

A critical flaw in PHP lets attackers take over websites. Here's what that means for you.

If you run a website, online shop or web portal, it very likely uses PHP and needs updating now.

Technical detail for your IT team, web agency or hosting provider is on page 2.

WHAT HAPPENED

The software behind most websites has two flaws

PHP is the software that runs around 70% of websites, including WordPress, WooCommerce, Magento, Drupal and Joomla, plus many customer portals, webmail and admin tools that you may not think of as websites.

On 21 October the PHP team released emergency fixes for two critical flaws, nicknamed PHP2Shell. Details leaked a day early, and attackers began probing for them within hours.

WHY IT MATTERS

One web request, no password needed

Someone exploiting these flaws could:

- **Take control of the web server**, run their own commands, plant backdoors or add card-skimming code to an online shop.
- **Read secrets from the server's memory**, such as passwords, login sessions and API keys, even without taking full control.

Your website's own code doesn't need to be faulty. A fully updated WordPress site is still exposed until the PHP underneath it is updated.

WHERE THINGS STAND

Fixes exist, but not for old versions

Fixes are out for PHP 8.2 to 8.5. **PHP 8.1 and older will not get an official fix**, so those systems need upgrading. Cloudflare and other web protection services have added temporary blocking rules, but these only help sites whose traffic passes through them.

WHERE TO CHECK

Systems that may run PHP

- ☐ **Your websites and online shops**, including ones built or hosted by an agency.
- ☐ **Customer and staff portals**, intranets, webmail, file sharing and billing systems.
- ☐ **Admin tools** such as phpMyAdmin and hosting control panels.
- ☐ **Appliances and software products** with PHP built in. Check with the vendor.
- ☐ **Old, forgotten, test and staging sites**. These are often the ones that get missed.
- ☐ **Server templates and container images** used to build new sites.
- ☐ **Suppliers and SaaS providers** that host services or data for you.

WHAT TO DO

Recommended steps

- 1 **Find every system running PHP**, starting with anything reachable from the internet.
- 2 **Update PHP itself on every server**, then restart the web services. Updating WordPress or plugins is not enough.
- 3 **Using an MSP, web agency or hosting company? Ask them today** whether PHP is updated on your sites, and by when. Page 2 is their checklist.
- 4 **Can't update straight away?** Restrict access to the site or take it offline until it's fixed.
- 5 **Check for signs of compromise** on anything that was exposed before it was patched.

Running PHP 8.1 or older? There is no official fix coming. Plan the upgrade now rather than waiting.

How Barefoot Cyber can help

For managed customers, we are checking the systems we look after for PHP and watching for signs of attack. If you'd like help finding where PHP runs in your business, reply to this email.

1 The vulnerabilities

CVE · CVSS v4	Root cause and impact
CVE-2026-109618 9.2 Critical	Out-of-memory handling interrupts PHP while request data is partly initialised; cleanup then treats it as valid. Unauthenticated RCE as the web worker, plus remote memory disclosure via the <code>PHPSESSID</code> cookie, even if the app never calls <code>session_start()</code> .
CVE-2026-109619 9.2 Critical	Request-owned state survives after its memory is freed; the next request on the same worker uses it. Fatal-error backtrace use-after-free plus OPcache type confusion. Unauthenticated RCE.

Affected: every SAPI serving web requests: PHP-FPM, Apache mod_php, php-cgi, FrankenPHP, LiteSpeed. Code dates to PHP 5.4.0; default configs are vulnerable. **mod_php** is **highest risk**: a single unauthenticated request exploits it before any application code runs.

Status: exploitation attempts reported by Cloudflare (from 07:43 UTC on 21 Oct, before the patch) and watchTower. Not on CISA KEV at time of writing.

2 Fixed versions

Branch	Fix
8.5	8.5.12
8.4	8.4.27
8.3	8.3.36
8.2	8.2.35 (EOL 31 Dec 2026)
8.1, 8.0, 7.4	No official release. Distro/LTS backports or source patches only
7.3 and older	No fix. Upgrade

Distro packages keep older-looking version strings. **Check the changelog for the CVE IDs**, not the version number.

3 Fix by platform

Linux servers

Update the PHP packages from your distribution or LTS vendor, then confirm: `apt changelog php8.3-fpm | grep 109618` or `rpm -q --changelog php-fpm | grep 109618`. Restart PHP-FPM, Apache and any long-running PHP workers. Remi's repository had not shipped fixes on 21 Oct; switch to distro packages or build from source.

Windows and IIS

Replace PHP with the patched build for your branch and recycle the IIS application pools. Check bundled stacks such as XAMPP and WAMP, and any vendor application that ships its own `php-cgi.exe`.

Hosting panels and shared hosting

Panels often run several PHP versions side by side and assign one per site. Update every installed version, not just the default, then check which version each site uses. On managed hosting, get written confirmation from the provider.

Containers and images

Rebuild on patched `php` base images (and patched `wordpress`, `drupal` and similar images) and redeploy. Search your registries for older PHP base images. Patching the host doesn't change images.

Appliances and third-party software

Many products bundle PHP in web consoles and portals. Track vendor advisories, using NCSC-NL's inventory as a starting point: github.com/NCSC-NL/php2shell. If a vendor has no fix, restrict access to the management interface.

Can't patch yet? **STOPGAP ONLY**

PHP says patching is the only complete fix. Meanwhile: put the site behind a WAF with PHP2Shell rules, or restrict it to VPN or allow-listed IPs, or take it offline. Partial config mitigations:

```
session.upload_progress.enabled=0 (plus
fatal_error_backtraces=0 on 8.5), a low post_max_size and
request-size limits at the web server. Don't use
RLIMIT_AS / RLIMIT_DATA; they can trigger the flaw.
```

4 Verify the fix

Check the version the web server actually runs, not the command line: `php-fpm -v`, or a temporary `phpinfo()` page that you delete afterwards. Confirm every worker restarted after the update.

After patching, review `memory_limit` and `post_max_size`. The fix adds a hard memory cap that ends the process when reached.

5 Hunt and rotate

Look for PHP processes (`php-fpm`, `php-cgi`, `httpd`, `apache2`) spawning shells, `curl`, `wget`, `nc` or `certutil`; multipart requests followed by memory errors or repeated 500s; oversized `PHPSESSID` cookies; new files in the web root, cron jobs or accounts.

If a server was exposed unpatched, rotate the database passwords, API keys and session secrets it held in memory.

6 MSP, agency or host?

Send them this page and ask them to confirm in writing:

- Every site and server running PHP for you, with versions
- PHP updated (or backport with the CVEs) and workers restarted
- An upgrade date for anything on PHP 8.1 or older
- Logs reviewed from 20 Oct, and secrets rotated where exposed
- Completion date and evidence