

COMERCIOS, PROVEEDORES E ISO

PHP2Shell: qué significan las nuevas vulnerabilidades de PHP para quien acepta pagos con tarjeta

Publicado el 22 de octubre de 2026

Dos vulnerabilidades críticas de PHP pueden permitir que un atacante sin autenticar ejecute código en un servidor web. Para un comercio supone un riesgo para la integridad del pago, las credenciales de pago y la disponibilidad. Para un proveedor, es un riesgo de cartera. Qué ha pasado, qué comprobar y qué preguntar a su hosting.

El 21 de octubre de 2026, el proyecto PHP publicó correcciones de emergencia para dos vulnerabilidades críticas del propio lenguaje, registradas como CVE-2026-109618 y CVE-2026-109619 y apodadas **PHP2Shell**. Ambas puntúan 9,2 en CVSS 4.0. Cualquiera de las dos puede permitir que un atacante sin autenticar ejecute código en un servidor web afectado, y los fallos están en PHP, no en una aplicación concreta construida sobre él.

Para los comercios, esto crea un riesgo para la integridad del pago, las credenciales de pago y la disponibilidad. Para los proveedores de pagos y los ISO, crea una posible exposición en toda su cartera de comercios. Esto es lo que ha ocurrido, lo que conviene comprobar y lo que hay que preguntar a su proveedor de alojamiento.

PHP sostiene una parte muy grande de la web, incluida la mayoría del software de tienda que usan los comercios pequeños y medianos. Por eso esta es una historia de pagos tanto como de seguridad.

Qué se ha encontrado

- **CVE-2026-109618** es un error de corrupción de memoria. Si a PHP se le acaba la memoria a mitad de la preparación de una petición, la limpieza posterior puede tratar como válidos datos a medio construir.
- **CVE-2026-109619** es un error de estado obsoleto, de tipo use-after-free. PHP puede conservar un estado interno que apunta a memoria de la petición que ya se ha liberado, y un procesamiento

posterior puede usarla. PHP señala como efectos la ejecución remota de código y la divulgación de memoria sensible.

- **Alcance.** El código afectado más antiguo se remonta a PHP 5.4. El proyecto PHP afirma que toda instalación que atiende peticiones web está en riesgo, ya use PHP-FPM, Apache con mod_php, CGI, FrankenPHP o LiteSpeed. PHP valora el riesgo como más alto en mod_php, donde el ataque demostrado puede llegar antes de que se ejecute código alguno del propio sitio. Eso describe el ataque que se ha mostrado, no afirma que las demás configuraciones sean seguras.
- **Versiones corregidas.** PHP 8.5.12, 8.4.27, 8.3.36 y 8.2.35. Las versiones 7.4 a 8.1 no tienen versión oficial, solo adaptaciones mínimas del código fuente. Las versiones 5.4 a 7.3 han llegado al final de su vida útil y el proyecto PHP no las corregirá en absoluto. Algunas distribuciones de Linux y proveedores de soporte a largo plazo sí trasladan las correcciones a ramas más antiguas, así que consulte el aviso de seguridad de su propio proveedor antes de dar por hecho que una instalación concreta está sin parchear.

En qué punto estamos

Los detalles se filtraron el 20 de octubre, un día antes de la publicación prevista, que a su vez se había adelantado desde el 22 de octubre. El equipo de inteligencia de amenazas de Amazon informó de un intento contra sus honeypots el 17 de octubre, cuatro días antes de la publicación, y de más desde entonces, al parecer de varios actores. Cloudflare y watchTowr informan de tráfico de ataque, y watchTowr dice haber reproducido las dos vulnerabilidades. Existe código de explotación público, aunque su fiabilidad no está verificada, y el proyecto PHP dijo en la publicación que no había identificado ningún exploit funcional. Todavía no está establecido si los intentos vistos hasta ahora tendrían éxito contra sistemas reales en producción.

Es una amenaza creíble y urgente. Los intentos de explotación y el código de prueba de concepto no demuestran, por sí solos, que se hayan robado datos de pago a comercios, y a 22 de octubre no tenemos constancia de ningún compromiso confirmado de datos de pago atribuible a PHP2Shell. Parchee ahora los sistemas afectados, investigue cualquier cosa sospechosa y conserve los registros pertinentes. Que no haya una brecha confirmada no es motivo para esperar.

Por qué importa para los pagos con tarjeta

Un comercio no necesita ejecutar código PHP propio para verse afectado. Basta con el software que hay debajo de la tienda, y las consecuencias llegan al lado del pago de cuatro maneras.

1. **El servidor que construye la página de pago.** Ejecutar código en ese servidor permite a un atacante cambiar lo que hace la página de pago. Inyectar un script que copia los datos de la tarjeta mientras el cliente los escribe (a menudo llamado skimming o ataque tipo Magecart) es la vía de siempre por la que salen datos de tarjeta robados de las tiendas en PHP.
2. **Credenciales y claves.** Las claves de API de la pasarela de pago, los secretos de webhook y los accesos de administrador suelen estar en la configuración de la tienda. Quien pueda ejecutar

código en el servidor puede leerlos.

3. **Fraude y abuso.** Las claves robadas pueden usarse para emitir reembolsos o hacer transacciones de prueba. Un sitio comprometido también puede alojar un segundo pago oculto, que para el proveedor se parece mucho al [transaction laundering](#).
4. **Caídas.** Incluso un ataque fallido puede hacer caer procesos de trabajo. Una tienda caída no recibe pedidos, y una recuperación que tarde días puede romper los ciclos de [facturación de suscripciones](#) y provocar reembolsos y disputas.

