

BOLETÍN DE SEGURIDAD

Alerta de Seguridad en PHP - 23 Octubre 2026

OCTUBRE 2026



El material incluido en este documento es propiedad exclusiva de SOFTSECURITY SAS. Este documento no puede ser divulgado de cualquier manera o a cualquier persona a excepción del destinatario y los empleados o representantes de la firma a quien va dirigido, quienes son los directamente responsables de la evaluación de su contenido. Este documento no puede ser utilizado en cualquier forma diferente al propósito con el que fue distribuido. Se prohíbe terminantemente cualquier uso, reproducción o transmisión no autorizada.

INTRODUCCIÓN

En un mundo cada vez más conectado, la ciberseguridad no es solo responsabilidad del área de TI: es un compromiso de todos. Este boletín tiene como objetivo mantenerte informado sobre las amenazas más recientes, las buenas prácticas de seguridad digital y las acciones clave que puedes tomar para proteger tanto tu información personal como los activos de las organizaciones.

En el siguiente informe se encontrará información sobre amenazas actuales, recomendaciones e indicadores de compromiso.

¿¿Qué se conoce sobre las vulnerabilidades críticas en PHP (CVE-2026-109618 y CVE-2026-109619)?

PHP ha publicado dos vulnerabilidades críticas en su núcleo que permiten a un atacante remoto, sin autenticación, ejecutar código con los privilegios del proceso web. Ambas tienen una puntuación CVSS v4.0 de 9,2 y fueron catalogadas por INCIBE-CERT con importancia 5 (crítica) en su aviso INCIBE-2026-762.

Se han reportado ataques contra servidores PHP antes de que se publicaran las actualizaciones. La actividad observada se refiere al conjunto de ambas vulnerabilidades, sin precisar cuál de los dos CVE se utilizó en cada caso. Al tratarse de uno de los lenguajes más usados en el desarrollo web, el alcance potencial es muy amplio.

El equipo de PHP liberó las versiones corregidas el 21 de octubre de 2026, y las instalaciones que procesan solicitudes web deben considerarse en riesgo, incluidas las que usan PHP-FPM, Apache mod_php, php-cgi, FrankenPHP y LiteSpeed. El riesgo es mayor en Apache mod_php.

Versiones afectadas

- PHP 8.2.x anteriores a 8.2.35
- PHP 8.3.x anteriores a 8.3.36
- PHP 8.4.x anteriores a 8.4.27
- PHP 8.5.x anteriores a 8.5.12

PHP 8.1 y las ramas anteriores ya no reciben actualizaciones regulares. Aun así, el proyecto publicó parches de código fuente para las ramas 7.4, 8.0 y 8.1.

Análisis de las vulnerabilidades

1. CVE-2026-109618: ejecución remota de código por corrupción durante un agotamiento de memoria (OOM)

En ciertas rutas de procesamiento, una asignación de memoria puede fallar cuando los datos de la solicitud aún no están completamente inicializados. PHP interrumpe el procesamiento, pero al cerrar la solicitud puede usar ese estado inválido, lo que provoca corrupción de memoria y permite ejecutar código de forma remota.

El vector más destacado son las solicitudes multipart por POST: muchas cabeceras en las partes, controles de archivo vacíos o nombres de campos anidados pueden disparar el consumo de memoria, aunque la solicitud parezca respetar los límites configurados. PHP validó la ejecución remota de código sin autenticación contra Apache mod_php, antes de que se ejecute el código de

Carrera 32 A BIS No. 30 - 53 • Bogotá • TELEFAX 269 87 31 • www.softsecuritycorp.com

la aplicación. El análisis del código también considera susceptible a PHP-FPM, aunque no se ha demostrado un exploit completo para esa ruta.

2. CVE-2026-109619: ejecución remota de código por uso de estado global obsoleto

Parte del estado interno de una solicitud puede permanecer en el proceso trabajador después de liberarse su memoria. Si una solicitud posterior llega al mismo trabajador y usa ese estado antes de que PHP lo reinicialice, se produce un uso después de liberar memoria (use-after-free), con posible divulgación de memoria y ejecución remota de código.

PHP confirma rutas de explotación en Apache mod_php y PHP-FPM. La explotación puede depender de un error fatal o de una operación de sesión, y de que el mismo trabajador procese solicitudes sucesivas. En PHP 8.5 también se demostró una ruta relacionada con las trazas de errores fatales.

