



PHP Is Not Out of the Woods Yet After PHP2Shell

A researcher claims six more unpatched PHP flaws after PHP2Shell. Unconfirmed, and nothing to do today, but PHP is not out of the woods. What to prepare now.

Phil E. Taylor | 25 October 2026

On 24 October 2026, three days after [the PHP project's security release](#) fixed the two PHP2Shell flaws, a researcher published a GitHub repository titled [php2shell-zeros](#) claiming six further unpatched vulnerabilities in fully patched PHP, with exploit scripts, and RedEye Security wrote them up the same day as [PHP2Shell, Part Two](#). None of the six is confirmed by the PHP project, and there is nothing a Joomla or WordPress site owner can do about them today. They are the reason PHP is not out of the woods yet, and the reason to spend this week preparing rather than relaxing.

The ground under you is solid. The two critical flaws Wiz named PHP2Shell, [CVE-2026-109618 and CVE-2026-109619](#), are fixed in PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35, CISA has not observed a successful exploitation, and no one else has confirmed a compromise either. If your host applied that release, there is nothing new for you to do today. This post takes the new claims first, then the two smaller items that are also still open, then the boring preparation that turns the next release into a quiet afternoon rather than a bad week.

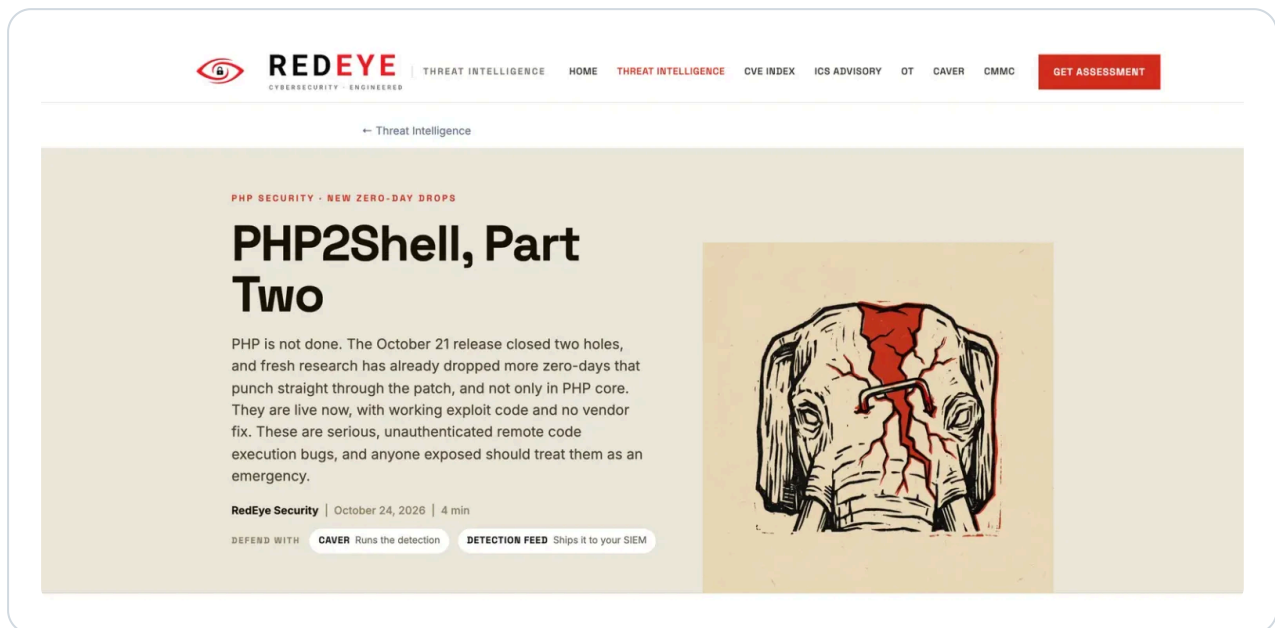
What to do today

If your host applied the 21 October release, nothing new. If you have not yet set `session.upload_progress.enabled=0`, do that now: [the earlier post](#) has the one-line change and where it goes. Otherwise, prepare. At least one more PHP release with a security fix is coming, and the section on preparing below is the work to do before it arrives.

