



URGENT SECURITY BULLETIN

PHP2Shell | 22 October 2026

For website owners and managers

ACTION REQUIRED: VERIFY AND UPDATE PHP

Two critical PHP security flaws were fixed on 21 October 2026. **Ask your IT provider to check your site and patch accordingly.** They should verify that the PHP serving your site contains both fixes and that the updated workers are running. ^[1]

What is PHP2Shell?

PHP is software on a server that builds web pages and runs online services. PHP2Shell is the name given to two serious faults in that engine. Under exploitable conditions, an attacker can send specially crafted web requests to run commands or read sensitive server-process memory, without first signing in. ^[4]

WHO SHOULD ACT

Cloud servers, webserver and PHP-backed online services need checking wherever an affected PHP installation handles web requests. Shared hosting, virtual machines and containers are all in scope. ^[1]

WordPress runs on PHP. Your IT provider should check the PHP runtime as well as WordPress, themes and plugins. Updating the website application alone does not establish that PHP itself has been fixed. ^{[6],[7]}

WHY IT MATTERS

1. Data exposure

A successful attack may expose passwords, tokens or other sensitive data held by PHP. ^[4]

2. Website compromise

An attacker may gain the PHP worker's ability to run code, alter accessible files or access connected services. ^[4]

3. Service disruption

Crashes, emergency maintenance and recovery can interrupt business.

WIDELY USED ACROSS THE WEB

PHP is used by **69.3% of websites whose server-side language W3Techs can identify**, as measured on 22 October 2026. That is roughly seven in ten of the measured group. The figure covers sites where that technology could be identified. Exposure depends on the installed PHP build and its patch state. ^[5]

START HERE

Watch the short video: [Ask Baxter: PHP2Shell website vulnerability](#)

Run the free public check: togethia.tools/php2shell - no account required.

The online checker is a free initial check. Complete professional checks and patching involve a cost; agree scope and charges with your IT provider.



URGENT SECURITY BULLETIN

PHP2Shell | 22 October 2026

Plain-English action guide

Check carefully. Patch promptly.

A safe web check cannot establish the patch state of every website.

TWO PRACTICAL OUTCOMES FOR THE OWNER

Confirmed affected - "definitely yes". Your IT provider must verify vulnerable installed packages or perform authorised controlled validation. An affected-version banner is an urgent warning, but cannot prove exploitability or exclude a vendor backport.

Further checks needed / unknown. Hidden PHP, proxies and firewalls can prevent a verdict. Even a fixed-version banner does not prove every worker is patched. A URL-only check cannot give a universal "definitely safe" result. ^{[8],[10]}

WHY THE TEST DOES NOT TRY TO PROVE EVERY CASE

Definitive tests can exhaust memory or corrupt a PHP worker, crashing or disrupting a site. A failed attempt does not prove a patch. Togethia inspects normal public headers and **does not run exploits**. Some websites cannot be conclusively checked from outside without potentially damaging tests. ^[10]

Your IT provider's usual work process

Applying a supported PHP patch is usually straightforward. The complete work process typically includes the following steps. **This may take a reasonable amount of time per website.** ^[7]

1 Check your site is compatible with the patches

Review the site's PHP version, applications, plugins and extensions. Check patch compatibility in a representative environment where practical to avoid unexpected results. ^[7]

2 Back up your data

Back up website files, configuration and databases, and confirm restoration is possible, before applying changes.

3 Apply the patch

Apply both PHP2Shell fixes to every relevant runtime, then restart or replace the PHP workers. ^{[1],[4]}

4 Test the website afterwards

Check relevant logins, forms, uploads, checkout, integrations and scheduled jobs. Review errors and confirm normal operation.

Costs for a complete assessment

There will be a cost involved for the complete checks pertinent to your online services. Ask your IT provider to confirm the scope and charges, including patching and testing.



