SECURITY ADVISORY / OCTOBER 22 2026

PHP2Shell puts business websites at risk

Critical flaws in PHP can expose websites, portals and business applications to unauthenticated code execution and sensitive memory disclosure. Confirm where PHP runs, apply the security update, and verify the fix is live. [1, 2]

SEVERITY	EXPOSURE	PRIORITY
Critical 9.2	**No login required**	**Patch PHP now**
CVSS v4.0 for both CVEs. [3]	Attackers may reach vulnerable services remotely. [1]	Update the runtime and confirm every worker is patched. [1, 3]

Why business leaders should act

PHP2Shell describes CVE-2026-109618 and CVE-2026-109619, vulnerabilities in PHP itself. Successful exploitation could expose data, disrupt operations or give an intruder access under the web service's privileges. AI-assisted research helped turn longstanding bugs into working exploits, reinforcing the need to identify exposure and verify remediation promptly. [1, 2]

Where exposure can hide

Websites and portals

WordPress, Drupal, Joomla, online stores, customer portals and patient-facing applications may depend on PHP. [2]

Internal and vendor systems

Include wikis, help desks, custom web tools and products that bundle PHP. Ask each owner or vendor to confirm exposure.

Discover your exposure with a free assessment

Not sure where PHP is running or whether your hosting provider has patched it? SolluCIO offers a free, focused initial exposure assessment to help identify likely gaps and prioritize your next steps.

Request your free assessment

YOUR RESPONSE PLAN

Confirm exposure and close the gap

Start with internet-facing services. Assign an owner to each finding and verify the running PHP service, including deployments managed by a hosting provider.

PHP SECURITY RELEASES / RELEASED OCTOBER 21 2026

PHP 8.5	PHP 8.4	PHP 8.3	PHP 8.2
8.5.12	**8.4.27**	**8.3.36**	**8.2.35**

Use these releases, later security releases, or a vendor-confirmed backport that addresses both CVEs. [1, 3]

01 Find every PHP dependency

Inventory websites, portals, internal applications, containers and vendor products. Confirm the PHP runtime serving web requests; the command-line version may differ.

02 Patch and activate the fix

Update the runtime or vendor package. Restart affected workers, or rebuild and redeploy containers. Confirm every instance uses the patched build. [1, 3]

03 Verify service and security

Validate the running build or backport evidence and test key application functions. Investigate suspicious web files, processes, privileged accounts and outbound connections.

04 Close the ownership gap

Record affected assets, owners, patch evidence and exceptions. Replace unsupported software. Establish patch deadlines and escalation paths with hosting and managed-service providers.

If you cannot patch immediately

Temporary controls reduce specific attack paths but do not make unpatched installations safe. Follow current vendor guidance and escalate systems that cannot be adequately isolated. [1]

Reduce reachable attack paths. Enable relevant WAF protections, restrict access, and limit request bodies and uploads to application needs. [1, 3]

Apply server-level settings carefully. Disable session.upload_progress.enabled and restart workers. On PHP 8.5, disabling fatal_error_backtraces blocks another confirmed path. Neither covers every variant. [1]

Treat appliances separately. Follow product-specific advisories. PAN-OS and Panorama exposure concerns the management web interface. Restrict management access to trusted sources. [4]

PHP 8.1 and earlier lack routine upstream security support; confirm backport coverage or upgrade. PHP 8.2 reaches end of life December 31, 2026. [1]

Discover your exposure

FREE PHP2SHELL EXPOSURE ASSESSMENT

Know where to focus first

Bring your website, hosting or application questions to SolluCIO. Our free initial assessment helps you understand potential exposure and the actions that deserve priority.

Request your free assessment

security@solluciopartners.com

What the free assessment helps you discover

01

Locate exposure

Discuss the web systems and vendor services that may depend on PHP.

02

Review patch status

Review available version information, vendor responses and remediation evidence.

03

Prioritize action

Identify the most urgent gaps and practical next steps for your team.

A focused initial review with scope agreed before work begins. Active testing, remediation and incident response are separately scoped.

Five questions for your hosting provider

1. Which systems you host or supply for us use PHP, and who owns each one?
2. Have both CVEs been fixed, including all running workers, images and replicas?
3. What package, runtime version or vendor advisory proves the fix is applied?
4. For any remaining exposure, what controls are in place and when will patching finish?
5. Have you reviewed exposed systems for suspicious activity, and what did you find?

Keep written responses with your remediation record. Confirming a vulnerability does not by itself establish that a breach occurred; suspicious activity should prompt a separate investigation.

Primary sources

[1] PHP security advisory and fixed releases

[2] Wiz Research on PHP2Shell

[3] Google Cloud response and deployment guidance

[4] Palo Alto Networks product exposure and mitigations

Technical guidance reviewed October 22, 2026. Check the linked advisories for subsequent changes.