Today's News: A Second Wave Is Being Claimed

The new news is the write-up itself. RedEye Security's [PHP2Shell, Part Two](#) opens with "PHP is not done" and describes the new findings as live, with working exploit code and no vendor fix. So far it is one researcher's repository and one security vendor's write-up, and no one else has verified any of it. If the PHP project confirms even one of the

six, this becomes a thing, and the week after a critical release is exactly when that tends to happen.



RedEye Security's Part Two write-up, 24 October 2026, as it looked on 25 October.

Nothing has yet been confirmed by the PHP Foundation or the PHP project

As of 25 October 2026 there is no CVE, no advisory and no public statement from the PHP project on any of the six claims. Until there is, treat them as claims.

What the “Part Two” Claims Say, and What Has Been Confirmed

RedEye’s write-up, updated on 25 October, presents the repository’s six claimed vulnerabilities as unpatched flaws in fully patched PHP. Among them: remote code execution against a patched PHP 8.5.12, code execution on a live WordPress host, root on a stock Ubuntu 22.04 LTS server, a webshell on stock Windows, and a takeover of the NativePHP bridge. RedEye scores three of them at CVSS 9.8 itself and says it has escalated them to CISA.

What has been confirmed is less. None of the six has a CVE. None has been confirmed by the PHP project, by Canonical, or by any independent party. CISA's alert and its catalogue do not mention them. The one with real substance behind it, the NativePHP item, is a flaw in NativePHP's own code, and NativePHP is a framework for building desktop and mobile apps with Laravel, which a Joomla or WordPress website does not run. Set side by side, claim against record, it looks like this.

What the repository README said

“ Remote code execution against a fully patched PHP 8.5.12”

The facts (on 25 Oct 2026, 22:00 UTC)

No CVE yet, and no confirmation from the PHP project yet. The closest independent signal is issue #24481 on PHP's own tracker, which is an untriaged crash report, not code execution.

The source

[php-src issue #24481](#)

What the repository README said

“ Root on a stock Ubuntu 22.04 LTS server”

The facts (on 25 Oct 2026, 22:00 UTC)

Ubuntu's USN-8948-1 covers the two known CVEs. No USN and no CVE exists for this path, and Canonical has not confirmed it yet.

The source

[Ubuntu USN-8948-1](#)

What the repository README said

“ Takeover of the NativePHP bridge”

The facts (on 25 Oct 2026, 22:00 UTC)

Real, and fixed on 25 October in nativephp/mobile 4.6.1 and nativephp/mobile-file 1.0.3. The flaw is in NativePHP's own code, not in PHP, and no Joomla or WordPress site runs NativePHP.

The source

[NativePHP advisory GHSA-7927-vv5p-fv4f](#)

What the repository README said

“ Escalated to CISA,
with three of them
scored at CVSS 9.8”

The facts (on 25 Oct 2026, 22:00 UTC)