URGENT SECURITY BULLETIN

PHP2Shell | 22 October 2026
Technical appendix A - assessment

Technical detail for administrators

CVE-2026-109618: OOM CORRUPTION

PHP can leave partially initialised request data reachable when an out-of-memory error causes an early exit via **longjmp()**. Cleanup can then operate on invalid state. Crafted inputs can reach this condition and lead to unauthenticated RCE. ^[2]

The demonstrated mod_php path can act before application code runs. Reliability depends on the SAPI, build, extensions, request handling and allocator. ^[2]

CVE-2026-109619: STALE GLOBAL STATE

A persistent worker can retain a pointer to memory owned by a finished request. A later request handled by the same worker may consume it before normal initialisation resets it, producing a **use-after-free**. ^[3]

Producing/consuming paths include error and session handling. Confirmed impacts include unauthenticated memory disclosure and RCE. Worker recycling may reduce per-request success; it does not remove the underlying flaw. ^[3]

Severity: both PHP advisories rate the flaws **Critical, CVSS v4.0 9.2**. The vector is AV:N/AC:H/AT:P/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N. Exploitation is deployment-dependent; no login or victim interaction is required for the demonstrated paths. ^{[2],[3]}

First fixed upstream releases

PHP branch	First fixed release	Earlier branch releases
8.2	8.2.35	Require both fixes / vendor verification
8.3	8.3.36	Require both fixes / vendor verification
8.4	8.4.27	Require both fixes / vendor verification
8.5	8.5.12	Require both fixes / vendor verification

PHP 8.1 and older need a verified vendor fix/backport or an upgrade. Distribution packages may retain an older upstream-looking version while containing the fixes. Compare the full package release with its vendor advisory, rather than applying this upstream table blindly. ^{[1],[9]}

REMOTE CHECKER: EVIDENCE BOUNDARIES

Ordinary GET requests inspect Server / X-Powered-By evidence. Bounds: one IPv4 vantage point, two addresses per hostname, three redirects, eight attempts and 22 seconds. The exact `www/apex` pair and root `/index.php` fallback are supported. No response bodies, uploads or exploit probes are used. ^[10]

An affected-version flag is triage evidence; a fixed-version flag is not installed-package proof. Hidden versions, session hints, access blocks, TLS/DNS failures and third-party redirects cannot establish a patch. PHPSESSID alone is only a hint. Authorised host-side verification remains necessary.



URGENT SECURITY BULLETIN

PHP2Shell | 22 October 2026

Technical appendix B - verification

Remediation and acceptance evidence

- 1 Inventory the real serving stack**
Map each virtual host, PHP-FPM pool, mod_php instance, container image and replica, including origins behind CDNs. Include PHP-backed administration services and reused images. The CLI binary may differ from the web runtime.
- 2 Prove the package contains both fixes**
Record vendor, repository, full package/build release and the advisory covering each CVE. Prefer a supported patched branch or a vendor-supported backport. Rebuild containers from corrected bases; avoid reintroducing old images. ^[9]
- 3 Check, back up, patch and test each website**
The IT provider should verify patch compatibility, back up data and confirm restoration, apply both fixes, replace affected workers and test key functions. Allow a reasonable amount of time per website and agree the scope, charges and maintenance plan. ^[7]
- 4 Accept on evidence, not banners**
Record running process/build verification, restart/replacement times, every origin/pool covered, representative functional tests and post-change error/availability checks. A successful HTTP response or empty error log is not proof of a patch. Investigate possible prior compromise separately; a security update does not remove an existing intrusion.

TEMPORARY MEASURES ARE INCOMPLETE

Only an update containing both fixes is complete remediation. ^[1] If updates are blocked, consult the PHP team's deployment-specific advice. `session.upload_progress.enabled=0` and, on PHP 8.5, `fatal_error_backtraces=0` require worker restarts and block specific stale-state paths only. ^[3] Memory/request-limit changes can affect availability and compatibility; do not apply them blindly. ^[2] WAF rules, suppressed diagnostics and worker recycling do not establish safety.

Sources and further reading

Primary technical guidance checked 22 October 2026. Links are clickable. Recheck vendor status at deployment time; this bulletin is dated guidance, not a scan of any particular installation.

[1] [PHP security release and upgrade advice](#)

[3] [PHP: CVE-2026-109619 / stale global state](#)

[5] [W3Techs: PHP usage, 22 October 2026](#)

[7] [WordPress: preparing for a PHP update](#)

[9] [Red Hat: security backporting policy](#)

[2] [PHP: CVE-2026-109618 / OOM corruption](#)

[4] [Discoverers: PHP2Shell overview](#)

[6] [WordPress: hosting requirements](#)

[8] [Tenable: version-based detection limits](#)

[10] [Togethia: public checker and method](#)

Short video: [askbaxter.dog/videos/php2shell-website-vulnerability/](#)