Los comercios que redirigen a la página de pago alojada del proveedor, o que incrustan los campos del propio proveedor, están mejor situados: el número de tarjeta nunca pasa por su servidor. Aun así, siguen expuestos en el administrador, los reembolsos y las claves de API descritos arriba.

Qué comprobar esta semana

- **Averigüe qué PHP ejecuta.** En su propio servidor, ejecute `php -v`. En un alojamiento gestionado o compartido, mire el panel de control o pregunte al proveedor. Los complementos y el software de tienda no lo informan, y una tienda puede quedarse en una versión antigua de PHP años después de haber actualizado su software.
- **Aplique la corrección del proveedor.** Para la mayoría de las distribuciones de Linux y de los alojamientos, eso significa la actualización de paquetes del proveedor, seguida de un reinicio de PHP para que el cambio surta efecto. No se fíe solo del número de versión de PHP, porque las distribuciones suelen trasladar la corrección sin cambiarlo.
- **Si usa PHP 7.3 o anterior, planifique el cambio.** El proyecto PHP no corregirá esas versiones. Pregunte a su proveedor de alojamiento o a su desarrollador si su proveedor traslada una corrección, y pida la fecha, no la intención.
- **Las mitigaciones son temporales.** El proyecto PHP documenta ajustes que reducen cada vulnerabilidad, y un cortafuegos de aplicaciones web o restricciones de red pueden disminuir la exposición mientras prepara una actualización. Ninguna es una corrección completa. Actualice a una versión corregida o aplique el parche de seguridad verificado de su distribución lo antes posible y, si un sistema expuesto a internet no puede parchearse con rapidez, considere restringir su exposición.
- **Cambie los secretos si no puede descartar un compromiso.** Claves de la pasarela de pago, secretos de webhook, contraseñas de administrador y credenciales de la base de datos. Busque nuevos usuarios administradores, archivos de pago modificados y scripts inesperados en la página de pago.
- **Mire el pago.** Compare los scripts que carga la página de pago con lo que espera. Los dominios de terceros desconocidos son la señal de alerta.

PCI DSS y su proveedor

Si está dentro del ámbito de PCI DSS, esto entra en sus responsabilidades existentes de gestión de vulnerabilidades. El requisito 6.3.1 exige identificar las nuevas vulnerabilidades y asignarles clasificaciones de riesgo. El requisito 6.3.3 exige que los parches críticos o de alta seguridad, según esa clasificación, se instalen en el plazo de un mes desde su publicación.

Eso es un máximo, no una recomendación de esperar. Dada la gravedad de PHP2Shell y los informes de actividad de explotación, los sistemas afectados expuestos a internet deben corregirse de inmediato. Cuando la página de pago está dentro del ámbito, también pueden ser pertinentes los requisitos sobre scripts en páginas de pago.

Un comercio que no puede demostrar que parcheó con rapidez, y no tiene constancia del motivo, está en una posición débil si algo sale mal. Una nota breve por escrito con la fecha en que se identificó el fallo, lo que se hizo y cuándo basta para empezar.

Avise pronto a su proveedor si tiene motivos para pensar que la tienda se vio comprometida. Los proveedores de pagos pueden tener obligaciones de notificación y escalado según las normas aplicables de las marcas de tarjeta, y un comercio que lo comunica el primer día recibe un trato muy distinto del de uno al que el proveedor descubre por una alerta de la marca o por una serie de informes de fraude.

Para proveedores e ISO

La exposición abarca toda su cartera y no se reparte de forma pareja. Los comercios con PHP al final de su vida útil, con alojamiento compartido barato y con software de tienda autogestionado son los primeros que conviene mirar. Pasos razonables son:

- Contactar con los comercios que usan software de tienda autoalojado, con un aviso breve y claro y la petición de la fecha en que aplicaron el parche.
- Vigilar las señales de un pago comprometido: un aumento repentino de pruebas de tarjetas, rechazos inesperados, reembolsos que no coinciden con pedidos o disputas que denuncian fraude en tarjetas usadas solo en ese comercio.
- Volver a ejecutar la supervisión de sitios web en los comercios de mayor volumen y buscar scripts nuevos o destinos de pago desconocidos.
- Recordar que la misma vulnerabilidad afecta a sus propias aplicaciones y a los sitios de sus socios y afiliados.

En el alta, una pregunta útil es qué plataforma y qué alojamiento usa el comercio, y quién es responsable de mantenerlos al día. La respuesta dice mucho sobre cómo se comportará una cuenta en una semana como esta.

Fuentes

- [Tenable: PHP2Shell \(CVE-2026-109618, CVE-2026-109619\) frequently asked questions](#)
- [Qualys: PHP announces critical security release for two vulnerabilities](#)
- [Cisco Talos: Two critical RCEs in PHP core](#)
- [CERT-EU: Critical vulnerabilities in PHP \(2026-018\)](#)
- [watchTowr: PHP remote code execution exploited](#)

Leer en línea: <https://payments-marketplace.com/es-ES/insights/php2shell-payments/>

Payments Marketplace Ltd

Registered in England & Wales No. 17424288

Garden Ctg, Dereham Road, Dereham, Norfolk, NR19 2PW, UK

connect@payments-marketplace.com · +44 1366 727140