



EMERGENCY THREAT ADVISORY

Critical PHP2Shell Vulnerabilities



Services Performed By:

UltraViolet Cyber TIDE Team
tide@uvcyber.com

Published Date:

October 21, 2026

TLP:GREEN



Executive Snapshot

CVE-2026-109618 and CVE-2026-109619, collectively tracked as PHP2Shell, are two critical, unauthenticated remote code execution (RCE) vulnerabilities in PHP core, each rated 9.2 under CVSS v4.0. Technical details leaked publicly on October 20, 2026, two days ahead of a coordinated disclosure originally planned for October 22, and the PHP Foundation shipped an emergency fix the next day, October 21, at 14:00 UTC, in PHP 8.5.12, 8.4.27, 8.3.36, and 8.2.35. Both flaws let an unauthenticated attacker send a single network request and corrupt internal PHP memory state to achieve remote code execution, one through out-of-memory handling and the other through stale request state that survives after it should have been freed. Because the defect lives in PHP core rather than in any one application, every PHP installation serving web requests is in scope, including PHP-FPM, mod_php, php-cgi, FrankenPHP, and LiteSpeed, regardless of the framework or content management system running on top. Cloudflare reported early attack attempts before the patch even released, and PHP's own developers have stated that patching is the only complete fix, since no configuration workaround closes either hole. Organizations running PHP 8.2 through 8.5 below the fixed versions, or any earlier, unsupported branch, should treat this as an emergency patch priority.

- **Patch immediately.** Update every PHP install to 8.5.12, 8.4.27, 8.3.36, or 8.2.35, whichever branch you run, since the fix for both CVE-2026-109618 and CVE-2026-109619 shipped together. Update the runtime itself, such as PHP-FPM, mod_php, or CGI, not just the application layered on top of it.
- **Verify patch coverage before standing down.** Confirm the update against your package changelog or the runtime binary itself, not the application layer, since patching is the only complete fix. Prioritize internet-facing and customer-facing systems first if the rollout must be staged.
- **Inventory and monitor.** Account for every PHP install across the environment, including vendor appliances and control panels such as cPanel and Plesk, and watch for PHP worker processes spawning shells or unexpected network activity.



TIDE Team Analysis

PHP2Shell is the collective name for CVE-2026-109618 and CVE-2026-109619, two critical vulnerabilities in PHP core that leaked publicly on October 20, 2026. The PHP Foundation responded the same evening with advance notice of an emergency release. GitHub Security Advisories for both flaws followed at 13:34 UTC on October 21, 26 minutes ahead of the 14:00 UTC fix. Both vulnerabilities permit unauthenticated RCE, but they are distinct bugs with different root causes.

CVE-2026-109618 arises from out-of-memory handling that can interrupt PHP while internal request data is only partially initialized. Request cleanup may then treat that invalid data as valid, corrupting internal state in a way an attacker can use to achieve remote code execution. **CVE-2026-109619** arises from request-owned internal state that can survive after its backing memory has been freed, so a later request handled by the same worker process can use that stale state before initialization resets it, again resulting in remote code execution. PHP’s advisory does not publish an official CWE classification for either vulnerability.

Both vulnerabilities require no authentication, no special headers, and no targeted URL, and both reach every PHP SAPI that serves web requests, including PHP-FPM, mod_php, php-cgi, FrankenPHP, and LiteSpeed. PHP’s advisory rates the risk highest on mod_php deployments, where a demonstrated exploit can trigger the vulnerable path before any application code runs. A workaround that spread quickly on social media for CVE-2026-109619, disabling Session Upload Progress, was proposed and then retracted by the security firm that first suggested it, and PHP’s own core developers called it insufficient on its own. Per PHP core developer Gina Peter Banyard, the only real mitigation is patching or upgrading PHP, and PHP 8.6 release manager Joe Ferguson added that ini settings will not substitute for the fix.

The fix for both vulnerabilities were released together on October 21, 2026, at 14:00 UTC, in PHP 8.5.12, 8.4.27, 8.3.36, and 8.2.35, the four branches PHP currently supports. Distributions received patches in advance of the public release, so organizations should check their vendor’s advisory for backported packages, confirm against the changelog rather than the version string, and avoid rpms.remirepo.net, which will not carry same-day updates. Notably, PHP 8.1 and all earlier branches are end of life and receive no official fix, while PHP 8.2 itself reaches end of life on December 31, 2026.

VULNERABILITY PROFILE

CVE-2026-109618

Unauthenticated RCE via Out-of-Memory Corruption in PHP Core (PHP2Shell)

CRITICAL v4.0 **9.2**

FIRST EXPLOITED

No confirmed exploitation, though Cloudflare reported early attack attempts before the October 21 patch shipped. **KEV** Not listed.

SOFTWARE

PHP Core by The PHP Foundation

AFFECTED

PHP 8.2, 8.3, 8.4, and 8.5 through their respective fixed releases (8.2.35, 8.3.36, 8.4.27, 8.5.12)

PATCH

8.5.12 / 8.4.27 / 8.3.36 / 8.2.35 Released October 21, 2026, with CVE-2026-109619. Update the runtime itself, not the application layer. [GHSA-w8hw-w9fx-j79w](#)

VULNERABILITY PROFILE

CVE-2026-109619

Unauthenticated RCE via Stale Global State in PHP Core (PHP2Shell)

CRITICAL v4.0 **9.2**

FIRST EXPLOITED

No confirmed exploitation, though Cloudflare reported early attack attempts before the October 21 patch shipped. **KEV** Not listed.

