mySites.guru

T4 Page Builder 2.3.0 Fixes an Unauthenticated Mail Relay

JoomlArt's T4 Page Builder 2.3.0 closes an unauthenticated open mail relay we reported in August. A week on, five in six installs we see are still older.

Phil E. Taylor | 3 September 2026

JoomlArt published T4 Page Builder 2.3.0 on 28 August 2026. It closes an unauthenticated open mail relay in the Joomla extension's front-end contact endpoint, along with a missing CSRF token, absent rate limiting, and a live Mailchimp API key hardcoded into the shipped source. If you run T4 Page Builder on a public Joomla site, update it.

We reported the flaw to JoomlArt on 17 August, having found it that afternoon on a client's server while investigating abuse that was already under way. The fix is real, and we have read it rather than taking the changelog's word for it: we audited the released 2.3.0 package on 3 September and all four issues are closed.

One piece of housekeeping before anything else, because the two are easy to confuse. This is T4 Page Builder by JoomlArt, `com_t4pagebuilder`. It is not SP Page Builder by JoomShaper, which had a mail relay of its own this summer under CVE-2026-65879. Different vendor, different codebase, different bug. If you run both, both need attention.

> **What to do now**
>
> Update T4 Page Builder to 2.3.0 or later, then clear the Joomla cache and test every contact and subscription form on the site. JoomlArt's own instructions are a manual download from the member area followed by an upload through System, Install, Extensions. Every version below 2.3.0 is affected, including the whole 1.x branch. If you cannot update today, unpublish any page with a T4 contact or subscription form on it.

What T4 Page Builder 2.3.0 Fixes

The component exposes a front-end JSON endpoint that dispatches named actions. One of those actions, `contact`, was exempted by design from the permission checks that guard the rest of the editor, so it was reachable by any anonymous visitor. That exemption is still there in 2.3.0, and it is fine now for reasons we will come to, but in 2.2.0 and earlier it opened onto an action that would send mail wherever it was told to.

Four separate problems sat behind it:

Problem	Effect before 2.3.0
Attacker-controlled recipient	The action read a `recipient` field straight from the request and delivered to it, using the site's configured `mailfrom` and `fromname`. The attacker chose the recipient, the subject and the HTML body.
No CSRF token	No Joomla form token was required, so the request needed nothing but a URL and a body.
No rate limiting	Nothing throttled the endpoint, so the same request could be sent in volume.
Hardcoded Mailchimp API key	A live API key sat in the component's source, readable in every copy of the extension.

The subscribe task on the same endpoint had the same gap, which let unauthenticated requests write rows into a site's AcyMailing subscription tables. We have written before about unauthenticated AJAX and JSON endpoints being the CMS security blind spot they are, and this is a textbook instance: the ACL check existed, and one action was excused from it.

We are not publishing the request shape, because most installs have not updated yet. The Joomla Extensions Directory takes the same line in its own guidance to vendors, which says explicitly that it does not ask developers to give away vulnerability details and advises against it.

How Did a Joomla Contact Form Become an Open Mail Relay?

The damage is not that someone can email you through your own contact form. It is that they can email anyone else, and it arrives looking like you.

Mail sent through that endpoint left the site using the sender identity configured in Joomla's Global Configuration. On a properly set up site that means it went out with the

domain's SPF pass and its DKIM signature. To a receiving mail server it was authentic mail from the domain, because in every technical sense it was. An attacker got a free, authenticated, reputable sending platform, and the site owner got the consequences.

Those consequences outlast the bug. Once a domain has been used to push phishing at scale it goes onto blocklists, and on shared hosting the sending IP takes every other site on the box down with it. We have written about verifying that your Joomla site's email actually works because deliverability is one of the most expensive things to break and one of the slowest to repair. An hour spent installing 2.3.0 is cheaper than a fortnight spent getting a domain delisted.

The hardcoded API key is a smaller problem with a longer tail. A credential shipped inside a distributed extension is readable by everyone who has ever downloaded it, so it stops being secret the moment it ships. We have seen the same pattern in a live Google Fonts API key hardcoded into Helix3. Removing it from the source is necessary, but it does not un-leak it, and the key should be rotated by whoever owns it.

Is the T4 Page Builder Fix Any Good?

Yes, and it is worth being specific about why, because the vendor's own summary puts the emphasis in the wrong place.

