



Release Radar, a Free Joomla and WordPress Release Tracker

A free public log of new Joomla and WordPress extension releases, including the paid ones no plugin directory covers. RSS and JSON, no account needed.

Phil E. Taylor | 1 September 2026

FREE!!!! tldr? <https://releases.mySites.guru> go there instead of reading.

If a Joomla extension developer shipped a new version of their component this morning, how would you find out?

For most people the honest answer is that they would not, at least not on purpose. They would find out the next time they opened a site's update screen and saw a number sitting there. Or they would catch a forum post. Or an email would arrive from a mailing list they joined in 2019 and have filtered to a folder ever since.

WordPress looks better served until you check. The wordpress.org plugins API does publish version and last-updated data, machine-readable, for the 71,000-odd plugins hosted in the directory, and it is useful. The plugin RSS feeds that people still recommend redirect to HTML pages now and return no feed at all. The bigger limit is that the directory covers free plugins only. Elementor Pro, ACF Pro, WP Rocket, Gravity Forms and the rest of the commercially distributed plugins have no directory entry, so the software agencies most depend on is the software with no release feed.

Joomla has no equivalent for anything. Extensions there are distributed by their developers directly, each with its own update server, so there is no index to aggregate in the first place. The Joomla Extensions Directory tells you an extension exists; its "Recently Updated" page orders by edits to the listing rather than by releases, and the old `feeds.joomla.org` extension feed stopped resolving years ago.

We have been sitting on the answer for fourteen years and deleting it twice a day.

What Release Radar is

Release Radar is a free public log of new Joomla and WordPress extension, plugin and template releases. It is live now at `releases.mysites.guru`, it needs no account, and it has an RSS feed and a JSON endpoint.

Each entry records the extension name, the vendor, the version number, the date the release was first observed, and a link to the vendor's own page. You can filter it to Joomla only or WordPress only, and you can look at a single vendor's history.

It also covers paid extensions, which nothing else does. Other public release sources read a central directory, so they see only what that directory holds, and commercial software sits outside it. Release Radar sees what real sites see.



New Joomla and WordPress Extension/Plugin releases, as they appear



Release Radar watches the update feeds that thousands of live sites poll every day and records each new extension version once. Free, public, and no account needed.

A plugin only appears here once at least one mySites.guru customer runs it on a real site. That keeps the list to what people have installed, rather than an index of everything ever published.

RELEASES TRACKED
29

JOOMLA
7

WORDPRESS
22

[Hide intro](#)



Release Radar says what shipped. mySites.guru finds which of your sites are behind and updates them.

[Find out more](#) →

Filter releases

All

Joomla

WordPress



Today	EB Ajax Search 2.4	ExtnBakers	Link
	Admin Notes 1.5.3	Jeroen Moolenschot Joomill	Link
	CookieHint 3.2.0	reDim GmbH	Link
	BaGallery 2.5.1.2	Balbooa	Link
	Ultimate Addons for Elementor Pro 1.45.4	Brainstorm Force	Link
	Hostinger AI 3.0.54	Hostinger	Link
	Joomill Installer 1.0.1	Jeroen Moolenschot Joomill	Link
	Themify - WooCommerce Product Filter 1.5.6	Themify	Link
	Ultimate Addons for WPBakery Page Builder 3.21.6	Brainstorm Force	Link
	CatalogX — Catalog Mode, Enquiry & Quotes for WooCommerce 6.1.4	MultiVendorX	Link
	Quttera Web Malware Scanner 4.1.0.32	Quttera team	Link
	Unicode 2.12.8	Undsgn	Link
	WPvivid Backup Plugin 0.9.134	WPvivid Team	Link

The public log. Name, vendor, version, the date it first appeared, and a link to the vendor.

There is a second half, inside mySites.guru, that answers a different question.

The public log tells you what shipped. The Release Radar page in your mySites.guru account tells you what shipped *for the software you actually run*, across your whole portfolio, and how many of your sites run each of them.

That page is part of the subscription.

2x

Snapshots per day
of every connected site

100,000+

Site connections daily
across 90,000+ connected sites

3

Checks before publishing
corroboration, novelty, lineage

0

Extra vendor requests
nothing new is fetched from vendors

Release Radar is built from update checks the platform already performs.

Where does the release data come from?

