DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

CYBER THREAT ADVISORY

# PHP2SHELL VULNERABILITY

## Immediate Security Measures for Web Applications and Servers

OCTOBER 2026

# PHP2Shell Advisory:
# CVE-2026-109618 and CVE-2026-109619

## Overview

Two critical flaws in the PHP engine (CVSS v4.0 9.2 each) allow an unauthenticated attacker run code on a web server. The fix has been shipped on 21 October 2026, so it is recommended to immediately update PHP now, restart the web workers and check internet-facing hosts for any compromise, since 17th October.

Both bugs belong to the same family of memory-safety flaws in how PHP handles fatal errors, especially out-of-memory errors, but they are separate bugs with different mechanisms. In CVE-2026-109618, PHP can be interrupted while setting up a request and then processes the resulting invalid state during that same request's cleanup. In CVE-2026-109619, state left behind by a failed request is picked up by the next request handled by the same worker. In both cases, an attacker who controls what sits in that memory can steer what PHP does next.

All installations that serve web requests are at risk: PHP-FPM, Apache with mod_php, php-cgi, FrankenPHP and LiteSpeed with the risk being highest for **mod_ph.** Updating is the only complete fix.

## Vulnerability Technical Analysis

**CVE-2026-109618: Partly initialized data treated as valid -** While setting up a request, PHP creates internal records for the incoming data. PHP describes the flaw as request memory that can be reachable before it is fully initialized, and notes that the allocation that fails does not need to be related to that memory. In plain terms, a record is treated as complete before it has been filled in. An attacker sends a request crafted to exhaust PHP's memory while the record is in that state. The record is left looking complete but holds leftover bytes from earlier use of that memory. During cleanup, PHP treats those bytes as valid data and acts on them. Because the attacker shapes those bytes, they can steer what PHP does. On Apache with mod_php this happens before any of the website's own code runs, so the security of the application does not matter.

**CVE-2026-109619: Stale references to freed memory -** PHP keeps references that point at memory belonging to the current request. When a request ends in a fatal error, that memory is freed but some of the references are not cleared. The next request handled by the same worker can follow a reference into memory that no longer holds what PHP expects. The attacker arranges for their own data to occupy that freed memory, so PHP ends up using attacker-controlled data as if it were its own. Depending on the path, this leaks memory contents such as passwords or API keys or allows code execution.

## Threat Overview and Severity

- No login is needed. A crafted web request is enough.

- Attackers can force the out-of-memory error on purpose with requests built to consume memory.
- Injected code runs as the web server's account and can read whatever the website can read.

## Security Implications

A successful attack runs code as the web worker account, such as www-data, apache or nginx, without logging in. From there an attacker can read application secrets, steal data, deface sites, install web shells and pivot to other systems that the host's credentials can reach. Separately, one bug discloses live worker memory without needing code execution. It can expose passwords, tokens, API keys and certificates belonging to other requests or to the server.

Because the flaws sit in PHP itself, the application on top does not need to be vulnerable. Researchers demonstrated unauthenticated RCE against current versions of WordPress, Drupal, Joomla and other applications, and Wiz notes that even a minimal script on an affected PHP version can be exposed.

## Affected Products and Environments

All installations that serve web requests are at risk: PHP-FPM, Apache with mod_php, php-cgi, FrankenPHP and LiteSpeed. Risk is highest for **mod_php**, where PHP demonstrated an unauthenticated request that triggers the attack before any application code runs, using default settings.

The earliest affected code dates to PHP 5.4.0, and every release of the past decade carries at least one of the bugs. Source analysis suggests CVE-2026-109618 may affect FPM, but no complete FPM exploit has been demonstrated. The validated FPM attack uses CVE-2026-109619: application code stores some state after a normal startup and then hits a fatal error, and on a later request the session upload progress feature uses the stale state. It works against a normal FPM pool but depends on a suitable application operation. The application does not need to call session_start().

| PHP branch | Status |
|---|---|
| 8.5, 8.4, 8.3, 8.2 | Fixed in 8.5.12, 8.4.27, 8.3.36 and 8.2.35 |
| 8.1, 8.0, 7.4 | No official release. PHP published source-only backport branches; vendors may offer their own |
| Older than 7.4 | No fix from PHP. Upgrade or isolate |

## Mitigations and Remediations

Updating is the only complete fix. PHP states that none of its workarounds make an installation safe.