The release notes lead on CSRF protection. A Joomla form token is now required on the contact action:

```
// Public endpoint – require a valid Joomla CSRF token (POST body or X-CSR
if (!Session::checkToken('post')) {
    return ['error' => Text::_('JINVALID_TOKEN')];
}
```

On its own, that would not have closed the relay. Joomla hands a valid form token to any anonymous visitor who loads a page, so a script that fetches the page first and posts second has a token like everybody else. CSRF tokens stop a third-party site forging a request in a logged-in visitor's browser, which is a real problem and worth

fixing, but it was never the mechanism here, because the attacker was talking to the endpoint directly rather than borrowing anyone's session.

What actually kills the relay is one line further down, where the attacker's recipient stops being used at all:

```
$result = self::sendEmail(self::getAdminMails(), $subject, $emailContent,
```

`getAdminMails()` returns two things and nothing else: the `mailfrom` address from Global Configuration, and the first Super User's email. The `recipient` field is still parsed out of the request a few lines above, and the action still will not run without one, but it is never passed to the mail call. It is an untidy way to reach a correct answer, and the answer is correct: there is no longer any input that steers delivery.

Behind that sits a new `RateLimitSecurity` helper doing IP-based fixed-window limiting through Joomla's cache, defaulting to five requests per ten minutes and configurable in the component parameters. Captcha validation is now explicit rather than inferred from whether a plugin happens to be enabled, and because T4 stores rendered pages statically, a `FormSecurity` helper injects the captcha widget and the token into contact forms at render time. The Mailchimp key now comes from component parameters, and the old key is absent from the 2.3.0 package.

That is defence in depth, built by someone who understood the report. Our only quibble is the one above: the announcement sells the weakest of the four fixes as the headline.

Which of Your Joomla Sites Still Run a Vulnerable Version?

This is where the story stops being about JoomlArt and starts being about everyone who has to act on it.

A week after 2.3.0 shipped, here is what the T4 Page Builder installs on Joomla sites connected to mySites.guru looked like on 3 September 2026:

- Five in six were still running a version below 2.3.0.
- Roughly one in eight had updated.
- Nearly half were sitting on a single older release, 2.1.1.
- Another one in eight were still on the 1.x branch.

T4 Page Builder is not on many of the sites we monitor, so read those as proportions of a small population rather than as a survey of the Joomla world. The shape is still the shape. Most people did not update, and the ones who did not are clustered on releases that are years of development behind.

There is a plausible reason for the clustering, and it is not laziness. The Joomla Extensions Directory listing for T4 Page Builder still shows version 2.1.0, and the fix is a manual download from JoomlArt's member area. Anyone who installed from the directory and has never had cause to log in to the member area has no signal at all that they are four releases behind, let alone that one of those releases was a security release.

That gap is why our vulnerability tracking exists. mySites.guru keeps a live inventory of every extension on every connected Joomla and WordPress site, with the installed version on each, and flags any site running below a vendor's fix. Our rule for T4 Page Builder below 2.3.0 was written from the vendor changelog and dated to the day of release, which means subscribers with an affected site were told on 28 August. JoomlArt got back to the person who reported the flaw on 3 September. That ordering is the problem rather than a boast: the detection had to be reverse-engineered from a changelog because no one was told directly.

If you look after more than a handful of Joomla sites, the multi-site Joomla tooling and the vulnerable extension list are the parts of the subscription that turn an announcement like this into a work list instead of an afternoon of logins. A full site audit covers the other half of the question, which is whether anything reached the site before you got there.

The Ten Days Between the Acknowledgement and the Release

Start with what JoomlArt got right, because it is more than the ending suggests.

When the report reached them, they were fast and they were serious. The Joomla Security Strike Team forwarded it at 11:01 on 18 August. JoomlArt replied at 11:11, ten minutes later, thanked us for the write-up, said they took it very seriously given that it was being actively exploited, and confirmed they were sending it straight to their dev team. A ten-minute acknowledgement is better than most vendors in this ecosystem manage, and the CERT guidance treats 24 to 48 hours as normal. Nine days later they shipped a complete, well-engineered fix. When they finally did reply on 3 September they apologised without being asked to, in plain terms: “We should have notified you directly as soon as the fix was released, rather than leaving you to follow up.”

Now the rest of it.

