

Source Code Disclosure [S.C.D.]

By The-XC3LL

Se me ocurrió la idea de publicar acerca de esta vulnerabilidad no muy conocida (tal vez porque no es de las que consideramos como de 1ª clase, tales como RFI, LFI, o SQL Injections pero que aún así puede suponer un gran impacto sobre la seguridad de una web) al ver el exploit codeado por SetH.

La vulnerabilidad en sí consiste en permitir ver el código fuente de archivos alojados en el servidor, conllevando a que un atacante malicioso obtenga información sensible, tales como los datos de las DBs (nombre de usuario, password, etc) o poder ver el funcionamiento de una tercera aplicación que esté en el servidor, permitiendo buscar vulnerabilidades de forma más fácil.

Nosotros vamos a diferenciar entre dos tipos de SCD, el primero que pasaremos inmediatamente a analizar se basa en la posibilidad de descarga a nuestro ordenador de archivos de la web. El segundo tipo del que hablaremos es caracterizado por ver el código fuente dentro de otro archivo, es decir, lo podremos ver desde el navegador sin necesidad de descargarlo.

Imaginemos un posible código fuente de una aplicación encargada de descargar los archivos que nosotros le indiquemos a través de una sencilla petición GET. El código vulnerable podría ser este:

```
<?php
/* Ejemplo de código vulnerable
=====
Paper sobre Source Code Disclosure
By Vengador de las Sombras
=====
Flaming our Skills TeaM (F.O.S.)
===== */
ob_start();
$archivo=$_GET['descargar'];
if ($archivo == " "){ //Si no se introduce nada en descargar, mostrara un error
echo '<script>alert("ERROR! No ha introducido ningún archivo!!")</script>';
exit;
}
header("Pragma: public");
header("Expires: 0");
header("Content-type: application/octet-stream");
header("Content-disposition: attachment; filename=$archivo");
header("Content-Length:".filesize("$archivo"));
readfile("$filename");
?>
```

El funcionamiento de esta sencillísima aplicación sería el de descargar un archivo que introduzcamos a través de la URL, por ejemplo

`http://SCD.com/index.php?descargar=/documentos/ficha.doc` . Como podeis ver en el código fuente, no hemos añadido ningún tipo de filtro, por lo que un usuario malintencionado podría poner en `?descargar=` otro archivo distinto, lo que conllevaría la descarga de otro archivo alojado en el servidor.

Es decir, si nosotros modificamos la URL y colocamos `http://SCD.com/index.php?descargar=/admin/admin.php`, obtendríamos como resultado la descarga del index del panel de administración... Con lo que podríamos buscar posibles vectores de ataque contra ese archivo (o contra cualquier otro que nos descargemos).

Pero de esta forma no se le saca el auténtico jugo de esta vulnerabilidad, puesto que puede ser dificultoso encontrar bugs dentro del source de las aplicaciones. El auténtico jugo se le saca cuando dentro del código fuente de lo que nos descargamos aparecen datos de la DB. En este caso, nos abría tocado el gordo.

Retomemando el ejemplo anterior, imaginémonos que dentro del código fuente de `admin.php` encontramos una línea similar a esta:

```
include("/admin/includes/database.inc");
```

Inclusiones de este tipo dentro del código fuente de un `.php` suele ir asociado a la asignación de los datos necesarios para conectarse a la Base de Datos y que la aplicación trabaje sobre ella. Entonces siempre deberemos de descargarnos esta clase de archivos, puesto que es muy alta la probabilidad de encontrar datos como `DB_USER`, `DB_PASS` y demás.

El segundo tipo del que os iba a hablar es del que aparece en el siguiente post de una vulnerabilidad descubierta por SetH ([Post aquí](#)). Se produce cuando permiten la inclusión de un archivo (el típico LFI), pero filtran el código para evitar que se ejecute.

A diferencia del tipo anteriormente comentado, ahora para ver el código fuente no recurriremos a la descarga del archivo, si no que al hacer la inclusión pueden ocurrir dos cosas:

- a) Veamos el código fuente del archivo
- b) Sólo veamos pequeñas partes del código y algunos tags HTML ejecutados

De encontrarnos en el segundo caso, para ver el código fuente únicamente tendremos que darle a "Ver Código fuente" dentro del navegador, en FireFox existe el método abreviado de `Ctrl + U`. Ahí veremos perfectamente todo el código fuente :) (el PHP tambien se ve de esta forma)