WHMCS: el actualizador falla con «fatal error while fetching» en DirectAdmin (proc_close)

Actualizar WHMCS desde su propio actualizador (*Utilities > Update WHMCS*) en un servidor DirectAdmin puede fallar con un mensaje que no dice la verdad:

```
Update Failed
A fatal error occurred while fetching the update.
Please check to make sure your memory limits and execution time are set to the recommended values.
```

El mensaje apunta a la memoria y al tiempo de ejecución, pero en DirectAdmin la causa habitual es otra: una función de PHP desactivada por seguridad. Antes de subir `memory_limit` o `max_execution_time`, mira el registro del actualizador.

Síntoma y causa

El actualizador de WHMCS 8 usa Composer y Symfony Console por debajo, y estos necesitan las funciones `proc_open` y `proc_close`. DirectAdmin, con la opción `secure_php=yes` de CustomBuild, desactiva una lista de funciones en el `php.ini` de cada versión de PHP, y `proc_close` (en la lista por defecto, también `proc_open`) está en ella.

El detalle real queda en la tabla `tblupdatelog` de la base de datos de WHMCS:

```
SELECT created_at, level, LEFT(message, 200)
FROM tblupdatelog
WHERE level >= 300
ORDER BY id DESC
LIMIT 10;
```

Si aparece esto, es este caso:

```
Error: Call to undefined function Symfony\Component\Console\proc_close()
in .../vendor/symfony/console/Terminal.php
```

Los requisitos mínimos de WHMCS (versión de PHP, ionCube, MySQL o MariaDB) no tienen que ver con este fallo: el servidor puede cumplirlos de sobra y fallar igual.

Lo que no funciona: cambiarlo en el pool de PHP-FPM

La idea natural es reactivar la función solo para la cuenta, con un `php_admin_value[disable_functions]` en el pool de PHP-FPM del usuario (en DirectAdmin, el fichero `php-fpmXX.conf.custom2` del usuario). **No sirve.** PHP aplica el `disable_functions` del `php.ini` al arrancar, y un pool puede añadir funciones a la lista pero no volver a habilitar una que el `php.ini` ya desactivó. `ini_get('disable_functions')` mostrará la lista nueva y aun así `function_exists('proc_close')` seguirá devolviendo `false`.

La solución: el fichero custom de CustomBuild

En DirectAdmin esto se hace con CustomBuild. La lista de funciones desactivadas se personaliza con el fichero `custom/php_disable_functions`, que **sustituye entera** a la lista por defecto: lo que no figure en él queda permitido.

1. Comprueba si el fichero ya existe y haz copia:

```
cd /usr/local/directadmin/custombuild
ls -l custom/php_disable_functions
cp -a custom/php_disable_functions custom/php_disable_functions.bak-$(date +%Y%m%d%H%M%S)
```

2. Escribe la lista sin `proc_close` ni `proc_open`, en una sola línea separada por comas:

```
echo
"exec,system,passthru,shell_exec,dl,popen,show_source,posix_kill,posix_mkfifo,posix_getpwuid,p
osix_setpgid,posix_setsid,posix_setuid,posix_setgid,posix_seteuid,posix_setegid,posix_uname" >
```

```
custom/php_disable_functions
```

3. Aplícalo. CustomBuild reescribe el `php.ini` de todas las versiones de PHP instaladas y reinicia sus php-fpm:

```
./build set secure_php yes
./build secure_php
```

4. Comprueba cada versión. Un `0` significa que `proc_close` ya no está desactivada en ella:

```
for v in /usr/local/php*/lib/php.ini; do echo "$v"; grep '^disable_functions' "$v" | grep -c
proc_close; done
```

Comprobarlo como lo ve la web, no la línea de comandos

El PHP de la línea de comandos y el de PHP-FPM pueden leer configuraciones distintas. La prueba que vale es por HTTP, con un fichero temporal de nombre aleatorio en el sitio que se borra en el acto (ejecútalo como el usuario de la cuenta, desde su directorio del dominio):

```
N=t$(openssl rand -hex 8).php
echo '<?php echo
json_encode(["proc_close"=>function_exists("proc_close"),"proc_open"=>function_exists("proc_op
en"),"exec"=>function_exists("exec")]);' > public_html/$N
curl -s https://example.com/$N; rm -f public_html/$N
```

Lo esperado es `{"proc_close":true,"proc_open":true,"exec":false}`: las dos funciones que necesita el actualizador disponibles y el resto de la lista sigue bloqueada.

Antes de relanzar la actualización

- **Copia propia:** haz tu propia copia del código y un volcado de la base de datos antes de actualizar. La copia que hace el actualizador es lenta en bases de datos grandes y bloquea tablas mientras la hace.
- **Actualización marcada como en curso:** si un intento anterior se cortó a medias, WHMCS puede haberla dejado marcada. Compruébalo y, si vale `1`, ponlo a `0`:

```
SELECT value FROM tblconfiguration WHERE setting = 'AutoUpdateInProgress';
UPDATE tblconfiguration SET value = '0' WHERE setting = 'AutoUpdateInProgress';
```

Después de actualizar

- **Modo mantenimiento:** mientras está activo, el cron de WHMCS no registra ejecuciones y el panel avisa de que el cron no corre («Cron Invocation Frequency»). Al desactivarlo, el aviso desaparece en la siguiente ejecución.
- **Tareas que hayas suspendido:** si para la ventana de mantenimiento pausaste algo en el cron (por ejemplo, las campañas de correo con `cron.php skip --EmailCampaign`), vuelve a dejar la línea como estaba.
- **Módulos retirados:** el actualizador borra módulos antiguos que WHMCS ya no mantiene (pasarelas como CashU o inpay, la notificación de HipChat). Lo deja anotado en el registro de actividad.

Al terminar: dejarlo como estaba

Tener `proc_close` desactivada es una medida de seguridad para todas las cuentas del servidor. Si solo la necesitas para la actualización, vuelve a la lista original en cuanto termine:

```
cd /usr/local/directadmin/custombuild
cp -a custom/php_disable_functions.bak-<fecha> custom/php_disable_functions
./build secure_php
```

Repite la comprobación por HTTP: ahora `proc_close` debe salir `false`. La próxima actualización de WHMCS volverá a necesitar este procedimiento.

Alcance del cambio

El fichero `custom/php_disable_functions` afecta a **todas las versiones de PHP y a todas las cuentas** del servidor: no hay forma de cambiarlo solo para una cuenta o una versión. Por eso conviene que la ventana con la función habilitada sea corta.

Referencia: https://help.poralix.com/articles/how-to-customize-php-disable-functioons-in-directadmin

---

Revision #1
Created 2026-11-27 04:38:21 UTC by Abkrim
Updated 2026-11-27 04:38:21 UTC by Abkrim