Unidad 8

Seguridad

Contacto: consultas@elearning-total.com

Web: www.elearning-total.com

Seguridad

Si tenemos la posibilidad de poder modificar nuestro Servidor Web a través del archivo php.ini es importante tener en cuenta:

No mostrar los errores

- display_errors = Off;
- display_startup_errors = Off;

No desactivar register_globals

Viene desactivado por defecto a partir de PHP5. Es importante no activarlo.

Deshabilitar funciones peligrosas que no necesitemos

Esta característica puede ser configurada únicamente en el archivo php.ini, no puede ser modificada en tiempo de ejecución.

A través de la línea **disable_functions =** se pueden colocar las funciones que pueden exponer nuestro sistema.

Por ejemplo, deshabilitar aquellas que permitan modificar funciones del sistema (exec, shell_exec, passthru) que permitan correr scripts sin límite (set_time_limit) entre otras.

No permitir la apertura e inclusión de archivos remotos

No conviene que se incluyan o abran archivos por URL.

Para configurar esto debemos indicar en el php.ini:

```
allow_url_fopen = Off; //por defecto en On
allow_url_iclude = Off; //por defecto en Off
```

En la práctica aplicando esta configuración no podríamos realizar:

```
fopen('http://www.sitioexterno.com/datos.txt', 'r');
```

Si podemos abrir archivos almacenados localmente. De este modo evitamos que se inserte código malicioso en algún script PHP

Limitar los archivos y directorios que se pueden abrir

Es posible limitar el directorio del cual se puede tomar archivos con la función fopen y otras funciones de lectura y escritura.

Para ello debemos indicar: open_basedir = /usr/local/apache2/htdocs/datos

De esta manera solo se tomarán archivos y directorios en la ruta indicada.

Aplicar límites al tiempo de ejecución de nuestros scripts

Uso de memoria y tamaño de datos a enviar y recibir por POST o subida de archivos

```
memory_limit = 16M; Máximo de memoria utilizado en MB
max_input_time = 60; Tiempo máximo de ejecución en envío de datos
max_execution_time = 30; Tiempo máximo de ejecución en segundos
upload_max_filesize = 4M; Tamaño máximo en MB de archivos a subir
post_max_size = 8M; Tamaño máximo en MB de los requests por POST
```

No dejar un archivo con phpinfo() en nuestro servidor

Es muy común en las etapas de desarrollo subir algún archivo con esta función para conocer la configuración básica del sistema.

Por eso es fundamental recordar la eliminación de este archivo antes de dejar nuestro sitio en producción.

No permitir la apertura de código PHP con tags cortos

Lo ideal es que todos los scripts php se indiquen con su marca extendida y standard <?php ?>

La versión reducida solo funcionará si hemos activado la directiva:

```
short_open_tag = On //por defecto viene en Off
```

Magic_quotes_gpc=Off

Esta es la configuración por defecto.

Cuando están habilitadas, todos los caracteres ' (comillas simples), " (comillas dobles), (barras invertidas) y NULL son "escapados" automáticamente con una barra.

Este comportamiento es equivalente al de la función addslashes().

La directiva magic_quotes_gpc sólo puede deshabilitarse en el sistema, y no en tiempo de ejecución. En otras palabras, no se puede utilizar ini_set().

Si no es posible acceder a la configuración del servidor, para cambiar alguna variable se puede usar un archivo .htaccess.

Por ejemplo: php_flag magic_quotes_gpc Off

Mas información sobre Seguridad en: http://www.php.net/manual/es/security.php

XSS

Los ataques más comunes a los sitios web suelen darse por los puntos de ingreso de datos delos usuarios, y mucho más cuando estos datos almacenados son utilizados después para mostrarse en pantalla.

Un ataque XSS consiste en ingresar datos en forma maliciosa para que al ser mostrados generen el ataque.

Por ejemplo, si un sitio web de noticias aceptara comentarios, un atacante podría escribir:

```
<script>window.location = http://www.google.com.ar; </script>
```

Este código redireccionará a todos los visitantes a otro sitio web.

O podríamos hacer un script que genere infinita cantidad de mensajes emergentes consiguiendo que el navegador se cuelgue:

```
<script>
while (true) { alert('xss'); }
</script>
```

¿Como podemos prevenir esto?

Lo que debemos hacer es filtrar todo aquello que se ingresa. Para ello creamos una clase a la que le enviamos los datos y nos los devuelve seguros.

SQL Injection

Atacar un sitio web mediante ataques SQL consiste en manipular el ingreso de datos de los usuarios cuyo destino sea la formación de consultas SQL. Por ejemplo, un login mediante usuario y contraseña, especialmente, cuando se envían datos por URL o vía un formulario.

Mas información sobre seguridad:

- http://phpsec.org/
- http://www.php.net/manual/es/security.php
- http://php.net/manual/es/security.database.s ql-injection.php

OWASP

El Proyecto abierto de seguridad en aplicaciones Web u Open Web Applicattion Security Project (OWASP por sus siglas en inglés) es una comunidad abierta y libre de nivel mundial enfocada en mejorar la seguridad en las aplicaciones de software. Su misión es hacer que las organizaciones, en base a los riesgos de seguridad de las aplicaciones, puedan tomar decisiones para contribuir en la seguridad de las mismas. Todo el mundo es libre e participar en OWASP bajo una licencia de software libre y gratuito.

La fundación OWASP es una organización sin fines de lucro que asegura disponibilidad y apoyo a nuestro trabajo.

Pueden encontrar más información en:
https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#tab=Main