



PHP2Shell: Unauthenticated Remote Code Execution in PHP Core

October 21st, 2026

Vulnerabilities	CVE-2026-109618: out-of-memory (OOM) handling interrupts PHP while request data is only partly initialized, and request cleanup then treats the invalid data as valid. CVE-2026-109619: request state survives after its memory is freed, and a later request on the same worker uses that stale state before PHP resets it. Both allow unauthenticated remote code execution (RCE) and score 9.2 on the Common Vulnerability Scoring System (CVSS) v4.0 scale. Researchers call the pair PHP2Shell.
Vendor Advisory	PHP published a security advisory on October 21, 2026, a day ahead of the planned embargo date, after details of the flaws leaked on October 20. PHP 8.5.12, 8.4.27, 8.3.36, and 8.2.35 fix both flaws. PHP 8.1 and earlier received no official release. PHP also published minimal source backports for 7.4, 8.0, and 8.1 for organizations that build from source or lack a long-term support (LTS) vendor.
Affected Products	PHP installations that serve web requests, including PHP-FPM, Apache with mod_php, php-cgi, FrankenPHP, and LiteSpeed. PHP rates mod_php highest, where a demonstrated exploit triggers before any application code runs. Wiz reports that every PHP version back to 5.4 has at least one of the flaws. WordPress, Drupal, Joomla, and Magento sites inherit the exposure because the flaws sit in PHP itself.
Exploitation Status	Exploit demonstrated. In-the-wild activity unconfirmed. Wiz built a working exploit chain that achieved unauthenticated RCE on default WordPress, Drupal, and Joomla container images.

Executive Summary

PHP [released](#) emergency fixes on October 21, 2026 for two critical flaws, **CVE-2026-109618** and **CVE-2026-109619**, that let an unauthenticated attacker run code on a web server. PHP states that a request can trigger the attack before any application code runs, so secure application code does not protect an affected server. Additionally, Wiz has demonstrated working exploits that achieved unauthenticated RCE on default WordPress, Drupal, and Joomla container images.

Exposure is broad because the flaws sit in the PHP engine and reach any underlying application. [Mallory](#) counted about 884,000 internet-facing hosts that reported PHP 8.1 or earlier, which is 59% of the 1.5 million hosts that reported a version. The counts come from Shodan banners and do not confirm

vulnerable systems. PHP released no fix for those branches, so SRA recommends treating them as exposed unless a distribution or LTS vendor confirms a backport.

A successful attack runs code as the web worker account. A separate memory disclosure path can expose passwords, tokens, API keys, and certificates without code execution. Wiz states that AI models made the exploit chain practical to build and adapt across targets in hours.

Organizations should immediately patch internet-facing systems to PHP 8.5.12, 8.4.27, 8.3.36, or 8.2.35 first, then rotate secrets on any host that faced the internet.

Technical Background

Wiz's write-up [describes](#) four exploitable bug variants grouped under the two CVEs. All four share the same root cause: a fatal error, most commonly an out-of-memory condition that an attacker can trigger with an oversized or malformed request, interrupts PHP after it has published internal state but before it has finished initializing or cleaning up. PHP destroys the request heap, the worker survives, and the next request on that worker trusts pointers that now reference freed and attacker-reclaimable memory.

- **Upload-progress unserializer use-after-free:** PHP's session module can decode session data during upload-progress handling before normal module initialization completes. A fatal allocation failure leaves the unserializer's global state published. A later request on the same worker consumes attacker-reclaimed memory as live unserializer bookkeeping, producing a memory-corruption primitive. Wiz combined this with the memory disclosure to achieve unauthenticated RCE on WordPress, Drupal, and Joomla.
- **Remote process-memory disclosure:** PHP parses multipart requests during request startup. Session upload-progress handling can publish a session ID early, and an out-of-memory error can abort the request before cleanup. A later multipart request on the same worker trusts the stale pointer as a live string and URL-encodes the reclaimed heap contents into a PHPSESSID response cookie. This works even when the application never calls `session_start()`, and Wiz demonstrated disclosure of an attacker-sized region of worker memory containing data from outside the attacker's own request.
- **Fatal-error backtrace use-after-free:** PHP retains the backtrace of the most recent fatal error in an executor global. An out-of-memory fatal can destroy the request heap without clearing that value. If the same worker emits a diagnostic early in the next request, it dereferences the dangling backtrace and follows attacker-reclaimed memory. Wiz used this for unauthenticated RCE against Drupal.
- **OPcache type confusion:** OPcache persists compiled structures in shared memory and coordinates runtime-pointer slots across workers. An allocation failure between advancing the process-local frontier and publishing the shared frontier leaves partially persisted objects referencing slots other workers still consider free. When another worker reuses a slot, stale metadata resolves to a pointer of the wrong type. Wiz used this cross-worker confusion for unauthenticated RCE against WordPress.

Researchers note that other tracked and likely untracked variants of this pattern exist in the PHP codebase, and that the PHP fix addresses the problem at its root (the changelog entry reads *Avoid state corruption via bailout*) rather than patching individual variants.

Detection Considerations

These are not fully developed detections, but rather the types of activities one would likely build detections around. As our research and testing progresses, we plan to post more details around detection and prevention capabilities.

