

PHP2Shell: Critical Remote Code Execution in the PHP Runtime

CVE-2026-109618 and CVE-2026-109619

21 October 2026 | Reference TA-2026-1021-001

1. What You Need to Know

On 21 October 2026, the PHP project released security updates addressing two vulnerabilities in the PHP engine, publicly referred to as PHP2Shell. Both vulnerabilities permit an unauthenticated remote party to execute code on a web server running PHP. One of them also permits the disclosure of process memory, which may contain credentials, session tokens, API keys and certificates.

Because the defects reside in the PHP runtime rather than in application code, organisations may be exposed irrespective of the security of their own applications. Exploitation attempts were reported prior to the release of fixed versions. We recommend that organisations upgrade all PHP runtimes to a fixed release as a matter of priority. Configuration changes published alongside the release reduce exposure to specific attack paths but do not constitute remediation.

Overall rating	CRITICAL (CVSS v4.0 9.2 for both vulnerabilities)
Affected component	PHP engine, across Apache mod_php, PHP-FPM, php-cgi, FrankenPHP and LiteSpeed deployments
Authentication required	None
Potential impact	Remote code execution under the PHP worker account; disclosure of data held in worker memory
Remediated releases	PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35, published 21 October 2026 at 14:00 UTC
Exploitation status	Attempted exploitation observed; no confirmed compromise reported as at the date of this advisory

2. What Happened

The vulnerabilities were identified by Wiz Research, Calif and the OpenAI Codex Security team, and were remediated in coordination with the PHP security team and Google. Coordinated disclosure had been scheduled for 22 October 2026. Following the publication of partial details on 20 October, the PHP project brought the release forward by one day. The chronology below summarises events as reported at the date of this advisory.

Date	Event
20 October 2026	Details referencing both CVE identifiers are published ahead of the scheduled disclosure. The PHP project issues an advance notice and moves the release to 21 October.
21 October 2026	Cloudflare reports attempted exploitation and deploys interim web application firewall rules. watchTower reproduces CVE-2026-109618 and reports no confirmed in-the-wild exploitation as at 11:26 UTC.
21 October 2026	The PHP project publishes fixed releases for the 8.2 to 8.5 branches, together with source backports for PHP 7.4, 8.0 and 8.1. Linux distributions and long-term support vendors received the patches in advance.
21 October 2026	A number of organisations, including BambooHR and several Dutch public sector services, take systems offline as a precautionary measure pending remediation.

The researchers have stated that working exploits were developed within approximately one week. Given that the vulnerability details and patch changes are now public, organisations should plan on the basis that the period before broader exploitation will be short.

3. How the Vulnerabilities Work

Each PHP request is processed by a long-running worker process, which is expected to initialise fresh internal state at the start of each request and release it on completion. Both vulnerabilities arise from failures in this lifecycle.

3.1 CVE-2026-109618: Out-of-Memory Corruption

Where PHP exhausts available memory during request initialisation, processing may be interrupted while internal request structures are only partially populated. The subsequent clean-up routine treats these structures as valid, resulting in memory corruption. A remote party able to induce this condition, for example through a large or deeply nested request, may gain control of the worker process. The researchers demonstrated unauthenticated code execution against current releases of WordPress, Drupal and Joomla.

3.2 CVE-2026-109619: Stale Global State

Certain internal state associated with a request may persist after the underlying memory has been released. When the same worker processes a subsequent request, that residual state may be referenced before it is reinitialised, constituting a use-after-free condition. The researchers demonstrated unauthenticated code execution against WordPress and Drupal and, separately, the disclosure of live worker memory independent of the hosted application.

Vulnerability	Rating	Potential Impact	Deployment Models Confirmed
CVE-2026-109618 Out-of-memory corruption	CRITICAL 9.2	Unauthenticated code execution under the PHP worker account.	Demonstrated on Apache mod_php. Considered applicable to PHP-FPM; no complete FPM exploit published.
CVE-2026-109619 Stale global state	CRITICAL 9.2	Unauthenticated disclosure of worker memory or code execution.	Demonstrated on Apache mod_php and on PHP-FPM, the latter dependent on application behaviour.

4. Who Is Affected

All PHP installations that process web requests are within scope. The PHP project assesses Apache mod_php as carrying the highest risk, as a single unauthenticated request can trigger the condition before any application code is executed. PHP-FPM terminates a worker where a request fails early in initialisation, which raises the difficulty of exploitation but does not remove the exposure. Releases from PHP 5.4 onwards are affected by at least one of the two vulnerabilities.