That report should never have needed the Strike Team. It went to JoomlArt’s published support address on the evening of 17 August and produced nothing. It only reached anyone because David Jardin forwarded it the following morning, noting in his own message that the reporter “tried to reach out to you before hand but had no luck”. JoomlArt’s reply acknowledged the same thing, apologising that “our previous contact attempts didn’t get through”. A reporting route that swallows an active-exploitation report without a word during an incident is a broken reporting route, and it is not hard to see why this one is fragile. JoomlArt publishes no `security.txt`, no security policy page and no dedicated security contact. Their `/security`, `/security-policy`, `/report-security` and `/responsible-disclosure` paths all return 404. Their `/contact` URL currently 301-redirects to `http://localhost/cw_2026/ja_mainsite/joomlart/contact-us`, which is a developer’s machine, not a website.

After 11:11 on 18 August there was silence. No initial assessment, no status update, no question about our test setup, no draft advisory, and no offer of a pre-release build to verify the fix against. The release went out on 28 August without a word to us. We

learned it existed because we track vendor changelogs for a living, and we confirmed it worked by downloading the public package and reading it.

Measured Against Joomla's Own Standard for Extension Developers

There is a reason to hold this to a written yardstick rather than to our own preferences. On 10 August 2026, one week before we reported this flaw, the Joomla project published twenty numbered rules for extension developers handling a security report, written by David Jardin in his capacity as Team Leader of the Joomla Security Strike Team, and merged into the official Joomla Manual the same day. It is documented best practice rather than an enforced requirement, and nothing checks compliance. But it is the Joomla project's own answer to the question, and it existed before any of this happened.

Rule	What it asks for	What happened
1. Establish a clear reporting channel	A dedicated security contact, a `SECURITY.md`, or a security page on the vendor site	None of the three exists. The direct report was never answered.
3. Acknowledge the reporter quickly	Receipt inside a business day, an initial assessment in three to five, regular updates during longer investigations	The acknowledgement was excellent. The assessment and the updates never came.
9. Coordinate a disclosure date	Agree a timeline, and accelerate substantially where a flaw is already exploited	No date was proposed or agreed. The release shipped unannounced to us.
11. Assign the CVE before disclosure	Request one from the Joomla CNA at `security@joomla.org`	None requested. The Joomla CNA was already a recipient on the thread.
12. Credit the security researcher	"Always offer appropriate credit to the reporter"	Not offered, not discussed, not given.

Rule	What it asks for	What happened
13. Publish a security advisory	A dedicated advisory, not just a changelog	A blog post and a changelog. No advisory.
14. Tell users what they need to do	State the affected versions, the fixed version and the recommended action	Fixed version and recommended action, yes. Affected versions, never stated at all.
17. Do not hide security fixes in vague language	Name the class of flaw that was fixed	An open mail relay is described as “Safer recipients”.
18. Preserve the disclosure timeline	The model timeline includes “Fix provided to reporter for verification”	No pre-release build was offered. We verified against the public package, six days late.

The pre-release build is the one item on that list where no standard actually obliges anybody, so it is worth saying so rather than dressing it up. It is a strong recommendation in FIRST’s PSIRT Services Framework, which says to provide the remedy to the finder so they can validate it, and it is step five of the CERT guide’s ideal flow.

Skipping it is also how a vendor ends up shipping the same fix twice, and we have watched that happen all year. PageBuilder CK answered an unauthenticated file-upload remote code execution with a login check in 3.6.0, which still left any Editor able to run code; the real fix came in 3.6.3, and 3.6.5 was the version people actually needed. Cotton Cloud’s first release closed the login route and left the data reachable, so a second one had to follow. JoomShaper shipped SP Page Builder 6.7.1 as the fix, then 6.8.0 sixteen days later for the flaw 6.7.1 had not closed. Each of those rounds is another emergency update for administrators who had already dropped everything to apply the last one, and there is a limit to how often you can ask people to do that before they stop treating your security releases as urgent at all.

A researcher who has just spent an afternoon proving a bug is the cheapest quality assurance a vendor will ever be offered, and the offer expires the moment the release

goes public. JoomlArt got this one right first time, which is worth saying. They had no way of knowing that before they shipped it.

Rule 14 is the one that costs site owners real money, because a reader on 2.1.1 has no way to learn from that announcement whether it concerns them.

