

PHP2Shell: Critical PHP Flaws Put Your Web Servers at Risk

Two vulnerabilities let attackers take control of unpatched PHP web servers without a password. Here is what they mean for your business, and what to do today.

SEVERITY	CVSS V4.0	LOGIN REQUIRED	EXPLOITATION	FIX
CRITICAL	9.2 (both)	None	Attempts reported	Available now

The bottom line

Emergency patches are disruptive. This one is worth the disruption.

On October 21, 2026, the PHP project released emergency updates for two critical vulnerabilities, now widely known as PHP2Shell (CVE-2026-109618 and CVE-2026-109619). Either flaw can let an attacker run code on an unpatched web server using specially crafted web requests: no password, no stolen credentials, and no one clicking a link.

PHP powers a large share of the web, including corporate websites, client and investor portals, WordPress and other content platforms, e-commerce sites, and the management consoles built into many vendor products. If your organization runs PHP anywhere, or relies on vendors who do, assume you are exposed until you confirm otherwise.

The window to act is short. Details leaked a day before the fix was released, proof-of-concept code is circulating, and several security firms reported exploitation attempts the same day patches shipped. CISA had not observed exploitation when it issued its alert, but it joined the FBI in urging organizations to patch immediately and to take systems offline if they cannot.

Four things to do today

- 01 Find every system running PHP, starting with anything reachable from the internet.
- 02 Update to PHP 8.5.12, 8.4.27, 8.3.36, or 8.2.35, or your vendor's patched package, then restart every PHP process.
- 03 Take offline or restrict access to any system you cannot patch today.
- 04 Ask hosting providers and critical vendors to confirm they have patched.

Why this matters to your business

PHP2Shell is not a niche software bug. It is a direct path from the public internet to the servers that hold your data and represent your brand.

01 Full server takeover

Successful exploitation runs the attacker's code with the permissions of the web server. Anything the application can reach, such as client records, database credentials, and API keys, is within the attacker's reach too.

02 Data exposure without a takeover

The PHP project confirms that CVE-2026-109619 can also disclose server memory. That memory may hold passwords, session tokens, and data from other visitors' requests.

03 Application defenses do not help

On Apache with mod_php, the PHP project demonstrated an attack that triggers before any application code runs. Login pages, application security controls, and secure coding practices cannot stop it. Hiding error messages does not help either.

04 Exposure you cannot easily see

PHP is often embedded inside other software. Hosting platforms, SaaS providers, network appliances, and business applications may run vulnerable PHP without listing it anywhere you would look. CISA warns that its prevalence may not be directly observable.

REGULATORY AND STAKEHOLDER EXPECTATIONS

Regulators, assessors, insurers, and investors increasingly judge organizations by how quickly and deliberately they respond to known critical vulnerabilities. If your firm answers to SEC Regulation S-P, CMMC, HIPAA, or cyber insurance attestations, expect to show when you learned of PHP2Shell, which systems were affected, when each was fixed, and how you managed exceptions. A well-documented response turns an emergency into evidence of a mature program.

Scope

Who is affected

Every PHP installation that serves web requests on an affected version is vulnerable, regardless of the application it runs. Use the table below to confirm your versions.

PHP BRANCH	VULNERABLE	FIXED VERSION	WHAT TO KNOW
8.5	Before 8.5.12	8.5.12	Current release line.
8.4	Before 8.4.27	8.4.27	Supported.
8.3	Before 8.3.36	8.3.36	Supported.
8.2	Before 8.2.35	8.2.35	Reaches end of life December 31, 2026. Patch now and plan an upgrade.
8.1 and older	Assume vulnerable	No official release	No longer supported. Use a distribution or long-term support vendor backport, or the PHP project's minimal source backport (7.4, 8.0, 8.1). Otherwise upgrade or take the system offline.

How the vulnerabilities work

Both flaws live in PHP itself, not in any one application, and both share the same CVSS v4.0 score of 9.2. Neither requires a login or any action by a user.

CVE-2026-109618

Code execution through memory exhaustion

Critical | CVSS v4.0 9.2 | No login required

An attacker sends requests designed to make PHP run out of memory at a precise moment, while it is still setting up internal data for the request. When PHP cleans up, it trusts that half-built data, and the attacker can turn that mistake into running their own code.

Confirmed: full remote code execution against Apache mod_php with default PHP settings.

CVE-2026-109619

Code execution through leftover request data

Critical | CVSS v4.0 9.2 | No login required

Internal data from one request can survive after the memory behind it has been released. A later request handled by the same PHP worker can use that stale data, letting an attacker read server memory or run code. A common attack path abuses a session feature that is on by default.

Confirmed: remote code execution against Apache mod_php and PHP-FPM, plus memory disclosure.

Risk by deployment type

HOW PHP IS DEPLOYED	EXPOSURE	WHAT THE PHP PROJECT HAS CONFIRMED
Apache with mod_php	HIGHEST	Remote code execution validated for both vulnerabilities. The attack can trigger before any application code runs.
PHP-FPM (Nginx, Apache proxy)	HIGH	Remote code execution validated for CVE-2026-109619. Source analysis shows susceptibility to CVE-2026-109618.
FrankenPHP, LiteSpeed, php-cgi	AT RISK	Named by the PHP project as at risk. Not validated; the project's analysis ranges from likely to conditional. Treat as vulnerable.

Recommended actions

Your action plan

Act in order of exposure. Internet-facing systems that handle sensitive data come first.

TODAY

Within 24 hours

01 Build a complete PHP inventory

Identify every server, virtual machine, container image, and hosted service running PHP, including vendor products and appliances that bundle it. Rank each by internet exposure and by the sensitivity of the data it touches.