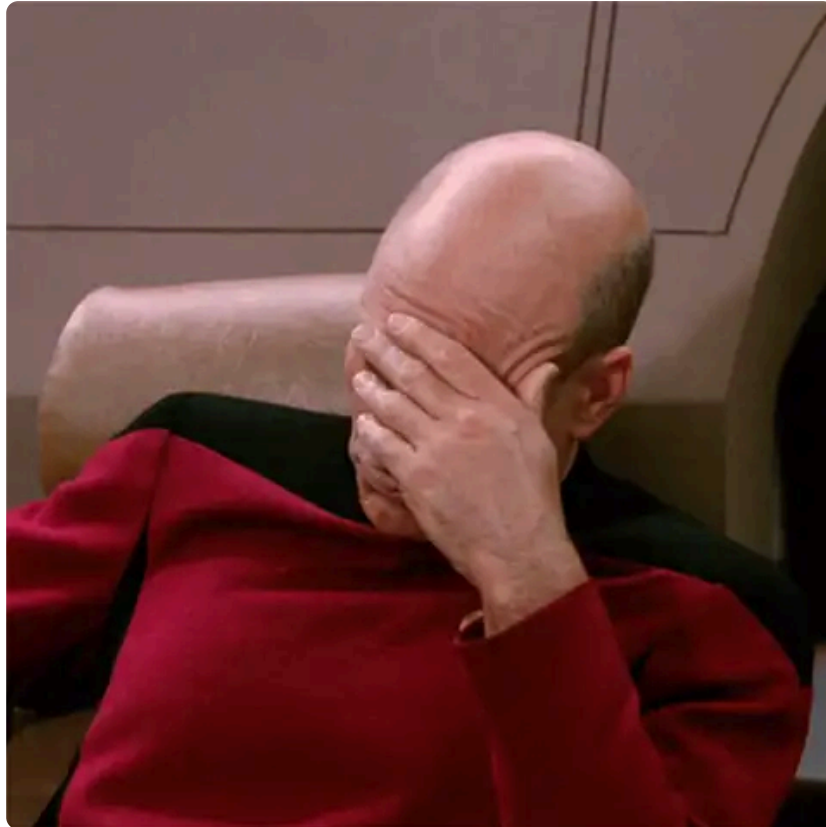
The scores are RedEye's own. CISA's alert names only the two fixed CVEs, and its Known Exploited Vulnerabilities catalogue holds neither of those, let alone any of these.

The source

CISA alert, 21 October, updated 23 October

I am not telling you the claims are wrong. No one can say that yet either, for the reason in the next section. They may well be correct, and if so the PHP project will, in time, respond to them. We will keep an eye on that.

Publishing the Exploit First and Emailing the PHP Project Afterwards



Facepalm. Star Trek: The Next Generation, "Deja Q" (1990).

The repository's own commit history gives the order of events. It was created on 24 October at 05:43 UTC and its first commit, with the exploit scripts, went in at 05:46 UTC. The "Disclosure" section of its README at that point read: "Unfiled at publication time. The choice is the author's." At 21:46 UTC the same day, about 16 hours after publication, that line was changed to: "Findings emailed to `sec@php.net` after public release. Awaiting response." RedEye's article describes its own role as coordinated vulnerability disclosure.

Coordinated disclosure means the vendor hears first, gets time to confirm and fix, and releases the fix before the details go public, so that by the time you read about a flaw there is an update to apply. Here the exploit code was public for most of a day before anyone emailed the people who maintain PHP, and there is no sign the PHP project was told before that. No one at PHP had a chance to confirm, fix or dispute any of it before the scripts were out.

For a site owner that has a practical consequence. There is nothing to verify and nothing to act on, because the party that would normally verify it has only just been asked. A vendor that has not been told cannot release a fix, and a vendor told after the fact starts from the same position as the rest of us. That is not a fix you are waiting for. It is an assessment you are waiting for. As of 25 October no public response from the PHP project has been found.

What Else Is Still Open in PHP After the 21 October Release

Away from the new claims, two smaller items are open. The confirmed one is CVE-2026-115560. On 23 October the PHP project posted [an informational advisory on php.net news](#) which says: "We suggest you set `session.upload_progress.enabled=0` until the next regular bugfix release, to fully protect from an unlikely RCE vector." It adds: "Since this setup is unlikely in practice, we are handling it in the regular release cycle and will publish a full advisory for CVE-2026-115560 with the next release."