Every site connected to mySites.guru asks its own vendors, twice a day, what updates are available for the software installed on it. That check uses the site's own licence key, from the site's own IP address, against the vendor's own update feed. It is what the site would do by itself if somebody clicked the update button in the admin.

We snapshot connected sites twice a day, which works out at over 100,000 site connections daily across the 90,000+ sites connected to mySites.guru. Until now, when a site came back and said "there are six updates waiting, here is what they are", the platform reduced that answer to the number six and discarded the rest. Six is what the dashboard needed. The detail went in the bin, twice a day, for years.

Release Radar keeps the detail. Each distinct release is recorded once, with the earliest date any connected site saw it.

One other source, for speed rather than breadth

There is a second input. A connected site only reports a release once it polls its own update feed and turns out to be behind, and a full pass over every connected site takes about ten hours. wordpress.org's own plugin registry knows within minutes.

So Release Radar polls that registry once an hour and folds in what it finds, with one restriction: a plugin no connected site runs is dropped on the spot. wordpress.org lists more than 71,000 plugins and this is not an attempt to mirror them. What the registry buys is latency, not breadth. For a wordpress.org-hosted plugin that connected sites already run, it closes a gap measured at between five and twelve hours down to under one.

Joomla has no equivalent to poll, so Joomla releases still arrive at the speed of mySites.guru's own update checks.

This is not a scraper

Release Radar makes no requests to any vendor's server. Every observation from a connected site is a by-product of a check that was already happening for that site's own benefit, and the one first-party source it does poll, wordpress.org's public plugin API, is a single page an hour. Crawling vendor download pages would have produced comparable data, a real support burden for the developers on the other end, and a scraper that broke every time somebody changed a page template.

How do we know a release is real, and actually new?

A mySites.guru connector reports what its site tells it. If a site is compromised, or somebody has been editing files by hand, or an extension is misconfigured, that site can report an extension name and a version number that are complete fiction.

Separately, a site that has not been scanned for a while reports everything it is behind on at once, most of which shipped months ago.

Three checks sit between an observation and the public log, and each one catches something the other two miss.

Corroboration

One site making a claim is a claim. Two unrelated customers, on different infrastructure, independently reporting the same extension at the same version, is a fact. So a release reaches the public log once two distinct customer accounts have seen it, or once wordpress.org's own registry confirms that exact plugin and version exists. A first-party registry cannot be somebody's hacked connector, which is the only thing the customer count was ever defending against.

This costs a little latency on obscure software, and for a commercial extension that only one customer in the world runs it means the release never publishes at all. We took that trade on purpose. A public log whose worst failure mode is "occasionally slow" is worth considerably more than one whose worst failure mode is "a single hacked site put a fake extension on a public website".

Novelty

Corroboration proves a version exists. It says nothing about whether it is new, and that turned out to be the bigger problem.

Because the platform only records a release when a site is behind on it, a site that connects after a long gap reports its whole backlog at once, and every version in it the corpus has not seen looks like news. On one morning during testing a single site created thirteen releases inside one second, one of them a build that hundreds of other connected sites were already running.

The evidence needed to catch that was already sitting there. mySites.guru records every build every connected site runs, so "was anybody already running this before we

first saw it offered?" is one indexed lookup. Applied to the releases published on the last day before this check existed, it caught 78% of them, in 18 milliseconds.

Lineage

The last check is the one nothing else asks. Joomla update feeds are branched by target platform, so a site on an old Joomla release is offered a maximum version that a site on a current one never sees. Two such sites are two customers, which clears corroboration, and the version is still absent from every site we can see, which clears novelty. A thoroughly superseded release passes both and arrives at the top of a feed of new releases.

So a release is also checked against the newest version already held on its own line, matched on the major version and on the branch tag that vendors write into the version string. It has to be both. "Never publish below the highest version we have ever seen" would suppress the security releases vendors ship on old branches, and a customer stuck on an old branch needs those more than anyone.

A release that fails any of the three is held for review rather than rejected, with the reason attached. Branch tags are vendor prose and this will read one wrong eventually, so the failure has to be recoverable.

The same reasoning runs through the rest of the feature. Release Radar's only real claim is that its facts are facts, so each design argument gets settled in favour of claiming less.

What Release Radar will not tell you

A log like this is easy to over-read, so the limits are worth stating.

