Elegir El Mejor Handler De PHP

Aunque este articulo caiga como anillo al dedo para aquellos administradores de Servidores VPS y Servidores Dedicados, puede ser una gran y perfecta guía para aquellos programadores que deseen saber cual es el mejor **controlador ó handler de PHP** que deben de elegir. En este artículo vamos a mostrar las diferentes opciones de **controladores de PHP** que tienes que elegir en su VPS ó servidor dedicado, para manejar de forma optima los recursos de la maquina.

¿Que Es Un Controlador Ó Handler de PHP?

Un **controlador ó handler de PHP** define el método que Apache utiliza para comunicarse con los datos que se envían y reciben con los **Scripts de PHP**. Conocer las diferencias entre los controladores de PHP puede ayudar a asegurarse de que está utilizando el mejor para el entorno de desarrollo en PHP.

Elegir el mejor **manejador de PHP** para sus necesidades específicas, puede marcar la diferencia en tu proyecto,

desarrollo ó el manejo de tu servidor ;) Cada **handler de PHP** cuentas con sus características, por lo que es bastante recomendable que las conozcas y decidas cual controlador usar para administrar los procesos de PHP.

Los Handler de PHP Disponibles

Como se ha mencionado, existen diversos controladores ó **Handler de PHP**, a continuación los conocerás, y ademas podrás saber cual puedes usar dependiendo de sus características:

Handler suPHP (Single user PHP)

Con el **manejador suPHP** seleccionado junto con el uso de la opción suEXEC, todos los **scripts PHP** se ejecutan como el usuario que posee la sesión en la consola de comandos, en lugar de ejecutar como usuario «**nobody**» de Apache. Por cada solicitud de PHP, se podrá analizar cual es el usuario que mas esta consumiendo recursos, ya que se podría analizar de forma individual la propiedad del script que se esta ejecutando.

La principal ventaja de utilizar el **manejador suPHP** es que aísla un usuario en el servidor de los demás usuarios. Así que si una cuenta se ha visto comprometida debido a un exploit en

uno de sus scripts PHP, el atacante sólo sería capaz de ver ó modificar los ficheros que pertenecen al usuario afectado. Es también uno de los más fáciles de usar con un CMS (Content Management Systems) como WordPress o Joomla.

La principal desventaja de usar el **manejador suPHP** es que sus **sitios web con PHP** podrían ser más lentos, debido a la sobrecarga adicional de tener que ejecutar un **proceso PHP separado por cada solicitud**. Además, si el servidor recibe un gran volumen de solicitudes de PHP en un corto período de tiempo, esto podría conducir a una carga pesada en el servidor. Finalmente no serás capaz de cambiar las directivas PHP directamente desde un **archivo htaccess**, en su lugar es necesario realizar estos cambios en el **archivo php.ini**.

DSO (Dynamic Shared Object)

El manejador DSO, es el **handler de PHP** más antiguo que ejecuta PHP como módulo de Apache. Todos los **scripts PHP** se ejecutan en Apache como el **usuario «nobody»**, lo que permite una ejecución más rápida, pero pueden volverse un dolor de cabeza a la hora de c**onfigurar los permisos**.

La principal ventaja de utilizar el **controlador DSO,** es que no tiene muchos gastos de recursos, por lo que es rápido fuera de la caja. También puede utilizar el **almacenamiento en caché de PHP,** junto con DSO para **acelerar las peticiones PHP**. Además, puede definir directivas de PHP directamente a través de los archivos htaccess para controlar algunas funciones de PHP.

La principal desventaja de utilizar el **controlador de DSO** es que todo **script de PHP** tendría que ser de propiedad y ejecutado por el usuario «nobody». Esto significa que las ejecuciones de PHP no pueden ser fácilmente rastreadas por un

usuario. Además, cuando se utiliza un (Content Management System) CMS como WordPress ó Joomla que requiere la capacidad de estas aplicaciones para escribir o modificar archivos, los permisos adecuados se deben establecer de antemano para permitir que esto suceda. Cada vez que un nuevo archivo es creado por un script PHP, será propiedad del usuario «nobody» y puede llevar a más **problemas de permisos**.

Otro problema importante del **handler DSO** es la seguridad. Debido a que todos los usuarios en el servidor están ejecutando sus scripts PHP por el mismo usuario, si se explota uno de sus sitios debido a una falla en uno de sus **scripts de PHP**, el atacante podría llegar a ver o modificar archivos fuera del directorio de dicho usuario.

FCGI (FastCGI)

Utilizando el **handler FastCGI de PHP** es una manera más rápida para atender las solicitudes de PHP. **FastCGI** es bueno para reducir el uso de CPU mediante el aprovechamiento de la memoria RAM disponible de su servidor para mantener los scripts PHP en la memoria en lugar de tener que poner en marcha un proceso PHP separado para cada **solicitud de PHP**.

La principal ventaja de utilizar el manejador FastCGI es que se puede utilizar suEXEC al igual que con suPHP para permitir la ejecución de los scripts de PHP ser ejecutadas por el usuario que ha ejecutado el Script. **FastCGI** no requiere solo una ejecución de procesos PHP por solicitud como suPHP lo requiere, por lo que puede ser mucho más rápido y reduce el uso de la CPU.

La principal **desventaja del controlador FastCGI** es que puede consumir bastante memoria RAM. Esto es porque **FastCGI mantiene las sesiones de PHP abiertas** en background en la memoria, para un acceso más rápido. Sin duda, este es el manejador indicado para un servidor dedicado que cuente con bastante capacidad de

memoria RAM.

CGI (Common Gateway Interface)

Al igual que el **handler suPHP y FastCGI** , el manejador CGI puede utilizar suEXEC de modo que las ejecuciones de PHP son dirigidos por el propietario del archivo de un script PHP en lugar del usuario «nobody» de Apache. La principal ventaja de utilizar el **controlador CGI** es que es bastante configurable y admite el uso de suEXEC para hacer que los permisos no se vuelvan un dolor de cabeza.

La principal desventaja del controlador CGI es que es el **handler de PHP** más lento. El controlador CGI normalmente no se utiliza mucho. Puedes elegir el handler CGI como su controlador de PHP, si por alguna razón suPHP, DSO, ó FastCGI no estaba disponible en el servidor.

Finalmente

Deberías tener una mayor comprensión sobre los diferentes manejadores de PHP disponibles para ser usados en el servidor. Elegir qué handler de PHP es el mejor para su entorno de PHP, puede llegar a ser una difícil decisión, y por lo general no es recomendable cambiarlos si no estás teniendo ningún problema de rendimiento ó de seguridad con la configuración por defecto.