Alerta Crítica

PHP · Ejecución remota de código sin autenticación

COLCERT AL – 20261021 - 123

TLP:CLEAR

1. Resumen Ejecutivo

El equipo de seguridad de PHP publicó el 21 de octubre de 2026 dos vulnerabilidades críticas (CVSS v4.0 9.2) que permiten ejecutar código de forma remota sin autenticación: *CVE-2026-109618 y CVE-2026-109619.*

Afectan a toda instalación de PHP que atienda peticiones web: PHP-FPM, Apache con mod_php, php-cgi, FrankenPHP y LiteSpeed. El riesgo es mayor en mod_php, donde existe un exploit demostrado que se activa con una petición no autenticada antes de que se ejecute el código de la aplicación.

Acción requerida: actualizar de inmediato a PHP 8.5.12, 8.4.27, 8.3.36 o 8.2.35. Las distribuciones Linux y los proveedores LTS recibieron los parches por adelantado, por lo que se recomienda verificar los paquetes publicados por cada proveedor.

No existe una mitigación completa. Las medidas de la sección 5 reducen la exposición a rutas de ataque específicas mientras se aplica la actualización, pero no hacen segura una instalación vulnerable.

PHP 8.1 y anteriores no reciben actualizaciones oficiales. El equipo de PHP publicó retroportes mínimos para 7.4, 8.0 y 8.1 dirigidos a quienes compilan desde el código fuente; la alternativa recomendada es migrar a una rama con soporte.

NIVEL DE RIESGO

ALTO

2. Vulnerabilidades y riesgo

Ambas fallas se explotan con una petición HTTP sin autenticar. Un atacante que logre ejecutar código en el proceso PHP obtiene los privilegios del servidor web y acceso a las bases de datos, credenciales y archivos de la aplicación.

El equipo de PHP anunció el parche el 20 de octubre y lo publicó el 21 de octubre de 2026 a las 9:00 a. m. (hora de Colombia). Con los detalles técnicos ya públicos, **el tiempo que una entidad tarde en actualizar define su nivel de exposición.**

¿Cómo se materializa el riesgo?

Ruta de ataque

Petición sin autenticar → **Fallo de memoria** → **Ejecución de código** → **Compromiso**

Ruta de respuesta

Inventario de PHP → **Actualización** → **Mitigación temporal** → **Monitoreo**

Vulnerabilidades publicadas

Vulnerabilidad	Descripción e impacto
CVE-2026-109618 RCE por corrupción OOM	Tras un fallo de memoria (OOM), datos de la petición a medio inicializar se tratan como válidos al limpiarla, lo que permite ejecutar código. CVSS v4.0 9.2.
CVE-2026-109619 RCE por estado global obsoleto	Estado interno de una petición sobrevive a la liberación de su memoria y otra petición del mismo worker lo reutiliza. CVSS v4.0 9.2.
Entornos afectados	Toda instalación PHP que atienda peticiones web: PHP-FPM, Apache con mod_php, php-cgi, FrankenPHP y LiteSpeed.
Mayor riesgo: mod_php	Exploit demostrado: una petición sin autenticar desencadena el ataque antes de que se ejecute cualquier código de la aplicación.

3. Exposición en Colombia

Shodan identifica 8.699 servicios con PHP expuestos a Internet en Colombia.

El 88 % está en redes de proveedores de internet y telecomunicaciones, donde el servidor puede pertenecer a cualquier entidad o empresa cliente.

Servicios con PHP expuestos por tipo de red, Colombia

¿Cómo leer estos datos?

- Las cifras se presentan agregadas por tipo de red. ColCERT no publica nombres de entidades, direcciones IP, ni dominios.
- La organización (ASN) que reporta Shodan es la red que asigna la dirección IP, no necesariamente la entidad responsable del servidor.
- Shodan infiere la versión a partir de la información que publica cada servicio; el resultado indica exposición potencial y debe confirmarse en cada servidor.
- Se recomienda que cada entidad revise sus propios activos con los comandos de la sección 4 y actualice los que estén en una versión vulnerable.

Fuente: Shodan, consulta del 21 de octubre de 2026, servicios agrupados por organización (ASN). Clasificación por tipo de red: ColCERT. «Otras redes» agrupa las organizaciones fuera de las 100 con más resultados.

4. Versiones afectadas y corregidas
Versión corregida por rama

Toda versión anterior a la corregida de su rama es vulnerable. Se recomienda identificar la rama en uso en cada servidor y aplicar la versión de la segunda columna.

Rama de PHP	Versión corregida	Fin del soporte de seguridad
8.5	8.5.12	31 de diciembre de 2029
8.4	8.4.27	31 de diciembre de 2028
8.3	8.3.36	31 de diciembre de 2027
8.2	8.2.35	31 de diciembre de 2026
8.1 y anteriores	Sin versión oficial	Fin de vida (EOL)

PHP 8.2 deja de recibir correcciones el 31 de diciembre de 2026. Si usa rpms.remirepo.net, cambie a paquetes de la distribución o compile desde el código fuente: ese repositorio no tendrá la actualización hoy.

