# Mejorando la seguridad en Apache2 .htaccess

En una entrada anterior enseñamos como configurar mod_rewrite en nuestro .htaccess para poner a "punto de caramelo" nuestro Apache2 pero ahora es necesario preveniros de "haber abierto la caja de Pandora": nuestro servidor web está expuesto a tácticas de intromisión y hasta toma de control por atacantes con malas intenciones. Sirva pues la presente para complementar y ayudaros a proteger vuestros equipos.

## Prerequisitos.

Es bueno que leaís nuestro tutorial para configurar mod_rewrite pero si soís más avanzado os decimos de una vez que tenemos configurado ya un servidor Apache2 sobre Ubuntu 16 en una red de área local con la dirección 192.168.1.47

Para ser más exactos, corremos Apache 2.4 y podemos conocer cuál versión tenemos instalada abriendo una ventana terminal y ganando acceso como "root" o administrador:

```
root@KEVIN:/home/jimmy# apache2 -v  Server version: Apache/2.4.18 (Ubuntu
)  Server built: 2016-07-14T12:32:26
```

Otra cosa que tenemos activado es el tratamiento de archivos .html o .htm como .php *lo cual abre la posibilidad a que nuestra seguridad quede expuesta si alguien logra acceder a escribir en nuestro .htaccess* (de nuevo os recordamos que todo está explicado en nuestra entrada anterior sobre como configurar mod_rewrite). Eso quiere decir que tenemos la siguiente instrucción en el archivo .htacces:

```
AddType application/x-httpd-php .html .htm
```

Al estar así configurado pasamos a estudiar las posibles brechas en seguridad.

## Lo bueno, lo malo y lo feo.

Recordando la famosa película de vaqueros así clasificaremos las tretas y estrategias, ***poned atención que cada una de ellas es un caso particular a estudiar, pausar y seguir adelante para, al final, tener un panorama completo de la estrategia a seguir que más convenga a***

***nuestro entorno de trabajo.***

Empezaremos por lo malo, luego lo feo y de último lo bueno que podemos hacer.

## Lo malo.

**php_value auto_append_file.**

Vale decir que este comando **php_value** bien puede estar ubicado ya sea en "PHP config", "virtualhost settings" o en un archivo **.htaccess** que es el caso que nos ocupa y más específicamente el valor **auto_append_file** porque coloca al final de nuestra página web (en este ejemplo didáctico el archivo **index.html** que trae por defecto Apache al ser instalado) los valores que tengamos en nuestro archivo **hosts** con la siguiente instrucción:

```
php_value auto_append_file /etc/hosts
```

El resultado sería el siguiente:

Que para nuestro servidor de pruebas nos aporta muy poca información **pero imaginad si lo tenéis en un servidor en producción y de paso sea compartido por otros sitios web, dicha información sería muy útil a los atacantes.**

Otra variante de esta treta es incluir al archivo .htacces en sí mismo y adicionalmente algún código php. Por ejemplo podemos agregar el comando **phpinfo();** con el cual podemos visualizar el entorno de variables con que trabajamos y buscar así software específico para nuestra configuración y poder de esta manera tomar control completo, si fueramos atacantes.

```
php_value auto_append_file .htaccess #
```