Calling a Mail Relay “Safer Recipients” Has a Cost

The T4 Page Builder 2.3.0 announcement runs to about 5,000 characters. Across all of it, these words appear zero times: vulnerability, exploit, relay, attacker, advisory, severity, CVE, report, researcher, credit, disclosure, discovered.

The relay is introduced as “Safer recipients: Hardened anonymous contact submissions to prevent arbitrary recipient addresses from being supplied through requests”, and later as “The contact workflow no longer trusts recipient addresses supplied directly through anonymous requests”. The section explaining why the update matters frames the whole release around “spam and automated abuse”. Nothing tells a reader that a fortnight earlier someone was using their site to send mail.

The word “unauthenticated” does appear, three times. Every one of them describes the AcyMailing subscribe task, the least severe of the four issues. The vocabulary was available. It simply was not applied to the mail relay.

None of this is a case of protecting users by withholding detail, which is the reasonable defence and the one the Joomla Extensions Directory explicitly endorses when it tells vendors not to publish vulnerability details. The directory’s test asks something else entirely: whether “any reasonably intelligent person reading the notice would understand that there is a new version available, that it is a security release, and that users need to update”. The first and third of those are satisfied here. The second is what “safer recipients” removes, and the joint guide that CISA, the NSA, JPCERT/CC and the Dutch and UK cyber security centres published in July 2026 has a name for the result: a silent fix, defined there as a release that fixes a vulnerability without mentioning its existence, and singled out because such releases are far less likely to be deployed.

Which brings it back to five in six. We cannot prove those two facts are connected on a sample this size, and we are not going to pretend otherwise. But the mechanism is not mysterious. An administrator triaging a fortnight of extension updates is deciding, one line at a time, what to do today and what to do eventually.

“Safer recipients: Hardened anonymous contact submissions to prevent arbitrary recipient addresses from being supplied through requests.”	“Your contact form can be used to send phishing from your domain. Update now.”
Scheduled for whenever.	**Done before lunch.**

The line on the left is JoomlArt's, verbatim from the T4 Page Builder 2.3.0 changelog. The line on the right is not. It describes the same fix.

What This Means for Anyone Running JoomlArt Extensions

T4 Page Builder is a small corner of JoomlArt’s presence. Their T4 and T3 template frameworks sit under roughly thirty-five times as many of the Joomla sites we monitor as the page builder does, which is why the process questions here matter more than this one component. A vendor whose security contact route drops reports, who ships fixes without telling the reporter, and who describes a mail relay as a recipient improvement is going to do those things again, on frameworks with a far larger footprint.

The practical response is to stop relying on vendors to tell you when something matters, rather than to stop using JoomlArt extensions, which are good and widely deployed for good reasons. Watch the versions, not the announcements. Treat any extension release you cannot explain as one worth reading about, keep an inventory

you can query in one place rather than fifty, and update the security ones the day they appear.

For JoomlArt specifically, four things would close most of this permanently, and three of them are an afternoon's work: publish a `security.txt` and a security contact that reaches a human, fix the `/contact` redirect that currently points at a developer's laptop, state affected versions in release announcements, and request a CVE from `security@joomla.org` when a release fixes a vulnerability. The Joomla project runs the CNA and has published the instructions. The fix in 2.3.0 shows the engineering is there. It is the paperwork around it that let this one down.

Timeline

2026-08-17

Unauthenticated mail relay found on a live client site

Traced during an investigation into a Joomla site that was already being abused. Reported the same evening to JoomlArt's support address and to the Joomla Security Strike Team, with the vulnerable code path, the hardcoded credential and a suggested fix. JoomlArt does not reply to the direct report.

2026-08-18

The Joomla Security Strike Team relays the report, and JoomlArt answers in ten minutes

David Jardin forwards the report at 11:01, noting that the reporter had tried to reach JoomlArt beforehand without luck. JoomlArt's support replies at 11:11: they take it very seriously given the active exploitation, and are sending it straight to their dev team. It is a good answer, and it is the last one for a fortnight.

2026-08-28

JoomlArt publishes T4 Page Builder 2.3.0

The announcement describes security hardening for contact and subscription forms and tells users to upgrade. It names no vulnerability, no affected versions, no CVE and no reporter. We are not told it has shipped.

2026-08-28 ● **mySites.guru flags every connected site below 2.3.0**

Our vulnerability rule was written from the vendor changelog and dated to the release, so subscribers running an affected version were told on the day. The first updated install appeared in our data that afternoon.

