

Cambios especificados en el php.ini disable_functions por dominio (Directadmin example)

Cambiar individualmente en un dominio disable_functions

Un escenario muy común en sistemas y mas en estos cuando se usa **Directadmin**, es la salida hacia adelante obviando los problemas que puede causar nuestra decisión.

Un caso común es el quitar las protecciones de forma global a través de el proceso general del servidor, en lugar de buscar una solución individual, para el caso concreto.

En el caso de Directadmin, si tenemos la opción [secure_php](#) esta añadirá a la configuración del php del servidor

```
disable_functions =
exec,system,passthru,shell_exec,proc_close,proc_open,dl,popen,show_source,posix_kill,posix_mkf
ifo,posix_getpwnuid,posix_setpgid,posix_setsid,posix_setuid,posix_setgid,posix_seteuid,posix_se
tegid,posix_uname
mysqli.allow_local_infile = Off
expose_php = Off
register_globals = Off
```

Existe el camino fácil, que es ir quitando las funciones que molestan a nuestros clientes, [How to customize the disable functions list](#) y existe otro camino, que es deshabilitar al que nuestro cliente necesita, si esto es totalmente necesario, ya sea por un tiempo limitado, o de forma definitiva.

Opciones de php.ini por dominio

[Opciones por dominio](#) o [How to use per-domain or per-path settings for php](#) nos ayuda a esto último ya que tratar de modificar `disable_functions` via `.user_ini` aunque este autorizado, no funcionará para modificar esa clave.

- Lo primero es buscar el path de nuestra instalación para nuestro Directadmin y nuestra distro.

En mi caso esta en `/usr/local/phpXX/lib/php.conf.d`

Así que, siguiendo el manual, introducido el path y el dominio o subdominio del que quiero tal corrección, el nos lo dará.

En nuestro caso el cliente tenía algunos despliegues con Laravel que necesitaban `shell_exec` por lo que lo eliminamos del fichero **ini** personalizado,

```
/usr/local/php83/lib/php.conf.d/30-subdomain.full.tld.ini
```

```
disable_functions =  
exec,system,passthru,proc_close,proc_open,dl,popen,show_source,posix_kill,posix_mkfifo,posix_g  
etpwuid,posix_setpgid,posix_setsid,posix_setuid,posix_setgid,posix_seteuid,posix_setegid,posix  
_uname
```

Después simplemente tenemos que hacer un rebuild de la función de seguridad.

```
da build secure_php
```

jit opcache.jit_buffer_size

Esta opción como muchas otras de php, tiene esta cuestión activa:

```
< opcache.jit_buffer_size es PHP_INI_SYSTEM, lo que significa que **solo puede definirse en  
php.ini o php-fpm.conf, no en .user.ini ni .htaccess.
```

Esto fuerza a usar el método de cambio

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Revision #5

Created 2024-12-16 17:41:28 UTC by Abkrim

Updated 2025-10-24 08:46:08 UTC by Abkrim