PHP issues emergency patches for two critical PHP2Shell remote code execution flaws



The PHP security team patched CVE-2026-109618 and CVE-2026-109619, unauthenticated RCE flaws affecting PHP installations that serve web requests. Early attacks were reportedly observed, so prompt patching is advised.

- Two unauthenticated RCE flaws, CVE-2026-109618 and CVE-2026-109619, are tracked together as PHP2Shell.
- They affect PHP serving web requests, including PHP-FPM, mod_php, php-cgi, FrankenPHP and LiteSpeed; risk is highest for mod_php.
- Fixed releases are PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35; PHP 8.1 and older no longer get project updates.
- Cloudflare reported early attacks before the patches were released.

> - Mitigations reduce exposure but do not fully block attacks; patching is the only permanent defense.

The PHP security team has released emergency patches for two critical remote code execution vulnerabilities, CVE-2026-109618 and CVE-2026-109619, collectively tracked as **PHP2Shell**. Both can be exploited without authentication. They were found by researchers from Calif, Wiz and OpenAI.

What is affected

Both flaws stem from the same underlying problem: PHP can leave its memory in a corrupted state when a request ends abnormally. They affect PHP installations that serve web requests, including PHP-FPM, Apache with mod_php, php-cgi, FrankenPHP and LiteSpeed. The PHP team considers mod_php the highest risk, with a demonstrated exploit that triggers before any application code runs. Code execution occurs with the privileges of the web worker account.

PHP is widely used, including by content management systems such as WordPress, Joomla and Drupal. The source cites VirusTotal estimates of more than 5.1 million PHP hosts exposed online.

Fixes and interim measures

Administrators of PHP 8.x servers should install 8.5.12, 8.4.27, 8.3.36 or 8.2.35. Those on PHP 7 or earlier should move to a fixed release. PHP 8.1 and older receive no project updates, although some distributions and LTS vendors provide backported fixes, and the PHP team issued minimal source backports for 7.4, 8.0 and 8.1.

Cloudflare said it had seen early attacks and that its WAF rules are only temporary. The UK NCSC advised updating or taking affected systems offline. PHP's published mitigations, such as setting memory_limit=-1, lowering post_max_size and disabling session.upload_progress, reduce some attack paths but do not fully protect. Patching remains the only permanent defense.

What to do now

1. Inventory all servers and applications running PHP, including CMS platforms and containers, and identify versions and SAPI (mod_php, PHP-FPM, etc.).
2. Upgrade to PHP 8.5.12, 8.4.27, 8.3.36 or 8.2.35 as an emergency change, prioritizing internet-facing systems and mod_php deployments.
3. For PHP 8.1 or older, obtain a vendor backport or migrate to a supported version; consider taking exposed systems offline if patching is delayed.
4. Where patching must wait, apply PHP's published mitigations and restart workers, treating them as temporary only.
5. Review web and WAF logs for abnormal requests and crashes, and escalate suspected compromise

through your incident response process.