Recomendaciones

De acuerdo con lo mencionado en el boletín compartido por INCIBE-CERT y de parte de Softsecurity se recomienda llevar a cabo las siguientes acciones:

- **Identificación y priorización:** Inventariar todos los sistemas que ejecutan PHP, incluidos servidores web, contenedores, imágenes base de Docker, entornos de desarrollo y pruebas, hosting compartido y aplicaciones de terceros que incluyen su propio intérprete. Priorizar los sistemas expuestos a Internet y los que usan Apache mod_php, que es donde PHP indica mayor riesgo. Reconstruir y volver a desplegar las imágenes de contenedor con la versión corregida, ya que actualizar el host no corrige lo que corre dentro de una imagen antigua.
- **Actualización:** actualizar de inmediato a la versión corregida de su rama (8.5.12, 8.4.27, 8.3.36 u 8.2.35). Si se usan paquetes de una distribución o de un proveedor de soporte LTS, verificar que incluyan las correcciones. Para PHP 7.4, 8.0 y 8.1 se pueden aplicar los parches retroportados o migrar a una rama mantenida. PHP advirtió que el repositorio voluntario rpms.remirepo.net no tendría actualizaciones el día de la publicación, por lo que recomienda usar los paquetes de la distribución o compilar desde el código fuente.
- **Controles perimetrales.** Aplicar reglas en el WAF o proxy inverso que limiten el tamaño de las solicitudes, el número de partes multipart y la profundidad de anidamiento, y que bloqueen solicitudes malformadas. Establecer límites de tasa por IP en los endpoints que aceptan formularios o cargas de archivos. Si es posible, restringir el acceso a paneles administrativos y entornos de pruebas por IP o VPN. Adicional instalar herramientas EDR/XDR en los servidores que contengan estas aplicaciones PHP o una solución SIEM que recolecte los datos de estos servidores y sea capaz de detectar este tipo de ataques.
- **Gestión continua:** Habilitar actualizaciones automáticas de seguridad en los servidores cuando el entorno lo permita. Planificar la migración de las ramas PHP sin soporte (7.4, 8.0 y 8.1), porque dependen de parches externos y quedarán expuestas ante futuras vulnerabilidades. Suscribirse a los avisos de seguridad de PHP, de la distribución y de los CMS o frameworks utilizados. Documentar y ensayar un procedimiento de parcheo urgente para este tipo de avisos críticos.
- **Detección de posible compromiso:** Como hubo ataques antes de la publicación de los parches, conviene asumir que un servidor expuesto y sin actualizar pudo haber sido comprometido. Revisar los registros del servidor web en busca de solicitudes POST multipart anómalas, con tamaños inusuales, muchas partes, cabeceras en exceso o nombres de campo muy anidados. Buscar caídas o reinicios repetidos de procesos de PHP-FPM o Apache (segfaults, errores de memoria), que pueden indicar intentos de explotación. Revisar archivos nuevos o modificados en los directorios web, especialmente web shells,

Carrera 32 A BIS No. 30 - 53 • Bogotá • TELEFAX 269 87 31 • www.softsecuritycorp.com

y comparar contra una copia limpia o el control de versiones. Vigilar procesos hijos inusuales lanzados por el usuario del servicio web (shells, curl, wget, herramientas de red) y conexiones salientes no habituales. Si hay indicios de compromiso, rotar credenciales, claves de API y secretos accesibles desde el servidor, y activar el procedimiento de respuesta a incidentes.

- **Reducción del impacto de una explotación:** Ejecutar cada sitio o aplicación con un usuario y un pool de PHP-FPM distintos, con los mínimos privilegios, para que un compromiso no se propague a otras aplicaciones. Restringir el acceso a archivos con `open_basedir` y montar el código de la aplicación en modo solo lectura cuando sea posible. Desactivar con `disable_functions` las funciones de ejecución de comandos que la aplicación no necesite (`exec`, `shell_exec`, `system`, `passthru`, `proc_open`). Filtrar el tráfico saliente del servidor web y segmentar la red, de modo que no pueda comunicarse libremente con otros sistemas internos ni con servidores externos. Mantener copias de seguridad fuera de línea y probar su restauración.

Referencias

INCIBE-CERT. (2026, octubre 21). [Actualización 21/10/2026] Múltiples vulnerabilidades en PHP. Recuperado de <https://www.incibe.es/incibe-cert/alerta-temprana/avisos/multiples-vulnerabilidades-en-php>

PHP Group. (2026, octubre 20). Security advisory: Critical remote code execution vulnerabilities, update now. Recuperado de <https://www.php.net/archive/2026.php#2026-10-20-1>