haak.
CYBERSECURITY CONSULTING

ALERTA TÉCNICA · PHP · 21 OCT 2026

RCE en PHP sin login y sin bug en tu aplicación.

CVE-2026-109618 y CVE-2026-109619, apodadas PHP2Shell. La falla está en el runtime de PHP: se explota sin autenticación y, en ciertas configuraciones, sin un bug propio de tu aplicación. Qué pasó, qué revisar hoy y qué hacer.

9.2
CVSS v4.0 · ambas críticas

PR:N
sin privilegios ni login

4
ramas con parche: 8.2 a 8.5

haak.
CYBERSECURITY CONSULTING

CRONOLOGÍA

Qué pasó y cuándo

17 OCT · REPORTADO

Intento registrado en un honeypot

Amazon (MadPot), desde una red de anonimización comercial. Cómo obtuvo el exploit: desconocido.

20 OCT

Pre-anuncio en php.net

Fija la publicación para el 21 de octubre a las 14:00 UTC. Ambas fallas con CVSS v4.0 9.2.

21 OCT · 02:00–07:34 UTC · REPORTADO

Actividad antes del parche

Amazon: primeras PoC a las 02:00. Cloudflare: reglas WAF a las 03:02. Patchstack: incidentes de hosting a las 07:34.

21 OCT · 14:00 UTC

Advisory y parches

PHP 8.5.12, 8.4.27, 8.3.36 y 8.2.35. Distribuciones y vendors LTS recibieron los parches por adelantado.

● relleno = fuente primaria (php.net / GHSA) ○ ámbar = reportado por Amazon, Cloudflare o Patchstack

haak.
CYBERSECURITY CONSULTING

DÓNDE ESTÁ LA FALLA · CASO mod_php

En el runtime; la ruta depende de la configuración

ANTES DE TU CÓDIGO

01 Request llega al worker

02 Parseo del body (multipart, upload progress)

109618: OOM en el parseo
↳ bailout con datos a medias
↳ el cleanup los trata como válidos

03 Inicialización de la request

04 Se ejecuta el script de tu app

109619 · falla independiente

05 Cleanup y shutdown

Estado que sobrevive
→ siguiente request
del mismo worker

Flujo conceptual en mod_php: 109618 corrompe el cleanup de la misma request; 109619, por rutas propias (sesión, backtrace de un fatal error), deja estado que hereda la siguiente request. En PHP-FPM, si falla el arranque, el proceso hijo termina.

haak.
CYBERSECURITY CONSULTING

LAS DOS FALLAS

Memoria a medias y estado que no muere

CVE-2026-109618 · RCE VIA OOM CORRUPTION

Corrupción por falta de memoria

Al agotarse la memoria, PHP abandona la request de golpe. Si eso ocurre con datos internos a medio inicializar, el cleanup puede tratarlos como válidos.

CVE-2026-109619 · RCE VIA STALE GLOBAL STATE

Estado global obsoleto

Estado global persistente sigue apuntando a memoria de una request anterior ya liberada (use after free). La siguiente request en el mismo worker puede consumirlo antes de la inicialización normal.

CWE: 416 (use after free), 755, 824, 1325 y 1419. Vector: CVSS:4.0/AV:N/AC:H/AT:P/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

haak.
CYBERSECURITY CONSULTING

WORKERS PERSISTENTES

Por qué parchar tu app no sirve

- **Un worker atiende muchas requests.**
  PHP-FPM, mod_php y similares reusan el mismo proceso. Lo que no se resetea puede heredarlo la request siguiente.

- **El código vulnerable es PHP mismo.**
  Actualizar WordPress, Laravel o tus dependencias no toca el runtime.

- **Instalar el paquete no reemplaza los procesos.**
  Los workers que ya corren siguen con el binario viejo en memoria. Reinícialos todos.

- **Sin login, sin interacción.**
  El vector es PR:N y UI:N. Requiere condiciones previas (AC:H, AT:P), pero no un bug tuyo.

haak.
CYBERSECURITY CONSULTING

¿TU STACK ESTÁ EN RIESGO?

Susceptibilidad por SAPI

SAPI	109618 (OOM)	109619 (ESTADO)
mod_php	RCE demostrado	RCE confirmado
PHP-FPM	Susceptible por análisis, sin exploit completo	RCE validado (ruta de sesión)
FrankenPHP	Probable	Sin ruta por defecto establecida; no es inmune
LiteSpeed	No evaluado en el advisory	Candidato condicional

php.net: en riesgo toda instalación que sirva requests web, incluidos PHP-FPM, mod_php, php-cgi, FrankenPHP y LiteSpeed.

En mod_php el RCE completo se validó sobre HTTP con los límites de input y memoria por defecto, en prefork normal con varios workers.