The setup it refers to is specific. An application has to set `session.serialize_handler` to `php_binary` or `php_serialize` in configuration, then switch it to `php` at runtime with `ini_set()` before `session_start()`, with upload progress still enabled. Stock Joomla and stock WordPress do not do that. The CVE ID is reserved but no record is published, so there is no score and no affected-version list yet. PHP puts out bugfix releases roughly every four weeks, which puts the next one in mid-to-late November 2026, though no date has been announced.

The unconfirmed one is [issue #24481](#) on PHP's own bug tracker, filed by a community member on 21 October. It reports a use-after-free, which is memory being used after PHP has already released it, in `var_destroy()` during `unserialize()` re-entrancy. `unserialize()` is the function that turns stored data back into PHP values, so this only comes into play where application code unserializes data an attacker controls. A second community member reported on 22 October that it reproduces on stock 8.5.12, 8.4.27, 8.3.36 and 8.2.35 under Valgrind, which is to say on the patched builds.

As of 25 October it is untriaged: no severity, no CVE, no maintainer response, and no one has shown it producing code execution. A crash report on a bug tracker is how a lot of fixes start, and most of them end as a bug fix with no advisory. It is worth knowing about and not worth losing sleep over.

Why There Is Nothing for a Joomla or WordPress Site Owner to Do Today

The fix for PHP2Shell lives in PHP, and PHP is installed and updated by your host. [WordPress.org's notice](#) says no WordPress update is needed, and the same is true for Joomla: there is no Joomla release for this. On shared or managed hosting you cannot update PHP yourself, so the question was always whether the host had applied the release. It did not help that [joomla.org was offline](#) that morning for anyone trying to check the Joomla side of things.

Beyond `session.upload_progress.enabled=0`, there is no setting to change. The PHP project's own line on the two critical flaws is that "there are no complete mitigations, updating is the only safe choice", and that update is out. [CISA's alert](#), published on 21 October and updated on 23 October with a Sigma detection rule, covers only the two fixed CVEs and states that CISA "has not observed successful exploitation of these vulnerabilities in the wild as of the time of publication". Neither CVE is in CISA's [Known Exploited Vulnerabilities catalogue](#) as of 25 October. Honeypot and firewall operators report large volumes of attempts and no confirmed compromise.

One point still catches people out. PHP 8.1 and older are [end of life](#) and got no official fixed release, but distributions back-ported the fix without changing the version string. [Ubuntu's USN-8948-1](#) patched 22.04 LTS at php 8.1.2-1ubuntu2.28 and 24.04 LTS at 8.3.6-0ubuntu0.24.04.13, so `php -v` on a patched Ubuntu box still prints 8.1.2 or 8.3.6. If your host says it is patched and the version number has not moved, that can be true, and [our agency post](#) covers how to confirm it with them.

How Do You Prepare for the Next PHP Security Release?

You prepare by doing the inventory now, while nothing is on fire: which PHP version each site runs, who applies PHP updates at each host, and whether your last backup restores. None of that is technical, and all of it is what made 21 October a long day for the people who did not have it written down.

Know the PHP version of every site

On 21 October the first question every agency had to answer was which sites were on 8.2 to 8.5 and would get a release, and which were on 8.1 or older and would not. If you look after a dozen sites across four hosts, that is a dozen logins to find out. mySites.guru shows the PHP version of every connected site in one list, through the PHP Version Should Be Latest Supported Series check, so when the next release is out you already know which sites it touches.

PHP Version Should Be Latest Supported Series

mySites.guru checks every connected site for this automatically and flags it the moment it appears. It runs as part of the full audit on every connected site.

Check your site for free →

Know who applies PHP updates at each host, and how to reach them

For every host you use, write down three things: whether PHP updates go on automatically or need you to pick the version in a control panel, where the host publishes its own status on security releases, and the support route that gets a human answer. On 21 October the slow part was finding out whether a given host had applied the release, and that is a question you can answer per host in advance, before anyone is in a hurry.