Not a complete index

Release Radar only sees software installed on at least one connected site. An extension no one in the connected estate runs is invisible to it. What that filter leaves behind is

software in live production use across a large number of real agency-managed sites, which is a reasonable working definition of the releases that matter. Treat it as a wide view of the ecosystem rather than a census of it.

No changelog text

Release Radar publishes the name, the vendor, the version, the date and a link. Changelog text is the vendor's own copyright and belongs on the vendor's own page. Republishing it would also put two pages in competition for the same search results, which helps neither the vendor nor the reader.

No back catalogue

The log starts on 1 September 2026 and fills forwards. Anything first seen before then is kept internally and never shown.

The reason is slightly counterintuitive. The platform records a release only when a site is *behind* on it, so the first harvest after deployment ingests the accumulated backlog of the whole connected estate at once rather than discovering anything new, and each of those rows would be stamped with the date it was imported. A release from 2021 would appear on the front page dated this morning, which would have made every date on the page meaningless.

That cut-off was originally calculated rather than fixed: a rolling window that opened a week after the earliest record. It had one fatal property, which is that rebuilding the corpus reset it. The corpus was rebuilt several times over two days while the way releases are identified was being corrected, and each rebuild pushed the window another week into the future. A guarantee that moves whenever the data is repaired is not a guarantee, so it became a fixed date instead.

The accepted cost is that a release which shipped before that date is invisible for good, which errs towards claiming less.

