PHP – CONSEJOS DE SEGURIDAD.

Por José Carlos Cruz Parra AKA Internia, septiembre de 2004.

- Si usamos encriptación para almacenar las contraseñas, enviar la contraseña ya cifrada desde el formulario de login.
- Validar los datos recogidos por un formulario tanto del lado del navegador (por una navegación más cómoda para el usuario) como del servidor (por seguridad).
- Deshabilitar session.use_trans_sid (habilitado hace que se añada a todos los enlaces relativos el identificador de la sesión):
  `ini_set("session_use_trans_sid","0");`
- Habilitar la propagación del identificador de sesión por cookie:
  `ini_set("session_use_cookies","1");`
- Habilitar la propagación del identificador de sesión sólo por cookie (deshabilita session_use_trans_sid):
  `ini_set("session_use_only_cookies","1");`
- El nombre de la variable que almacena el identificador de sesión es igual al nombre de la sesión, y también es el nombre de la cookie que contiene dicho identificador. Por defecto, este nombre es PHPSESSID. Se puede establecer otro (recomendado) con session_name().
- Eliminar el registro en el save_path (o en la BD, si se ha creado el método para ello) de la sesión serializada, cuando ésta se da por finalizada: `session_destroy();`
- Eliminar las variables de sesión actuales: `$_SESSION = array(); //Vacía $_SESSION`.
- Hacer expirar las sesiones a un tiempo razonable.
- Conservar información de la sesión que ayude a detectar anomalías. Por ejemplo, almacenar la dirección IP del usuario que inició la sesión y en el momento en que esta cambie sin que se haya cerrado la sesión, darla por finalizada (es muy inusual que un usuario rote su IP).
- Si usamos un servidor compartido, establecer el save_path dentro de nuestro propio árbol de directorios, de manera que otros usuarios no puedan acceder a él.
- Si usamos BD, es recomendable almacenar los datos de las sesiones en una tabla, mejor que como un archivo en el directorio save_path. En este caso, el valor de save_path será el nombre de la BD, el nombre de la sesión será el nombre de esa tabla, y habrá que escribir nuestros propios métodos para manejar las sesiones (open, close, read, write, destroy, gc).
- Para evitar que una página sea generada sobre una conexión no encriptada:

```
if($_SERVER["HTTPS"] != "on")
{
   die("Must be a secure connection.");
}
```

- Para que los datos recogidos por un formulario se envíen usando la conexión segura (si la hay), la URL de la propiedad action del formulario ha de indicarlo: https://URL
- Deshabilitar register_globals: `ini_set("register_globals","Off");`
- Inicializar siempre las variables.
- Personalizar variables_order:
  `ini_set("variables_order","ES");`
  //en vez de "EGPCS" (Environment, Get, Post, Cookies, Server).
- Desactivar el visionado de errores y advertencias (pueden das pistas a un hacker), almacenarlos en un archivo log:

```
ini_set("display_errors","Off");
ini_set("log_errors","On");
ini_set("error_log","ruta/php_errors.log");
```

- El lugar más seguro para almacenar información es una BD, no archivos.

- Almacenar las librerías de código y los datos (configuración) fuera del directorio raiz html.
- Las aplicaciones web nunca deberían conectarse a la BD como el usuario dueño de la misma, ni como super-usuario.
- No implementar toda la lógica del asunto en la aplicación web, sino también en el esquma de la BD, usando vistas, disparadores o reglas.
- Si se puede, usar SSL ó SSH para encriptar la comunicación cliente-servidor.
- Para mayor seguridad se pueden encriptar los datos al de almacenarlos en la BD y desencriptarlos al recuperarlos. PHP ayuda en este sentido con diversas extensiones, como Mcrypt y Mhash.
- Si la representación original de ciertos datos no se necesita, se pueden utilizar resúmenes criptográficos para almacenarlos (la representación original se pierde). Ejemplo clásico: las contraseñas con md5.
- Guardar siempre encriptadas las contraseñas.
- Validar los datos recogidos por un formulario o enviados por el usuario de alguna manera. Revisar si son del tipo apropiado:
  - Si la aplicación espera una entrada numérica, verificar la información con is_numeric(), o modificar silenciosamente su tipo con settype(), o utilizar su representación numérica dada por sprintf("%d",$var).
  - Cada entrada del usuario no numérica que vaya a ser pasada a la BD ha de ser ubicada entre comillas con addslashes() ó addcslashes().
- No dar a conocer a nadie información específica sobre la BD, especialmente sobre su esquema.
- Colocar siempre un caso default en cada bloque switch.

José Carlos Cruz Parra

Analista – Programador

www.programadorphpfreelance.com