Demostrado o validado ≠ probable: no son equivalentes. Fuente: GHSA-w8hw-w9fx-j79w y GHSA-6wmf-4rv5-p3rp.

haak.
CYBERSECURITY CONSULTING

VERSIONES CORREGIDAS

¿Estás en una versión con parche?

RAMA	VULNERABLE SI ES MENOR A	CORREGIDA EN
8.5	8.5.12	8.5.12
8.4	8.4.27	8.4.27
8.3	8.3.36	8.3.36
8.2	8.2.35	8.2.35

PHP 8.1 Y ANTERIORES

Sin versión upstream nueva

El equipo de PHP solo publicó backports de código fuente para 7.4, 8.0 y 8.1; la corrección depende de tu rama, proveedor y paquete. Si no puedes actualizar de inmediato o usas 8.1 o anterior, CISA pide sacar los sistemas PHP de línea hasta parchar.

Paquetes de distro (Debian, Ubuntu, RHEL): el número puede no cambiar. Verifica en el aviso de tu distro o en el changelog que el paquete incluya las dos CVE. PHP 8.2 llega a fin de vida el 31 dic 2026.

haak.
CYBERSECURITY CONSULTING

QUÉ REVISAR HOY · 1/2

La versión del proceso que sirve tráfico

⚠ SALIDA SIMULADA · REPO SURY (DEBIAN/UBUNTU)

```
faccio@srv-web-01: ~
$ php-fpm8.4 -v
PHP 8.4.26 (fpm-fcgi) (NTS)
# VULNERABLE (< 8.4.27, paquete upstream)
$ sudo apt update && sudo apt install --only-upgrade php8.4-fpm
php8.4-cli libapache2-mod-php8.4
$ sudo systemctl restart php8.4-fpm # mod_php: restart apache2
$ php-fpm8.4 -v
PHP 8.4.27 (fpm-fcgi) (NTS)
$ sudo needrestart -r l
No services need to be restarted.
# Paquete 8.4.27 en disco. needrestart ayuda a ver workers viejos, no
lo certifica
```

- **¿Paquetes oficiales de tu distro?**
  `-v` no cambia de número: confirma las CVE con `apt changelog php8.4-fpm` o el aviso de tu distro. En Debian, `needrestart` se instala aparte.
- **Inventario completo.**
  Contenedores (reconstruye la imagen), staging, paneles y appliances con su propio PHP.
- **¿Usas remirepo?**
  php.net avisó el 21 oct que no tendría el parche ese día: verifica su estado.

haak.
CYBERSECURITY CONSULTING

QUÉ REVISAR HOY · 2/2

Configuración: el CLI no es tu servidor

⚠ ILUSTRATIVO: SALIDA SIMULADA

faccio@srv-web-01: ~

```
$ php -i | grep -E 'upload_progress.enabled|memory_limit'
session.upload_progress.enabled ⇒ On ⇒ On
memory_limit ⇒ -1 ⇒ -1
# Ojo: es el php.ini del CLI (en Debian trae -1).
# El pool de PHP-FPM carga otro.
$ grep -E 'upload_progress.enabled|memory_limit'
/etc/php/8.4/fpm/php.ini
;session.upload_progress.enabled = On
memory_limit = 128M
# Comentada (;) = sigue activa: el default es On.
# Esto es la config en disco, no la efectiva.
```

Aplica la mitigación en `php.ini` o en la config del pool (`php_admin_flag`) y reinicia los workers. No la pases desde el frontend con `PHP_VALUE` / `PHP_ADMIN_VALUE` , y revisa que no lleguen valores que la reviertan. Configuración efectiva: un `ini_get()` temporal en ese pool, restringido por IP y borrado al terminar; `php-fpm8.4 -tt` solo muestra los overrides del pool.

haak.
CYBERSECURITY CONSULTING

QUÉ HACER

Defensa, en este orden

1 Actualizar PHP (todas las SAPIs) y reiniciar TODOS los workers

La única mitigación completa. Sin parche posible: fuera de línea (CISA).

2 Temporales en php.ini, pool o servidor

- `session.upload_progress.enabled = 0`
- `fatal_error_backtraces = 0` (solo PHP 8.5)
- `memory_limit = -1` + cgroup/contenedor, idealmente que mate solo al worker (con limitaciones): contingencia 109618
- Reinicia todos los workers después de cada cambio

3 Reducir exposición

- `post_max_size = 8M` (32M con uploads), `upload_max_filesize` menor
- Bloquear multipart en el borde si tu app no lo usa; límites de tamaño, partes, headers, anidamiento, también sin `Content-Length`
- `pm.max_requests = 1` : rota el worker en cada request; costo de rendimiento, no es reparación

Ninguna temporal sustituye el parche: bloquean rutas confirmadas, no todas las variantes.

haak.
CYBERSECURITY CONSULTING

LO QUE NO SIRVE

Seis falsas soluciones

- ✕ **Actualizar solo la aplicación.**
  La falla está en PHP, no en tu CMS ni en tu framework.