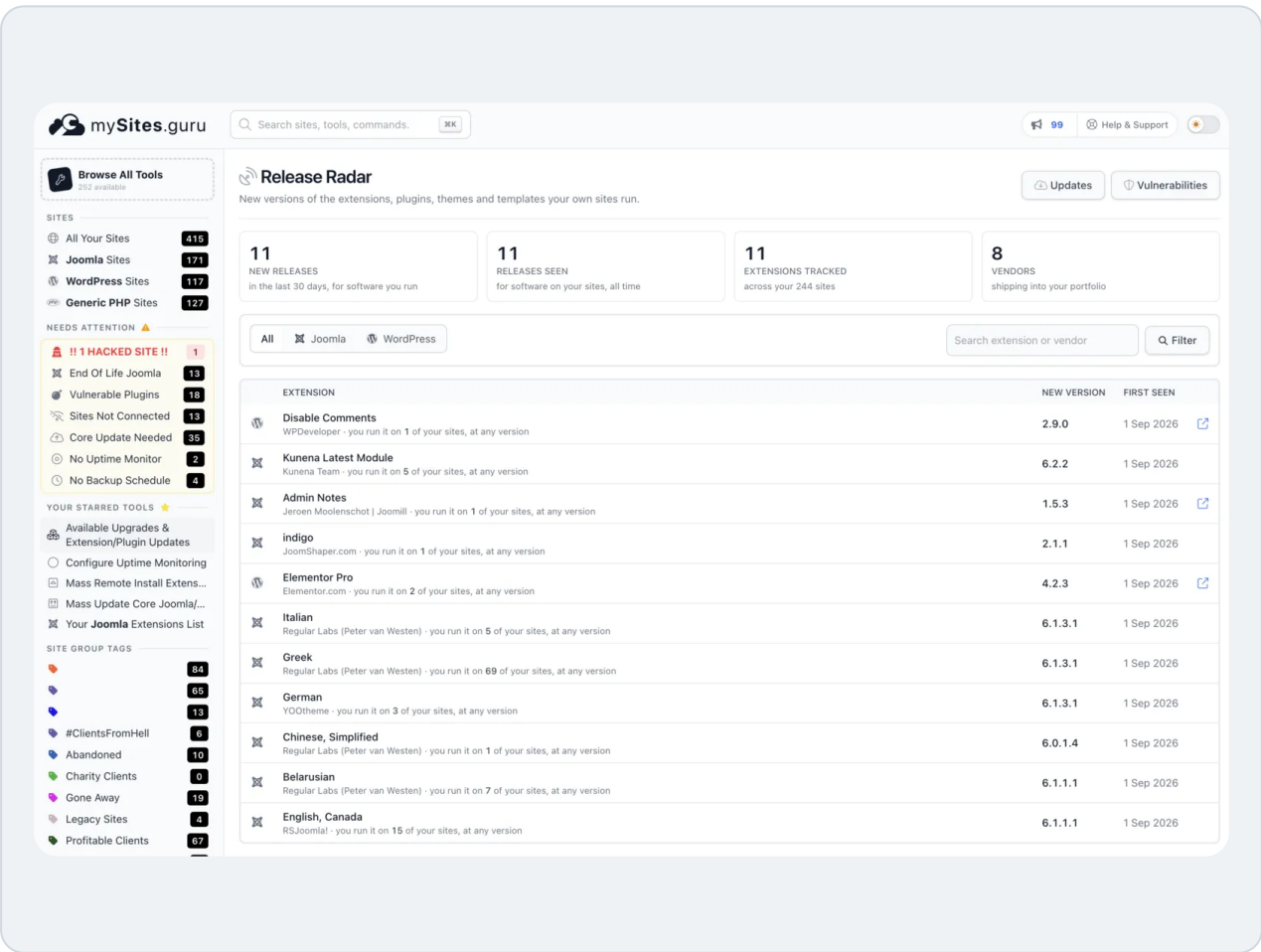
No judgement on vendors

Release Radar reports what shipped. It draws no conclusions about how, or whether, a vendor announced it.

BONUS: mySites.guru subscribers get their own Release Radar, specific to the products running on their sites

The public log answers “what shipped this week”. Your account answers “what shipped for the software I actually run”.

The Release Radar page inside mySites.guru shows new versions of the extensions, plugins, themes and templates your own sites run, matched on the extension itself rather than on whether you happen to be behind on it. That distinction is the entire point of the page.



The subscriber view: the same log, narrowed to software your own sites run, with a count of how many of them run it.

Consider the agency that keeps everything patched. A list of what you are behind on shows them an empty page, because there is nothing to be behind on, and an empty page tells them nothing at all. Matching on the extension is what tells a diligent agency that Akeeba shipped something yesterday and their sites have already taken it.

The page stops there on purpose rather than becoming a second work queue. "Which of my sites need updating" is what the Updates page already answers, and it sits one click away in the header. Two features answering one question is worse than one.

It applies no moderation filter either. A release that has yet to clear the checks for public publication is still real software sitting on your server, and hiding it from you would be the wrong call.

From there the rest of mySites.guru is already wired up: [bulk updates across your sites at once](#), [per-extension auto-updates](#) if you trust a given vendor to ship safely, and the [update queue](#) if you would rather not fire two hundred updates at one shared server simultaneously.

How do I follow Release Radar?

RSS

`https://releases.mysites.guru/feed.xml` has the 50 most recent releases. Any reader will take it, and so will most chat tools if you want new releases dropping into a team channel.

JSON

`https://releases.mysites.guru/api/releases.json` returns the same data for scripts:

```
# Everything, most recent first
curl -s https://releases.mysites.guru/api/releases.json

# Joomla only, 200 rows, one vendor
```

```
curl -s "https://releases.mysites.guru/api/releases.json?platform=joomla&T
curl -s "https://releases.mysites.guru/api/releases.json?vendor=Akeeba%20L
```

No key, no rate-limit dance, and permissive CORS headers so a browser page can fetch it directly. `limit` caps at 200. An unrecognised `platform` value returns an error instead of dropping the filter with no warning, so a typo in your script announces itself.

MCP

Subscribers with the [mySites.guru MCP server](#) connected get a `list_releases` tool, so you can ask an AI assistant what shipped this week and have it cross-reference against your own sites in the same breath.

The paid extensions nothing else tracks

Public release trackers work by reading a central directory, which caps what they can see at whatever the directory holds. On wordpress.org that means free plugins. On the Joomla side it means nothing at all, because there is no central update index to read. Commercial extensions sit outside that model: they ship from the vendor's own update server, behind a licence key, and a crawler holding no licence sees nothing.

Release Radar approaches it from the other end, reading what real sites got back when they asked their own vendors, using their own paid licences. A commercial Joomla component sold by subscription, a premium WordPress plugin sold from the developer's own store, a template framework licensed per-domain: if it is installed on connected sites and it shipped a new version, it appears. The wordpress.org registry is read too, but only ever to hear about a free plugin sooner than the connected sites would have told us, and never to decide what belongs in the log.

That is also why the log skews the way it does. Widely-installed commercial software shows up quickly, because plenty of connected sites run it and their update checks agree. Obscure free extensions appear slowly or not at all. For an agency deciding what to test this week, that is the right bias.