- **Web worker process ancestry (SRA):** A web server or PHP worker appearing as the parent or grandparent of a command shell, scripting interpreter, or network utility whose command line matches post-exploitation activity. SRA deployed a detection built on this pattern to Defender, CrowdStrike NG-SIEM, and SentinelOne clients to alert on activity consistent with exploitation while vendor indicators remain unpublished. It requires an EDR agent on the affected servers, so hosts without one are not covered.
- **Exploitation attempts (pre-patch, Wiz):** Unusual multipart requests followed by memory-limit errors or repeated HTTP 500 responses, unexpectedly large or malformed PHPSESSID response cookies (the memory disclosure path returns heap contents in that cookie), and unexplained PHP worker churn in FPM or Apache logs.
- **Post-exploitation behavior (Huntress, Florian Roth):** PHP worker processes (php-fpm, php-cgi, php-cgi.exe, httpd, apache2) spawning shells (sh, bash, cmd.exe, powershell.exe), interpreters (python, perl), or download and networking tools (curl, wget, nc, certutil.exe). Command lines containing reverse-shell patterns such as /dev/tcp/, fsockopen, or mkfifo, and quick reconnaissance commands (whoami, id, uname -a, ipconfig) run directly from the web server process. Huntress notes this matches nearly every PHP intrusion it works.
- **Persistence artifacts:** Web shells and new files in web roots, new cron jobs or scheduled tasks, and new local accounts on web hosts.
- **Network coverage:** Cisco Talos published Snort rules 67359 and 67360, available in the free Community Ruleset. Cloudflare, Akamai, AWS, Google Cloud, Imperva, and Fastly prepared WAF mitigations.

Safeguards/Recommendations

- **Patch:** Upgrade to PHP 8.5.12, 8.4.27, 8.3.36, or 8.2.35, starting with internet-facing systems. Restart PHP-FPM or the web server so workers load the patched build. Linux distributions and LTS vendors received the patches in advance, so check vendor packages. Vendors may backport fixes without changing the upstream version number, so verify patch status against the vendor advisory. Rebuild container images from patched base images. Organizations that use the volunteer-maintained rpms.remirepo.net repository will not receive updates from it today and should switch to distribution packages or build from source.
- **Inventory PHP:** Find every PHP installation, including PHP bundled in vendor appliances, Windows software, and control panels such as cPanel and Plesk. Record the version and server interface (PHP-FPM, mod_php, or php-cgi) for each host, since PHP rates mod_php highest.
- **Handle Unsupported Versions:** PHP 8.1 and earlier receive no official release. Move to a supported branch, or ask the distribution or LTS vendor for a backport. Organizations that build from source can use PHP's minimal backports for 7.4, 8.0, and 8.1. Until then, restrict access with a VPN or IP allow list, place the host behind a web application firewall (WAF), or take it

offline. Include PHP 8.2 in migration planning, since it reaches end of life on December 31, 2026.

- **Apply Interim Mitigations If Patching Is Delayed:** PHP states that these settings reduce exposure to specific attack paths and do not make an installation safe. Restart every PHP worker after each change, and set values in `php.ini`, the FPM pool, or server configuration, not through `ini_set()` or `.user.ini`.
 - CVE-2026-109618: Set `memory_limit=-1` in a way applications cannot override, such as `php_admin_value`, and confirm no pool, directory, or application setting restores a finite limit. Set `post_max_size` to 8M for applications that do not need uploads and 32M for those that do, and keep `upload_max_filesize` below it. Enforce request size, field count, multipart part count, and header limits at the web server or proxy as well. Use container or cgroup limits as an availability boundary, and do not use `RLIMIT_AS` or `RLIMIT_DATA`, which can trigger the vulnerable allocation failure.
 - CVE-2026-109619: Set `session.upload_progress.enabled=0`. On PHP 8.5, also set `fatal_error_backtraces=0`. Remove invalid or deprecated per-request settings that emit diagnostics before initialization, and fix attacker-reachable fatal errors and execution timeouts. Hiding diagnostics with `error_reporting`, `display_errors`, or `@` does not help.
- **Enable CDN and WAF Protections:** Check whether your content delivery network or WAF vendor published PHP2Shell rules and enable them. WAF rules may compensate for a missing patch but should not replace it.
- **Respond to Suspected Compromise:** Isolate the server and preserve web server and PHP logs. Look for web shells and new files in the web root, new cron jobs and scheduled tasks, and new accounts.
- **Rotate Secrets After Patching:** For internet-facing PHP applications that ran an unpatched version, rotate database passwords, API keys, session secrets, and TLS private keys, starting with the systems that handle the most sensitive data. Memory disclosure may leave little in application logs, so a clean log does not rule out exposure.
- **Hunt for Exploitation:** Review web server and FPM logs for the request and error patterns under Detection Considerations, and PHP worker processes spawning shells, interpreters, or download tools. On Linux hosts without EDR, Florian Roth's free [process-watcher](#) provides this visibility quickly.

References

- [PHP: Security advisory, Critical remote code execution vulnerabilities, update now \(October 21, 2026\)](#)
- [PHP: Upcoming Critical PHP Security Release \(October 20, 2026\)](#)
- [GitHub Security Advisory GHSA-w8hw-w9fx-j79w \(CVE-2026-109618\)](#)
- [GitHub Security Advisory GHSA-6wmf-4rv5-p3rp \(CVE-2026-109619\)](#)
- [PHP: Supported Versions](#)
- [PHP2Shell: AI uncovers critical RCE vulnerabilities affecting 70% of the Internet, Wiz Research](#)
- [Rapid Response: php2shell, Leaked Critical PHP Vulnerabilities, Huntress](#)
- [PHP RCE in 8.5 and Earlier: CVE-2026-109618 and CVE-2026-109619, Mallory](#)
- [PHP2Shell: All You Need to Know About CVE-2026-109618 and CVE-2026-109619, JFrog Security Research](#)
- [PHP2Shell: Two Unpatched Critical PHP RCE Flaws, The Hacker News](#)
- [process-watcher PHP RCE quick start, Florian Roth](#)