SOFTWARE

PHP Core by The PHP Foundation

AFFECTED

PHP 8.2, 8.3, 8.4, and 8.5 through their respective fixed releases (8.2.35, 8.3.36, 8.4.27, 8.5.12)

PATCH

8.5.12 / 8.4.27 / 8.3.36 / 8.2.35 Released October 21, 2026, with CVE-2026-109618. Update the runtime itself, not the application layer. [GHSA-6wmf-4rv5-p3rp](#)



Why It Matters

PHP2Shell presents a direct risk to shared hosting providers, managed PHP platforms, and vendor appliances that embed PHP without advertising it as a dependency. Because the defect sits in PHP core rather than in a specific application, organizations that rely on a hosting provider, a control panel such as cPanel or Plesk, or a vendor-supplied appliance cannot remediate it by patching their own application code; they depend on that provider or vendor to ship the underlying PHP update. Teams that only track the CMS or framework version in their asset inventory, rather than the PHP runtime underneath it, risk missing this exposure entirely.

Since both vulnerabilities require no authentication and reach PHP before any application-level access control runs, a threat actor only requires network access to a PHP-based web service to attempt exploitation. No stolen credentials, no social engineering, and no user interaction are required. Successful exploitation grants remote code execution on the underlying server, which a threat actor could use to install a web shell, move further into the network, exfiltrate data, or deploy payloads, depending on what that server can reach. Cloudflare reported early attack attempts before the patch even released, which shows how quickly adversaries move once technical details are available, and the two-day compression of the disclosure timeline, from the October 20 leak to the October 22 planned embargo, left defenders less time to prepare than the PHP Foundation had intended.

PHP2Shell follows a pattern seen in other critical vulnerabilities that live in a runtime or library embedded across many applications, rather than in any single product. The infamous Log4Shell (CVE-2021-44228) exploit, disclosed in December 2021, carried a maximum severity score and reached countless applications that embedded the Log4j logging library as an indirect dependency, often without the application owner's knowledge, and many organizations spent months afterward just locating every place Log4j was buried in their own software. The XZ Utils backdoor (CVE-2024-3094), discovered in March 2024, pushed the same lesson further: a deliberately planted backdoor in the liblzma compression library, reached indirectly by OpenSSH through a chain of build dependencies on several Linux distributions, carried a maximum CVSS score of 10.0 and was caught only weeks before it would have reached the stable release branches most production systems run. Unlike PHP2Shell, that backdoor was intentional rather than accidental, but it proved the same point: a defect several layers down in the dependency chain can sit invisible to standard asset inventories and vulnerability scanners until someone examines the runtime itself, not just the application layered on top of it. PHP2Shell reinforces that lesson at the language level, since PHP core underlies the application rather than sitting beside it: when the defect lives in a runtime every application depends on, the visible asset inventory is not enough, and the vendor patch, not an informal workaround, is the response that holds up under real pressure.



How to Respond

- **Patch immediately.** Update every PHP install to the fixed release for your branch, since the October 21, 2026 release fixes both CVE-2026-109618 and CVE-2026-109619 together.
 - Identify every PHP install in the environment, including vendor appliances, control panels such as cPanel and Plesk, and PHP-based applications, so nothing is missed.
 - Confirm the current branch and version on each host, then apply 8.5.12, 8.4.27, 8.3.36, or 8.2.35, whichever matches that branch.
 - Update the PHP runtime itself, such as PHP-FPM, mod_php, or CGI, not just the application layered on top of it, and verify the fix against your package changelog rather than the version string alone.
 - Prioritize internet-facing and customer-facing systems first if the rollout must be staged.
 - For PHP 8.1 and earlier, which are end of life and no longer receive official updates, upgrade to a supported branch or apply an available vendor backport without delay.
- Perform annual technology refresh reviews to gain clear, complete visibility into your infrastructure. Speak with your UltraViolet Cyber TAM Representative to schedule a Red Team or Purple Team engagement to gain insight into the vulnerabilities in your environment.

What UltraViolet Cyber is Doing

- Tracking new CVEs and high impact vulnerabilities, analyzing and deploying public Proof-Of-Concept code against custom built targets.
- Proactively enabling custom detections based on the intelligence available for the PHP2Shell vulnerabilities. UltraViolet is also maximizing detection coverage proactively through [Equinox](#).
- Performing hypothesis driven threat hunts based on the intelligence available for the PHP2Shell vulnerabilities. UVCyber customers will be informed of the results through secure channels.
- Parsing available victim dump data for any social, financial, business, or technical relations to UVCyber Clients and partner organizations.
- Aggregating threat intelligence from myriad sources and applying the most up-to-date knowledge to proactive threat hunting and response.



About UltraViolet Cyber

UltraViolet Cyber is a leading tech-enabled managed security services provider, delivering unparalleled cybersecurity expertise that fills technology and talent gaps across Global 2000 and Federal Government customers. Founded and operated by security practitioners from the national intelligence community, UltraViolet Cyber connects offensive security, application security, detection and response, and security engineering to deliver a differentiated approach to cybersecurity operations. Transforming customers' security programs, UltraViolet Cyber's flagship security-as-a-service solution, UV Lens, removes complex operational silos, replacing them with integrated security capabilities. UltraViolet is headquartered in McLean, Virginia with technology centers across the world.

443.351.7630 / info@uvcyber.com |  UltraViolet Cyber |   @uv_cyber