Vendor communication is a separate problem and Release Radar does not solve it. An extension can still ship a security fix with a three-word changelog, leaving the people running it with no realistic way of finding out what changed. What the log fixes is narrower: being able to see that *something* changed at all.

A new version is not the same as a security fix

Release Radar tells you a version shipped. It does not tell you why. For known-vulnerable versions, the Joomla Vulnerable Extensions List and the CVE index are the right tools, and mySites.guru matches those rules against what is actually installed on your sites rather than leaving you to cross-reference by hand.

What is not there yet

Joomla has no registry to poll, so a Joomla release only appears once connected sites have polled their own update feeds and reported it. That is hours rather than minutes, and there is no way to shorten it that does not involve crawling vendors, which is the one thing this was built to avoid.

WordPress themes are inventoried now, and Joomla templates always were, but a theme release cannot be confirmed against wordpress.org the way a plugin release can. Themes wait for the connected sites, the same as Joomla does.

The vendor name used to be missing on a large share of entries, because the name a vendor puts in an update feed and the name that ends up in a site's installed-extensions table are frequently different strings: one Joomla extension arrives from its feed as "Admin Tools Professional package" and sits in the database as "Admintools". Matching on the extension's own identifier instead of its name fixed that, and every release published so far shows a vendor. Sites running an older connector build do not send that identifier yet, so the occasional entry will still appear without one until those sites update.

None of these affects the accuracy of what is published. A release that appears is a release that was corroborated, tested against what the connected sites already run, and checked against its own version line.

Try it

releases.mysites.guru is open now. No account, no email address, no cookie wall.

If you manage more than a handful of Joomla or WordPress sites and you want the version of this that knows which of *your* sites are affected, that is [what mySites.guru does](https://what.mysites.guru), and there is a [free month](https://what.mysites.guru) to try it on your own portfolio.

Further Reading

- [Joomla Extensions Directory](#) - the official Joomla extension catalogue, useful for finding an extension but not for tracking its releases.
- [WordPress Plugin Directory](#) - publishes version and last-updated data for plugins hosted on wordpress.org, though not for commercially distributed ones.
- [Joomla Vulnerable Extensions List](#) - the community list of known-vulnerable Joomla extensions.

Frequently Asked Questions

What is Release Radar?

Release Radar is a free public log of new Joomla and WordPress extension, plugin and template releases, published by mySites.guru at releases.mysites.guru. It records the extension name, the vendor, the version number, the date the release was first observed, and a link to the vendor's own page. No account is needed to read it.

Where does Release Radar get its release data from?

Mostly from update checks that connected sites already perform. Every site connected to mySites.guru asks its own vendors what updates are available, using that site's own licence from that site's own IP address, and mySites.guru snapshots connected sites twice a day, which is over 100,000 site connections daily. Release Radar keeps the answers those checks return instead of discarding them, so it adds no extra traffic to any vendor's server. There is one other input: wordpress.org's own public plugin registry, polled once an hour, used only to learn sooner about plugins the connected sites already run.

Does Release Radar list every Joomla and WordPress extension in existence?

No. Release Radar only sees releases for software that is installed on at least one site connected to mySites.guru. An extension no one in the connected estate runs stays invisible to it. In practice that skews the list towards widely-installed commercial extensions, which are the releases most agencies actually need to know about, but it is a wide view of the ecosystem rather than a complete index.

Does Release Radar cover paid and commercial extensions?

Yes, and that is the main thing it offers that no other source does. Public release trackers read a central directory, so they can only see free directory-hosted software: the wordpress.org plugins API covers free WordPress plugins only, and Joomla has no central update index at all. Commercial extensions ship from the vendor's own server behind a licence key. Release Radar sees them because it reads what real sites got back when they checked for updates using their own paid licences.

Why does a release take a while to appear on Release Radar?

Three checks stand between an observation and the public log. A release has to be corroborated, either by two separate mySites.guru customers independently reporting the same version or by wordpress.org's own registry confirming that plugin and version exists, because one site with a misconfigured or compromised connector can claim any extension

name and version it likes. It has to be new, which is tested by asking whether any connected site was already running that exact build before mySites.guru first saw it offered. And it has to be the newest release on its own version line, so a vendor still serving an older version to an older CMS branch cannot push a stale release to the top of the feed. Anything that fails a check is held for review rather than discarded.

Does Release Radar republish vendor changelogs?