Keep backups you have restored at least once

A backup you have never restored is a hope. Before the next release, restore one backup per site, or at least one per host, somewhere harmless and confirm the site comes up. mySites.guru [schedules and runs backups](#) through Akeeba Backup Professional on Joomla and All-In-One WP Migration on WordPress, and sends a daily missed-backup alert, so a backup job that stopped weeks ago with no error shows up the next morning rather than on the day you need it.

Leave session upload progress off

If you set `session.upload_progress.enabled=0` after the 23 October advisory, leave it that way until the next release and the full advisory for CVE-2026-115560 are out. mySites.guru records the value of that setting on every connected site at [each snapshot](#), twice a day, through the PHP Session Upload Progress Setting check. It is read-only: it reports the setting and does not change it, so a site that drifts back to the default shows up without anyone going looking.

Decide now what you do with sites on PHP 8.1 or older

When the 21 October release went out, 22.7% of the live sites under active management in mySites.guru were on PHP 8.1 or older, where no official release exists. Each of those sites depended on its distribution or host back-porting the fix, and the next PHP security release will put them in the same position again. Deciding now which of them move to 8.3 or newer, and which stay on a vendor back-port on purpose, is the difference between a plan and a scramble.

Where Should You Watch for the Next PHP Release?

Watch the PHP project's own two pages first. The php.net news page, where the 23 October advisory appeared and which is linked above, is where the release announcement will go, and the [php-src security advisories page](#) is where the full advisory for CVE-2026-115560 will be published alongside it.

mySites.guru covers the same ground for people who would rather be told. Security alerts go out on this blog, by [email newsletter](#), which covers security alerts, new features and tips, and on our [social accounts](#). Pick whichever one you read.

What the Last Four Days Looked Like From Here

After 15 years of looking after other people's Joomla and WordPress sites, the pattern after a big release is familiar: a quiet week, then a smaller follow-up release. This one already has its follow-up pencilled in, with CVE-2026-115560 due in the next regular release, and possibly more if issue #24481 turns out to matter or any of the Part Two claims stand up once the PHP project has looked at them.

The people who had a bad time on 21 October were the ones who did not know which PHP version each site ran, who applied updates at each host, or whether their last backup restored. The hard part of that day for agencies was the inventory and the host chasing, not the patch. Both of those are things you can finish this week, with nothing on fire, and then the next release is something you read about rather than something that happens to you.

What the Next Four Days Look Like From Here



Waiting. Mr Bean, as ever.

We will be watching three places: the php.net news page for a statement or a release, the php-src security advisories page for a new advisory, and issue #24481 and the repository for a maintainer response. If the PHP project confirms any of the six claims, or publishes a fix, this post gets a dated update at the top and a security alert goes out by newsletter and on social the same day. If the claims are disputed or go quiet, the update says that instead. Either way the facts column above is rechecked and redated.

Nothing changes on the mySites.guru side in the meantime, because nothing needs to. Snapshots keep recording the PHP version and the session upload progress setting of every connected site twice a day, the backup schedules keep running, and the missed-backup alert keeps watching them. When the next release does arrive, the list of sites it touches is already there.

If you would rather have that inventory kept for you, it is part of the [mySites.guru subscription](#): the PHP version of every site in one list, the upload progress check, scheduled backups with the missed-backup alert, and a security alert when the next

release is out. To see what it reports for one of your sites first, a [free audit](#) runs the same checks.