02 Patch, restart, and verify

Apply PHP 8.5.12, 8.4.27, 8.3.36, or 8.2.35, or the patched package from your Linux distribution or long-term support vendor, which received the fixes in advance. Restart every web server, PHP-FPM pool, and long-running PHP worker: a patched system is not protected until the processes using it restart. Then confirm the version your web server is actually running, not just the command-line version.

03 **Isolate what you cannot patch**

If a system cannot be updated today, or runs PHP 8.1 or older, take it offline or restrict access through a VPN or IP allowlist until it is fixed. A web application firewall may provide partial protection in the meantime. The interim settings in the appendix reduce risk, but none of them is a complete fix.

04 **Check where your packages come from**

If you rely on the volunteer-maintained remirepo repository, the PHP project advises switching to distribution packages or building from source. That repository did not have updates on release day.

THIS WEEK

Within 7 days

05 **Look for signs of compromise**

For any internet-facing system that ran a vulnerable version after details leaked on October 20, review web server and PHP logs for repeated worker crashes, out-of-memory errors, and unusual form uploads. Check web directories for unfamiliar files and watch for unexpected outbound connections. If anything looks wrong, treat it as an incident: isolate the system, preserve evidence, and rotate every credential the application held.

06 **Confirm your vendors are covered**

Contact hosting providers, managed service providers, and software vendors that run public-facing services or hold sensitive data for you. Ask whether they were affected, when they patched, and whether they found signs of compromise. Record their answers.

07 **Document the response**

Capture the inventory, patch dates, exceptions, compensating controls, and vendor confirmations in one record. That record supports regulatory examinations, compliance assessments, insurance renewals, and investor inquiries.

THIS QUARTER

Before year end

08 **Retire unsupported PHP**

PHP 8.1 and older no longer receive fixes from the PHP project, and PHP 8.2 reaches end of life on December 31, 2026. Build the upgrade plan now, not during the next emergency.

09 **Keep the hardening in place**

The PHP project suggests it may be prudent to leave session upload progress disabled even after patching wherever applications do not need it.

10 **Shorten your response time**

PHP2Shell tests how fast your organization can answer three questions: where is it, who owns it, and how quickly can we fix it? A continuously maintained asset inventory, a named owner for every internet-facing system, and a tested emergency patch process turn the next critical disclosure into a routine change.

FOR PRIVATE EQUITY SPONSORS AND MULTI-ENTITY ORGANIZATIONS

Exposure rarely stops at one company.

Ask each portfolio company or business unit to confirm by a set date whether PHP runs in their environment, whether internet-facing systems are patched, and which exceptions remain. Consolidate the answers into a single view of portfolio risk, and follow up on any company that cannot answer with confidence.

Interim measures if you cannot update immediately

These measures, drawn from the PHP project's advisories, reduce exposure to specific attack paths. They do not make an installation safe. **Updating PHP is the only complete fix.** Apply settings in `php.ini`, the PHP-FPM pool, or the web server configuration, then restart every worker.

MEASURE	ADDRESSES	LIMITATIONS
Disable session upload progress: <code>session.upload_progress.enabled=0</code>	CVE-2026-109619	Blocks the confirmed session attack paths, not other variants. Setting it through <code>ini_set()</code> or an FPM <code>.user.ini</code> file applies too late and does not work.
PHP 8.5 only: <code>fatal_error_backtraces=0</code>	CVE-2026-109619	Blocks the confirmed fatal error backtrace path only.
Remove and enforce the PHP memory limit: <code>memory_limit=-1</code> through <code>php_admin_value</code>	CVE-2026-109618	Confirm no pool, directory, or application setting restores a finite limit. Lower-level allocation failures can still reach the flaw. Pair with a container or cgroup memory limit that stops only the affected worker. Do not use <code>RLIMIT_AS</code> or <code>RLIMIT_DATA</code> , which can trigger the flaw.
Constrain request size: <code>post_max_size</code> of 8M without uploads or 32M with uploads, and <code>upload_max_filesize</code> below it	CVE-2026-109618	Also enforce limits at the web server or proxy on decoded request size, input fields, multipart parts, headers, and nesting depth, including requests without a usable Content-Length. Block multipart form data where it is not used.
Remove early diagnostics and reachable fatal errors	CVE-2026-109619	Remove invalid or deprecated per-request settings that emit warnings before initialization, and fix fatal errors and timeouts an attacker can trigger. Suppressing errors with <code>error_reporting</code> , <code>display_errors</code> , or <code>@</code> does not help.
Restrict access or place behind a web application firewall	Both	CISA notes a WAF may provide some protection. Access restrictions through a VPN or IP allowlist are stronger. Neither replaces patching.

Sources

References

1. Security advisory: Critical remote code execution vulnerabilities, update now PHP project, October 21, 2026
2. GHSA-w8hw-w9fx-j79w: RCE via OOM corruption (CVE-2026-109618) PHP project on GitHub
3. GHSA-6wmf-4rv5-p3rp: RCE via stale global state (CVE-2026-109619) PHP project on GitHub
4. Critical Vulnerabilities in PHP Require Immediate Patching CISA, October 21, 2026
5. PHP2Shell Patches Land as First Exploitation Attempts Hit Sensors SecurityWeek, October 21, 2026

About Coretelligent Security Advisory

Coretelligent Security Advisory helps organizations understand, prioritize, and act on cyber risk through virtual CISO leadership, governance and compliance programs, third-party risk management, and incident readiness. If you need help confirming your exposure, prioritizing remediation, or documenting your response for regulators and investors, contact your Coretelligent account team or vCISO.

Visit coretelligent.com | Call 1-855-841-5888



This advisory reflects information available as of October 21, 2026. The situation is evolving, and exploitation activity may change quickly. Consult the referenced sources for the latest guidance. This advisory is provided for informational purposes and does not constitute legal advice.