PHP Branch	Affected Releases	First Fixed Release	Remarks
8.5	Below 8.5.12	8.5.12	Supported branch
8.4	Below 8.4.27	8.4.27	Supported branch
8.3	Below 8.3.36	8.3.36	Supported branch
8.2	Below 8.2.35	8.2.35	End of life on 31 December 2026
8.1, 8.0, 7.4	All releases	Backport only	Outside PHP project support; source backports or vendor fix
7.3 and earlier	Most releases	No fix	Migration to a supported branch required

Linux distributions and long-term support vendors frequently apply security fixes without altering the upstream version number. Patch status should therefore be confirmed against the package changelog for both CVE identifiers rather than the reported version. Organisations should also note that updating an application such as WordPress or Laravel does not update the underlying PHP runtime, which is supplied by the operating system, hosting provider or container base image.

5. What You Should Do

5.1 Immediate Actions Checklist

We recommend the following actions, with internet-facing systems addressed ahead of internal applications.

Ref	Priority	Recommendation
1	Immediate	Upgrade all PHP runtimes to 8.5.12, 8.4.27, 8.3.36 or 8.2.35, or to a vendor package that references both CVE identifiers. Restart Apache, PHP-FPM and all long-running PHP workers, and rebuild container images that include PHP.
2	Immediate	For systems on PHP 8.1 or earlier, apply the vendor or PHP project backport, or migrate the application to a supported branch.
3	Short term	Where an upgrade cannot be completed immediately, apply the compensating controls set out in Section 5.2 and confirm that provider web application firewall rules are active.

Ref	Priority	Recommendation
4	Short term	Undertake the compromise review described in Section 5.4 for systems exposed prior to remediation, and rotate secrets held in PHP worker memory, including database credentials, API keys and session signing keys.
5	Follow-up	Complete the post-remediation review in Section 5.3 and establish a migration plan for systems on PHP 8.2 ahead of its end of life on 31 December 2026.

5.2 If You Cannot Patch Yet

The PHP project has confirmed that the following measures reduce exposure to the demonstrated attack paths but do not remediate either vulnerability. They should be regarded as temporary. Settings must be applied in php.ini, the PHP-FPM pool or the web server configuration, as ini_set() and .user.ini files take effect too late in request processing. All workers should be restarted following any change.

- CVE-2026-109618:** set memory_limit=-1 in a manner that applications cannot override, with a container or cgroup memory limit to protect availability; set post_max_size to 8M, or 32M where uploads are required; and enforce request size, field count and multipart limits at the web server or proxy. RLIMIT_AS and RLIMIT_DATA should not be used, as they can trigger the same condition.
- CVE-2026-109619:** set session.upload_progress.enabled=0 and, on PHP 8.5, fatal_error_backtraces=0. This addresses the confirmed session-based path only and has no effect on CVE-2026-109618.
- General:** set display_errors=0, reduce pm.max_requests to recycle PHP-FPM workers more frequently, and confine the PHP worker using SELinux or AppArmor.

5.3 After Patching

The fix introduces a hard memory limit. A process that reaches this limit is terminated immediately and shutdown handlers are not executed, which may affect applications that depend on them for logging or clean-up. We recommend that memory_limit and post_max_size are reviewed against application requirements, that any temporary memory_limit=-1 setting is reversed, and that expose_php is set to 0 to prevent disclosure of the PHP version in response headers.

5.4 Checking for Earlier Compromise

Remediation does not address access that may already have been obtained. For systems exposed before the update, the researchers recommend that web server and PHP logs are reviewed for unusual multipart requests followed by memory-limit errors or repeated HTTP 500 responses, abnormally large PHPSESSID values in response cookies, unexplained PHP worker restarts, and unexpected file changes in web roots or processes initiated by the web server account.

Risk: **CRITICAL** **HIGH** **MEDIUM** **TO VALIDATE** Status: **OPEN** **IN PROGRESS** **REMEDIATED**

We will shortly share the list of potentially impacted assets in your external estate

Meanwhile, you may directly check for potentially affected assets in **NST Assure** EA datasets.

How to check this in NST Assure. In NST Assure, go to Exposure Assessment and open the Technology Insights and Exploitation Intelligence datasets. Search for “php”, or a related keyword such as “X-Powered-By”, “PHPSESSID”, “WordPress” or “Drupal”, to find PHP instances across your external estate.

Sources: [PHP security advisory](#) | [GHSA-w8hw-w9fx-j79w](#) | [GHSA-6wmf-4rv5-p3rp](#) | [php2shell.com](#) | [Cloudflare](#) | [The Hacker News](#)

This advisory is based on information publicly available as at 21 October 2026 and is provided for the recipient's internal use. Findings from passive detection should be validated before remediation decisions are finalised.