1. **Find every PHP runtime -** host packages, container and CI images, hosting accounts and Windows hosts running php-cgi. Check versions with php -v or php-fpm -v.

2. **Update** to 8.5.12, 8.4.27, 8.3.36 or 8.2.35. Use your distribution or LTS vendor's packages, which received the fix early. Backported builds may keep the old version string, so confirm the CVE IDs in the package changelog.
3. **Restart workers -** restart Apache or PHP-FPM so running processes load the patched code. Patch internet-facing hosts first.
4. **Rebuild containers** and confirm the base image now resolves to a fixed version.
5. **Isolate what you cannot patch.** The UK's NCSC is urging organisations to consider taking systems that cannot be fixed immediately offline (as reported by Help Net Security). The FBI goes further, recommending that a system be taken offline if an immediate update isn't feasible.

| Setting | Addresses | Limit |
|---|---|---|
| Set memory_limit=-1 in php.ini, an FPM pool or server configuration, where applications cannot override it | CVE-2026-109618 | Removes the PHP memory limit failure, but an underlying allocation failure can still reach the vulnerable path. Needs a separate memory cap |
| Set low post_max_size (8M without uploads, 32M with) plus proxy limits that also cover requests without Content-Length | CVE-2026-109618 | Partial |
| session.upload_progress.enabled=0 in php.ini or the FPM pool, then restart workers | CVE-2026-109619 | Blocks the confirmed session paths only |
| fatal_error_backtraces=0 on PHP 8.5, then restart workers | CVE-2026-109619 | Blocks one confirmed path only |

6. Identify where PHP was installed from first. On Linux servers, PHP usually comes from one of two sources, and the fix differs for each:

| Install Source | What to do | Target version |
|---|---|---|
| OS vendor repository (e.g., Ubuntu archive) | The vendor has already backported the fix into its own packages. Run apt update and apt upgrade, then restart workers | The vendor's fixed build number. The upstream version string may look old, so check the build against the vendor's security notice |
| Third-party package repository (Ondřej Surý / sury.org) | Upgrade to the upstream fixed release | 8.5.12 (8.4.27, 8.3.36 or 8.2.35 for other branches) |

Ubuntu's notice lists fixed builds per release: 8.5.9-0ubuntu2.1 on 26.10, 8.5.4-0ubuntu1.5 on 26.04 LTS, 8.3.6-0ubuntu0.24.04.13 on 24.04 LTS and 8.1.2-1ubuntu2.28 on 22.04 LTS.

## References


- PHP news advisory and releases, 21 October 2026
- The PHP Foundation: Today's PHP Security Release
- Wiz Research - PHP2Shell: AI uncovers critical RCE vulnerabilities affecting 70% of the Internet
- The Hacker News: PHP Issues Emergency Patches for Two Critical PHP2Shell RCE Flaws as Exploit Attempts Emerge
- Amazon Threat Intelligence identifies PHP2Shell exploitation prior to public disclosure in MadPot
- GitHub: RCE via OOM corruption
- GitHub: RCE via stale global state


## Disclaimer

*This advisory is produced for informational purposes only and is intended to help organizations understand and respond to the PHP2Shell vulnerabilities (CVE-2026-109618 and CVE-2026-109619). The information contained herein is based on publicly available information as of 22 October 2026, and is subject to change as patches, mitigations, exploitation activity and attribution evolve.*

*However, DSCI disclaims all warranties as to the accuracy, completeness or adequacy of this information. The information contained herein should not be relied upon as a substitute for specific professional advice or for your own testing and validation before changes are applied to production systems.*

DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

# ABOUT DSCI

Data Security Council of India (DSCI), set up by Nasscom, is a not-for-profit industry body and think tank on data protection, cyber security, and critical technologies in India. DSCI is committed to making cyberspace safe, secure, and privacy-assured, and to advancing critical technologies vital for national security through policy advocacy, thought leadership, capacity building, and outreach initiatives. DSCI engages with the government, regulators, defence, law enforcement agencies, embassies, industry members, and academia to position India as a leader in the global digital ecosystem.

**Data Security Council of India**
4th Floor, Nasscom Campus, Plot No. 7-10, Sector 126, Noida, UP-201303

data-security-council-of-india/ DSCI_Connect dsci.connect

dsci.connect dscivideo security chips

All Rights Reserved © DSCI 2026