Timeline

-
- A vertical timeline with a central line and circular markers. The markers are solid black for major releases and alerts, and hollow for other events. The text is aligned to the right of the markers.
- 20 Oct 2026** ● **PHP pre-announces a critical security release**
Two vulnerabilities rated CVSS 4.0 9.2, release set for 21 October at 14:00 UTC, details withheld.
 - 21 Oct 2026, 14:00 UTC** ● **PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35 are released**
CVE-2026-109618 and CVE-2026-109619 are fixed, both unauthenticated remote code execution. The pair is named PHP2Shell.
 - 21 Oct 2026** ● **CISA alert published, Ubuntu USN-8948-1 released**
CISA says it has not observed successful exploitation. Ubuntu back-ports the fix to 22.04 and 24.04 LTS without changing the version string.
 - 21 Oct 2026** ○ **php-src issue #24481 filed by a community member**
A use-after-free during unserialize() re-entrancy, later reported to reproduce on the patched builds. Untriaged, no CVE, no maintainer response.
 - 23 Oct 2026** ● **PHP's informational advisory on CVE-2026-115560, and CISA's update**
The PHP project suggests `session.upload_progress.enabled=0` until the next regular release. CISA adds a Sigma detection rule to its alert.
 - 24 Oct 2026, 05:46 UTC** ● **php2shell-zeros repository goes public with exploit scripts**
Its Disclosure section reads "Unfiled at publication time. The choice is the author's."
 - 24 Oct 2026** ○ **RedEye publishes Part Two**

24 Oct 2026,
21:46 UTC



Six claimed unpatched vulnerabilities in fully patched PHP, three self-scored at CVSS 9.8, said to be escalated to CISA.

README changed to "Findings emailed to sec@php.net after public release"

About 16 hours after the exploit scripts went public. "Awaiting response."

25 Oct 2026



NativePHP releases 4.6.1 and 1.0.3; no PHP project response to the claims

The NativePHP flaws are real and fixed in NativePHP's own code. Nothing from the PHP project on the other claims as of today.

Further Reading

- [Security advisory: Critical remote code execution vulnerabilities, update now](#) - the PHP project's 21 October announcement of 8.5.12, 8.4.27, 8.3.36 and 8.2.35.
- [php.net news](#) - where the 23 October informational advisory on CVE-2026-115560 was posted, and where the next release will be announced.
- [Critical Vulnerabilities in PHP Require Immediate Patching](#) - CISA's alert, covering the two fixed CVEs and updated with a Sigma detection rule.
- [php-src issue #24481](#) - the community report of a use-after-free during unserialize() that reproduces on the patched builds.
- [php-src security advisories](#) - the PHP project's advisory page, where the full advisory for CVE-2026-115560 will appear.

Not sure which PHP version each of your sites is on, or whether your last backup restored? A [free audit](#) of one site shows the PHP version and the checks that matter here.

Frequently Asked Questions

Do I need to do anything today?

If your host has applied the PHP release that fixed PHP2Shell, nothing new. If you have not yet set `session.upload_progress.enabled=0`, do that, because the PHP project suggests it until the next regular release. Beyond that, the work is preparation: know each site's PHP version, know who applies updates at each host, and restore a backup to prove it works.

Is there another PHP update coming?

Yes. The PHP project has said it will publish a full advisory for CVE-2026-115560 with the next regular bugfix release. PHP puts those out roughly every four weeks, and no date has been announced.

Are the "survives the patch" zero-days real?

Unconfirmed. None of the six claims in the php2shell-zeros repository has a CVE, and none has been confirmed by the PHP project, Canonical or any independent party. The exploit code was published before the PHP project was emailed, so the people who could verify them have not had the chance yet. The NativePHP item is real but is a flaw in NativePHP, not PHP, and no Joomla or WordPress site runs it.

Does this affect Joomla or WordPress itself? Do I need a CMS update?

No. The fix is in PHP, which your host installs and updates. WordPress.org has said no WordPress update is needed, and there is no Joomla release for this either. Keep your CMS updated as normal, but nothing in Joomla or WordPress changes because of PHP2Shell.

My host says PHP is patched but the version number did not change. Is that right?

It can be. Distributions such as Ubuntu back-ported the fix into their existing packages without changing the version string, so `php -v` on a patched Ubuntu box still prints 8.1.2 or 8.3.6. Ask the host for the package version or the security notice it applied, rather than relying on the version number alone.