No. Release Radar stores and shows facts only: name, vendor, version, first-seen date, and an outbound link. A changelog is the vendor's own copyright and belongs on the vendor's page, so mySites.guru links out to it rather than reproducing it. That also avoids creating duplicate content that would compete with the vendor's own release notes in search.

Is Release Radar free, and do I need a mySites.guru account?

The public log at releases.mysites.guru is free and needs no account, including its RSS feed and its JSON endpoint. The part that narrows the log down to the software your own sites actually run, and joins it up with which of your sites are behind and which updates close a known vulnerability, is the Release Radar page inside mySites.guru and is part of the subscription.

How do I get Release Radar in my feed reader or my own scripts?

Subscribe to <https://releases.mysites.guru/feed.xml> in any RSS reader for the 50 most recent releases. For scripts, <https://releases.mysites.guru/api/releases.json> returns JSON and accepts `platform=joomla`, `platform=wordpress`, a vendor filter and a limit of up to 200. The JSON endpoint sends permissive CORS headers, so a browser page can read it directly.

AI MODIFIED

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

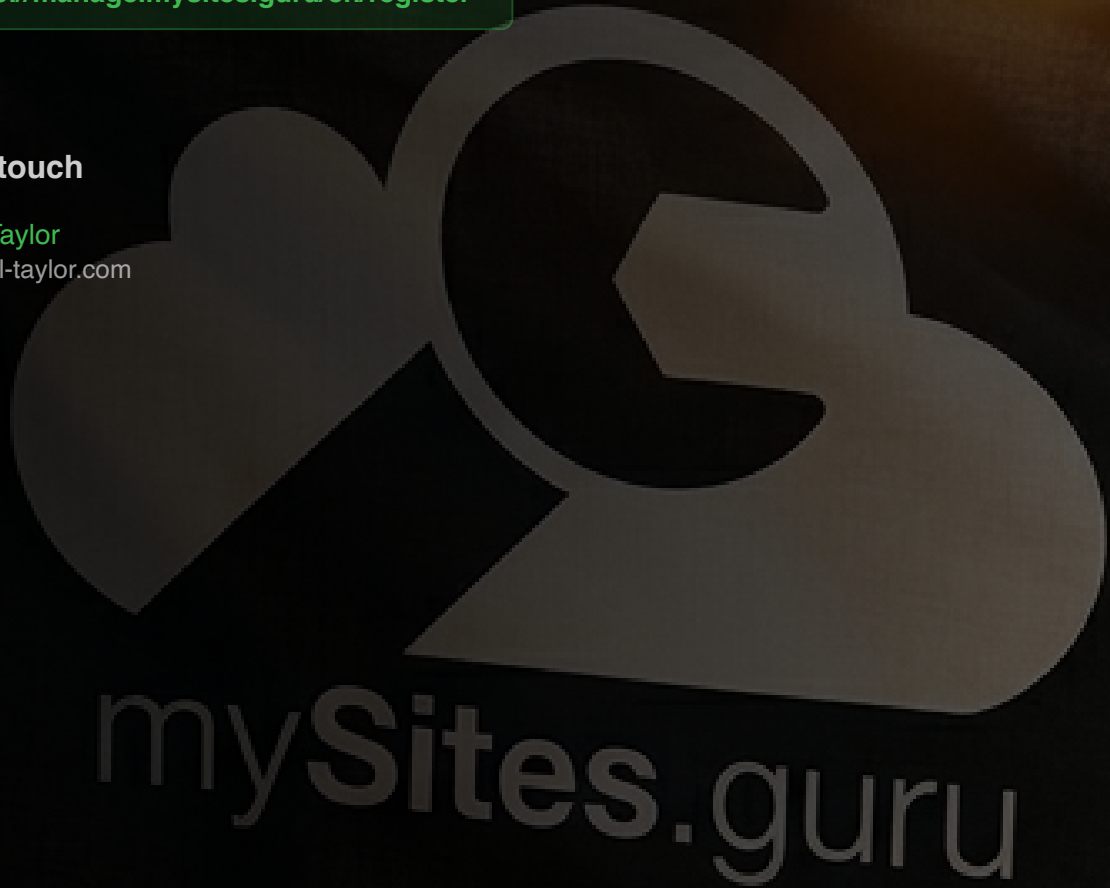
Get Your Free Site Audit

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