- ✕ **Instalar sin reiniciar workers.**
  Los procesos viejos siguen atendiendo tráfico.

- ✕ **`RLIMIT_AS` o `RLIMIT_DATA`.**
  Pueden provocar justo el fallo de asignación vulnerable.

- ✕ **`ini_set()` o `.user.ini` en FPM.**
  Llegan tarde: el parseo de la request ya ocurrió.

- ✕ **Ocultar errores con `@` o `display_errors`.**
  Silenciar diagnósticos no cierra la ruta.

- ✕ **Una sola medida temporal.**
  Cada una cubre rutas específicas; actualizar sigue siendo obligatorio.

haak.
CYBERSECURITY CONSULTING

DETECCIÓN

Qué buscar en tus logs

⚠ EJEMPLOS ILUSTRATIVOS DE INDICADORES Y ALERTAS

php-fpm / apache2 error.log · WAF · EDR/auditd

```
[alerta] PHP Fatal error: Allowed memory size of 134217728 bytes
exhausted
[alerta] WARNING: [pool web] child 2817 exited on signal 11 (SIGSEGV)
[alerta] AH00052: child pid 5678 exit signal Segmentation fault (11)
[alerta] WAF: POST multipart con PHP_SESSION_UPLOAD_PROGRESS en app
sin barra de progreso
[alerta] EDR/auditd: www-data ejecuta /bin/sh, curl o wget
```

- **Logs y alertas no son lo mismo.**
  Las tres primeras son mensajes de log; las dos últimas son reglas de WAF y telemetría de procesos (EDR/auditd), porque el body del POST no queda en access.log. Un log limpio no prueba que estés a salvo.

- **Intento no es compromiso.**
  Las primeras cuatro señales priorizan hosts. Un worker que lanza un shell o descarga herramientas (CISA pide monitorearlo) es prioridad alta: investígalo como posible compromiso. Ninguna señal aislada confirma ni descarta una intrusión.

haak.
CYBERSECURITY CONSULTING

LO QUE SABEMOS Y LO QUE NO

Confirmado vs. reportado

CONFIRMADO

php.net · GHSA

- Dos CVE críticas, CVSS v4.0 9.2
- RCE demostrado en mod_php
- 109619: ruta de sesión validada en PHP-FPM
- Actualizar es la única mitigación completa

ACTIVIDAD REPORTADA

Amazon · Cloudflare · Patchstack · CISA · al corte del 21 oct

- Amazon: intento el 17 oct, PoC desde el 21 oct; reporta intentos de explotación amplios en el entorno real
- Cloudflare: tráfico de ataque y payloads mutados para intentar evadir su WAF
- Patchstack: incidentes de hosting a las 07:34; a las 13:57, solo sondeos y aún sin evidencia de brechas
- CISA (21 oct): sin explotación observada al publicar

Hay intentos de explotación reportados, algunos anteriores a la divulgación. Son observaciones de fuentes y momentos distintos. Cuántos sistemas fueron comprometidos: no está cuantificado.

haak.
CYBERSECURITY CONSULTING

RESPUESTA

Si estuviste expuesto

1 **Contén**

Ante sospecha fundada, conserva logs y evidencia y evalúa aislar el host: parchar no expulsa a un atacante.

2 **Busca señales de compromiso**

Procesos hijos de PHP, shells, descargas inesperadas, archivos nuevos o modificados en el webroot; logs de Apache, PHP-FPM, proxy y sistema.

3 **Rota secretos**

Ante sospecha fundada de RCE o fuga de memoria: credenciales, tokens, llaves y sesiones que el proceso o el host pudieran leer.

4 **Recupera y valida**

Reconstruye desde una fuente confiable en vez de limpiar a mano: paquetes o imágenes corregidos, todos los workers reiniciados, runtime efectivo comprobado.

No encontrar indicadores no demuestra que no hubo compromiso. Ajusta la respuesta a la evidencia y al alcance.

haak.
CYBERSECURITY CONSULTING

FUENTES

Verifícalo tú

Advisory oficial · 21 oct 2026
php.net/archive/2026.php

CVE-2026-109618 · RCE via OOM corruption
github.com/php/php-src/security/advisories/GHSA-w8hw-w9fx-j79w

CVE-2026-109619 · RCE via stale global state
github.com/php/php-src/security/advisories/GHSA-6wmf-4rv5-p3rp

Actividad reportada (21 oct): AWS Security Blog, Cloudflare Blog, Patchstack y alerta de CISA. Links completos en la descripción del post.

Reporte: sagitzadik-wiz, tomash-areion, lcfr-hax, amyb-oai. Remediación: arnaud-lb, iluuu1994, bukka, ndossche.

¿Quieres saber si tu infraestructura está expuesta?

HaaK Cybersecurity Consulting: pentesting web, API, infraestructura y cloud. Escríbenos para una evaluación. Tú creces, nosotros te cuidamos.