How will I know when the next PHP release is out?

Watch the php.net news page and the php-src security advisories page on GitHub, which is where the full advisory for CVE-2026-115560 will appear. mySites.guru also posts security alerts on this blog, by email newsletter and on social media, and flags the PHP version of every connected site.

AI MODIFIED

Written and edited by a human, with AI assistance. [Our approach to AI](#)

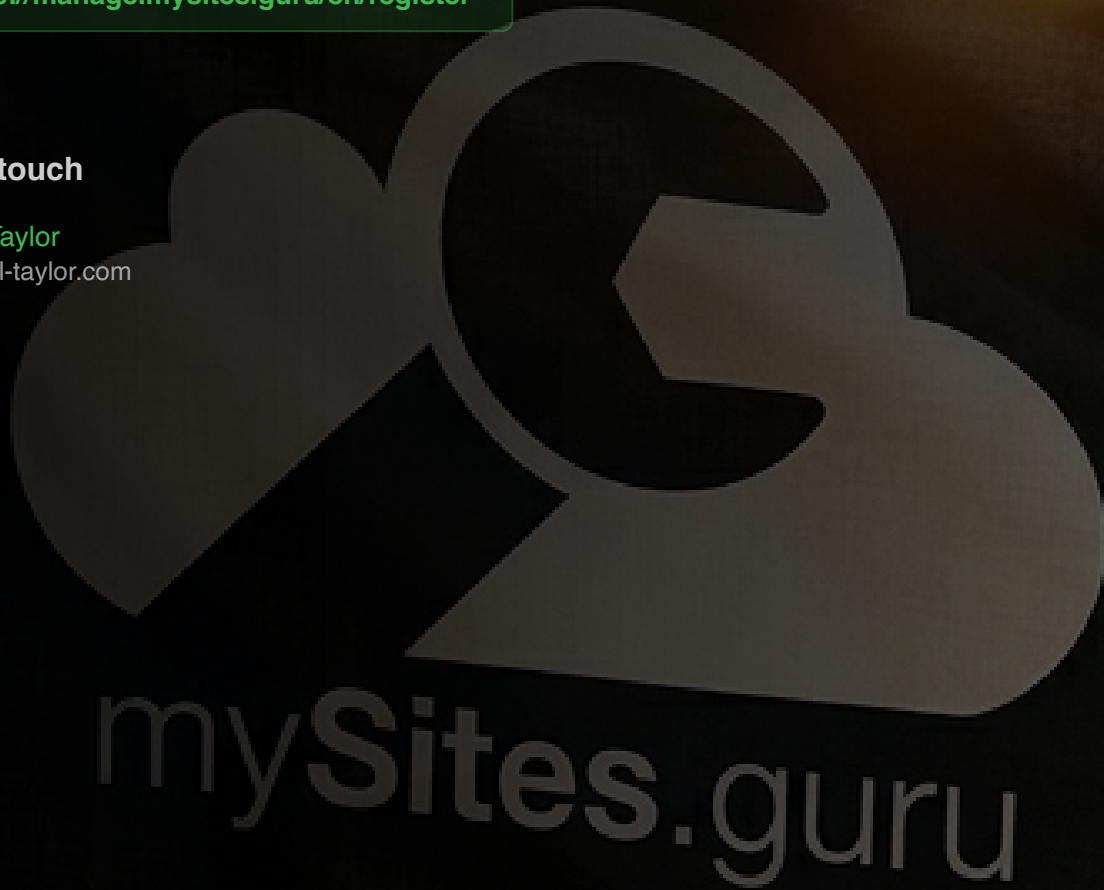
Get Your Free Site Audit

See how your WordPress and Joomla sites measure up.
No credit card required.

<https://manage.mysites.guru/en/register>

Get in touch

Phil E. Taylor
phil@phil-taylor.com



© 2026 Blue Flame Digital Solutions Limited. All rights reserved.

mysites.guru

