PHP patches phpAfterShell, a critical session-handling flaw tied to PHP2Shell



PHP fixed CVE-2026-126714 (CVSS 9.2), a flaw in session file writes that can lead to remote code execution under specific conditions. No in-the-wild exploitation has been reported.

- CVE-2026-126714, named phpAfterShell, has a CVSS score of 9.2 and was fixed on November 19 with six other critical issues.
- PHP overwrites session files without emptying them first, so an interrupted write leaves mixed fresh and old data that can trigger object deserialization.
- Exploitation needs several conditions, including default session storage, an exploitation route such as upload progress tracking, and a deserialization-to-code-execution path.
- Fixed versions are PHP 8.6.0, 8.5.13, 8.4.28, 8.3.37 and 8.2.36.
- No in-the-wild exploitation has been reported.

PHP has patched a critical vulnerability, tracked as CVE-2026-126714 and named phpAfterShell by the security firm Depthfirst. It carries a CVSS score of 9.2 and was fixed on November 19 together with six other critical issues.

What the flaw is

PHP overwrites a session file from the beginning without emptying it first. If the write is interrupted before the later truncation step, the file holds fresh session data followed by leftovers of the old session. Depthfirst reports reliable ways to interrupt the operation during session upload-progress updates, which lets an attacker splice parts of two valid sessions. A later session operation then treats the corrupted data as legitimate, giving the attacker control over session values and the ability to trigger object deserialization, which can lead to **remote code execution**.

The issue is related to PHP2Shell. That earlier flaw involved the upload-progress callback decoding a file-backed session before the target script ran. PHP restricted which objects could be created, but the underlying file update behaviour was not changed.

Conditions and status

Exploitation is not trivial. It requires default session storage, an exploitation route (upload progress tracking is the easiest), the PHP process stopping mid-write, a request calling session_start() on the corrupted file, and a path from deserialization to code execution. The PHP advisory notes that the same problem can arise when an application stores large user-controlled strings in a session, and that in threaded builds a memory-limit failure in another thread can end the process during a write.

Patches are in PHP 8.6.0, 8.5.13, 8.4.28, 8.3.37 and 8.2.36. No in-the-wild exploitation has been reported so far.

What to do now

1. Identify PHP-based applications and servers and record their versions against the fixed releases: 8.6.0, 8.5.13, 8.4.28, 8.3.37, 8.2.36.
2. Schedule upgrades to a fixed version through change management, prioritising internet-facing systems.
3. Check whether upload progress tracking is enabled and whether default file-based session storage is used, and disable features that are not needed.
4. Review applications for storing large user-controlled strings in sessions and for unsafe deserialization paths.
5. Confirm earlier PHP2Shell fixes were applied, and monitor web logs for abnormal multipart upload requests.