¿Cómo identificar la versión en uso?

```
# Versión de PHP en línea de comandos y en PHP-FPM
php -v
php-fpm8.4 -v                       # ajustar a la versión instalada

# Versión expuesta por el servidor (si expose_php está activo)
curl -sI https://entidad.gov.co | grep -i "x-powered-by"

# Apache: confirmar si usa mod_php (mayor riesgo)
apachectl -M | grep -i php

# Mitigaciones temporales efectivas (CLI; en FPM revisar también el pool)
php -i | grep -E "memory_limit|post_max_size|session.upload_progress.enabled"

# Paquetes instalados (Debian/Ubuntu y RHEL)
dpkg -l | grep php
rpm -qa | grep php
```

5. Mitigaciones temporales (si no puede actualizar de inmediato)

```
; php.ini: debe regir antes de procesar la petición
; No usar ini_set() ni .user.ini

; CVE-2026-109618 · RCE por corrupción OOM
memory_limit = -1             ; verificar que ningún pool o directorio lo redefina
post_max_size = 8M            ; 32M si la aplicación recibe archivos
upload_max_filesize = 6M      ; siempre menor que post_max_size

; CVE-2026-109619 · RCE por estado global obsoleto
session.upload_progress.enabled = 0
fatal_error_backtraces = 0    ; solo PHP 8.5

; Equivalente en el pool de PHP-FPM (la aplicación no puede sobrescribirlo)
php_admin_value[memory_limit] = -1
php_admin_value[session.upload_progress.enabled] = 0
```

Reiniciar todos los workers para aplicar los cambios

```
systemctl restart php8.4-fpm     # ajustar a la versión instalada
systemctl restart apache2        # Apache con mod_php
```

Limitaciones y medidas adicionales

- Estas medidas reducen la exposición a las rutas de ataque confirmadas, pero no hacen segura la instalación. Solo la actualización corrige las vulnerabilidades.
- memory_limit = -1 elimina el fallo controlable por límite de memoria, pero un fallo de asignación subyacente aún puede alcanzar la ruta vulnerable. Se recomienda limitar la memoria con un contenedor o cgroup que termine solo el worker afectado; no usar RLIMIT_AS ni RLIMIT_DATA, ya que pueden provocar el fallo vulnerable.
- Limitar en el servidor web o proxy el tamaño decodificado de la petición, el número de campos, de partes multipart y de encabezados, y la profundidad de anidamiento, también para peticiones sin Content-Length válido.
- Eliminar directivas inválidas u obsoletas aplicadas por petición que generen avisos antes de la inicialización. Ocultar los avisos con error_reporting, display_errors o @ no protege.
- Corregir los errores fatales y tiempos de ejecución agotados que un atacante pueda provocar, incluidos fallos en callbacks, en la carga o importación de archivos y en la limpieza.

6. Consideraciones antes de aplicar

Las mitigaciones temporales pueden afectar funcionalidades legítimas. Se recomienda probarlas antes cuando el tiempo lo permita y retirarlas una vez instalada la versión corregida.

Cambio	Qué puede romperse
Reinicio de workers	Reiniciar PHP-FPM o Apache para aplicar la actualización o las mitigaciones interrumpe el servicio por unos segundos.
memory_limit = -1	Sin límite por proceso, un script puede consumir toda la memoria del servidor. Debe acompañarse de un límite por contenedor o cgroup.
post_max_size	Un valor de 8M o 32M rechaza formularios y cargas de archivos más grandes.
upload_max_filesize	Debe ser menor que post_max_size; un valor bajo impide cargas legítimas.
session.upload_progress	Deshabilita las barras de progreso de carga que dependen de esta función.
fatal_error_backtraces = 0	Los errores fatales dejan de registrar la traza de llamadas, lo que dificulta el diagnóstico (PHP 8.5).
Límites en web o proxy	Límites muy bajos de campos, partes o encabezados pueden rechazar formularios extensos o llamadas a API.

7. Monitoreo y señales de alerta

- Reinicios o terminaciones inesperadas de workers de PHP-FPM o procesos de Apache por falta de memoria.
- Errores fatales por memoria agotada o tiempo de ejecución excedido sin cambios en la aplicación.
- Peticiones POST o multipart inusualmente grandes, con muchos campos o partes, o sin Content-Length.
- Procesos desconocidos (sh, bash, curl, wget) ejecutados con el usuario del servidor web.
- Archivos PHP nuevos dentro del DocumentRoot o en directorios de carga.
- Conexiones salientes desde el servidor web hacia destinos no habituales.
- Se recomienda revisar los registros desde el 21 de octubre de 2026 y centralizarlos con retención mínima de seis meses.

8. Lista de verificación

Confirma que cada servidor quedó protegido. Las filas de mitigación aplican solo mientras no haya parche.