2026-09-01 ○ **The Joomla Security Strike Team chases for an update**

"Two weeks is a rather lengthy response time for an actively exploited issue." We confirm the same evening that we have heard nothing since 18 August.

2026-09-03 ● **JoomlArt replies, six days after the fix shipped**

"We should have notified you directly as soon as the fix was released, rather than leaving you to follow up." The reply points at the release announcement as the answer. There is still no advisory, no CVE and no credit.

2026-09-03 ● **We verify the fix by auditing the released package**

Read the shipped 2.3.0 component rather than taking the changelog's word for it. All four reported issues are closed, and the hardcoded Mailchimp key is gone from the package.

Further Reading

- T4 Page Builder 2.3.0 Released: Security Hardening and Admin Improvements - JoomlArt's own announcement, and the only public description of the release.
- Building Extensions: Security - the Joomla Manual's twenty rules for handling a security report, written by the Team Leader of the Joomla Security Strike Team.
- What Does A Security Release Notice Look Like? - the Joomla Extensions Directory's own test for a security announcement, and its advice that vendors should not publish exploit detail.

- Establishing a Coordinated Vulnerability Disclosure Program - the July 2026 joint guide from CISA, the NSA, JPCERT/CC and the Dutch and UK national cyber security centres, which names the silent fix as a practice to avoid.
- CERT Guide to Coordinated Vulnerability Disclosure - the reference text on coordinated disclosure, including the Avoid Surprise principle.
- CWE-352: Cross-Site Request Forgery and CWE-798: Use of Hard-coded Credentials - the two weakness classes in this release.

Frequently Asked Questions

What does T4 Page Builder 2.3.0 fix?

Four things. The front-end contact action in com_t4pagebuilder accepted an attacker-supplied recipient address and sent mail to it using the site's own sender identity, which made any site with a published T4 contact form an open mail relay. The same endpoint required no Joomla form token, applied no rate limiting, and relied on whether a captcha plugin happened to be enabled rather than validating a captcha response. The component also shipped with a live Mailchimp API key hardcoded into its source. Version 2.3.0 requires a form token, adds IP-based rate limiting, validates captcha explicitly, locks the recipient to administrator addresses, and reads the Mailchimp key from component parameters instead. We verified all four by reading the released 2.3.0 package on 3 September 2026.

Which T4 Page Builder versions are affected?

Every version below 2.3.0, including the whole 1.x branch. JoomlArt's release announcement never states an affected range, so there is no vendor-published floor to work from and the safe assumption is that anything older than 2.3.0 is exposed. If you run a T4 contact or subscription form on a public site, update.

Is this the same as the SP Page Builder mail relay?

No, though the two are easy to confuse. This is T4 Page Builder by JoomlArt, element com_t4pagebuilder. SP Page Builder is a separate page builder by JoomShaper with its own codebase. Both had an unauthenticated mail relay fixed in the summer of 2026, and both were reported by mySites.guru, but they are different products from different vendors and the underlying bugs are different. The SP Page Builder issue received CVE-2026-65879. The T4 one has no CVE.

Is there a CVE for the T4 Page Builder mail relay?

Not at the time of writing. We checked the National Vulnerability Database, the GitHub Advisory Database and Patchstack, and none of them hold a record for T4 Page Builder or com_t4pagebuilder. JoomlArt did not request one, even though the Joomla Security Strike Team was already on the email thread and Joomla operates the CVE Numbering Authority that assigns identifiers for Joomla extensions. No CVE today is not the same as no CVE ever, because one can still be requested from security@joomla.org.

Was the T4 Page Builder mail relay being exploited?

Yes. We found it on 17 August 2026 while investigating a client server where the abuse was already under way, and demonstrated delivery of an attacker-supplied message from the site's own identity against a test mail server before reporting it. That is why the report went to JoomlArt and to the Joomla Security Strike Team the same evening rather than through a longer disclosure window.

How do I find every Joomla site I run that still has an old T4 Page Builder?

mySites.guru keeps a live inventory of every extension on every connected Joomla and WordPress site, with the installed version on each, and flags any site running a version below a vendor's fix. Our rule for T4 Page Builder below 2.3.0 was built from the vendor changelog on the day of release, so affected sites were flagged before JoomlArt replied to the person who reported the flaw. Without that, finding your exposure means logging in to each site in turn and reading its extensions list.

