER_AGENT} ^EirGrabber [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailSiphon [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailWolf [OR] entre otros...
```

➔ *EVITAR MOSTRAR LAS WWW:*

Para evitar que tu página web en vez de http://www.xyz.com tenga el formato http://xyz.com incluiremos el siguiente código:

```
Options +FollowSymlinks
RewriteEngine on
RewriteCond %{http_host} ^www\.tuWeb\.com[nc]
RewriteRule ^(.*)$ http://tuWeb.com/$1 [r=301,nc]
```

Estos son solo unos ejemplos de las posibilidades que ofrece el fichero de configuración .htaccess.