Componente	Verificación
Inventario	Servidores con PHP identificados, con rama y versión
Versión	PHP en 8.5.12, 8.4.27, 8.3.36, 8.2.35 o posterior
Paquetes	Paquetes del proveedor actualizados; sin depender de rpms.remirepo.net
Ramas EOL	PHP 8.1 o anterior con retroporte aplicado o plan de migración
Workers	PHP-FPM y Apache reiniciados después de actualizar
mod_php	Servidores con Apache y mod_php actualizados primero
Memoria	Sin parche: memory_limit = -1 y límite por contenedor o cgroup
Peticiones	post_max_size y upload_max_filesize limitados; límites en web o proxy
Sesiones	session.upload_progress.enabled = 0 en php.ini o en el pool
PHP 8.5	fatal_error_backtraces = 0 mientras no se actualice
Errores	Sin directivas inválidas por petición ni errores fatales provocables
Monitoreo	Registros revisados desde el 21 de octubre de 2026

Mensaje clave

Actualizar es la única solución completa.

Las mitigaciones temporales reducen rutas de ataque específicas, pero un servidor con una versión vulnerable sigue expuesto a ejecución remota de código sin autenticación.

Se recomienda actualizar hoy los servidores expuestos a Internet, empezando por los que usan Apache con mod_php.

Indicadores de Compromiso (IOCs) Red

Indicador	Tipo	Contexto	Reputación / Confianza
09090clami09090930032[.]duckdns[.]org	Dominio de DNS dinámico	Solicitud DNS desde el proceso inyectado; coincidencia IntelDomainHigh con XWorm (RAT)	Maliciosa
loscaminosdelavidanosoncomoyopensaba[.]duckdns[.]org	Dominio de DNS dinámico	Solicitud DNS el 28/09 a las 14:25 en el mismo contexto de actividad	Sospechosa
77[.]110[.]114[.]115:3008	IP:puerto	Conexiones salientes desde choice.exe (24 y 25/09)	Sospechosa; probable C2
45[.]92[.]1[.]229:3010	IP:puerto	Conexión saliente desde choice.exe (24/09, 16:32)	Sospechosa; probable C2
duckdns[.]org	Proveedor de DNS dinámico	Infraestructura de alojamiento del C2	Servicio legítimo abusado; monitorear o restringir según política

Recomendaciones Técnicas

Las siguientes medidas de mitigación se formulan a partir del análisis de las tácticas, técnicas y capacidades de la amenaza descritas anteriormente.

Fortalecimiento de controles básicos

- Windows Script Host: deshabilitar la ejecución de archivos .js, .vbs y .wsf a través de wscript.exe y cscript.exe en los entornos donde no sean estrictamente necesarios, o restringir su ejecución mediante políticas de AppLocker o Windows Defender Application Control (WDAC). Adicionalmente, se sugiere modificar las asociaciones de archivo mediante Directivas de Grupo (GPO) para que estas extensiones se abran por defecto con un editor de texto (ej. notepad.exe).

- Políticas de PowerShell: Habilitar y centralizar el registro avanzado de eventos (Script Block Logging, Module Logging y Transcription). Evaluar la implementación del Modo de Lenguaje Restringido (Constrained Language Mode), garantizar que la interfaz AMSI (Anti-Malware Scan Interface) permanezca activa y deshabilitar PowerShell 2.0. Asimismo, generar alertas en el SIEM/EDR sobre ejecuciones anómalas que dependan del consumo de variables de entorno.

Fuentes

- **The PHP Group. Security advisory: Critical remote code execution vulnerabilities**, 21 de octubre de 2026 (php.net/archive/2026.php).

- **The PHP Group. GHSA-w8hw-w9fx-j79w (CVE-2026-109618) y GHSA-6wmf-4rv5-p3rp (CVE-2026-109619)** (github.com/php/php-src/security).

- **The PHP Group. PHP 8.5.12, 8.4.27, 8.3.36 y 8.2.35 Released,** 21 de octubre de 2026 (php.net).

- **The PHP Group. Upcoming Critical PHP Security Release,** 20 de octubre de 2026 (php.net).

- **The PHP Group. Supported Versions y Unsupported Branches** (php.net/supported-versions.php).

¿Sabe si su organización está expuesta?

El ColCERT incluye, dentro de su portafolio de servicios, la ejecución de análisis especializados de vulnerabilidades en sitios web, servidores y demás activos de información. Estos escaneos permiten identificar de manera proactiva brechas de seguridad, configuraciones deficientes y exposiciones críticas, y entregan los insumos técnicos necesarios para que las organizaciones puedan aplicar medidas de mitigación y fortalecer su postura de ciberseguridad.

Para más información, le invitamos a comunicarse a través de los canales oficiales del CSIRT Gobierno / ColCERT.

Conéctate con el ColCERT:

contacto@colcert.gov.co

Línea directa:
+57 601 344 2222

http://www.colcert.gov.co

La información contenida en este documento, bajo clasificación TLP:CLEAR - -Pública puede ser utilizada y compartida libremente con fines informativos, técnicos y de prevención, siempre que se cite como fuente al **Equipo de Respuesta a Emergencias Cibernéticas de Colombia (ColCERT).**
Uso permitido con atribución. © ColCERT, 2026.