Why does an open mail relay on my Joomla site matter?

Because the spam and phishing go out under your domain, from your mail server, with your SPF and DKIM signatures on them. The immediate cost is that your legitimate mail stops arriving: the domain gets blacklisted, the sending IP is flagged, and on shared hosting that reputation damage reaches every other site on the same address. Cleaning up a blacklisting takes far longer than installing the update that would have prevented it.

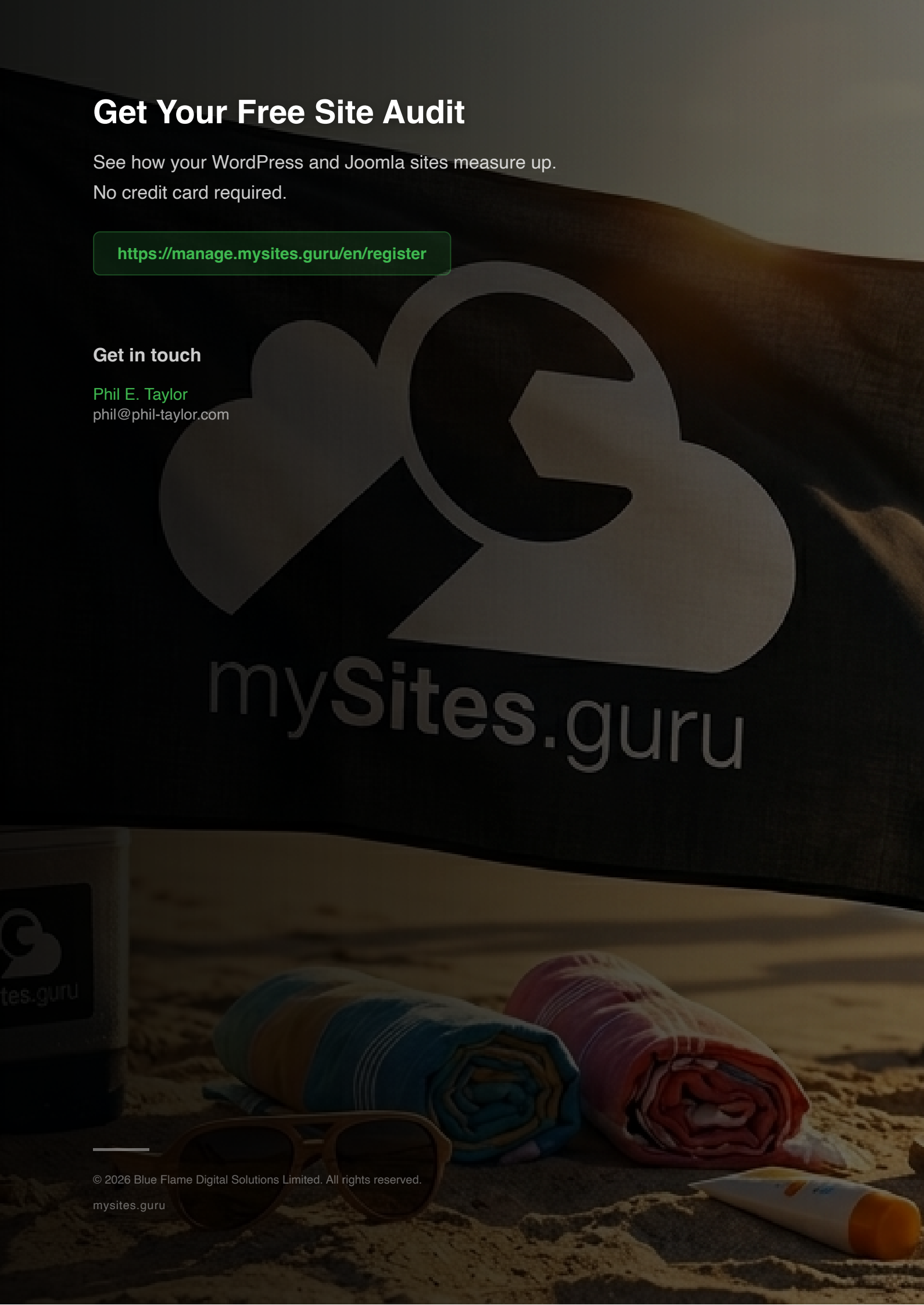
Written and edited by a human, with AI assistance. Our approach to AI

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