mySites.guru

Helix3 Shipped a Critical Fix as "Security Update"

Helix3 3.1.1 patches an unauthenticated file write and arbitrary file delete in the Helix3 ajax plugin. JoomShaper announced it, but told nobody what it fixes. Here is the detail, and why you should update now.

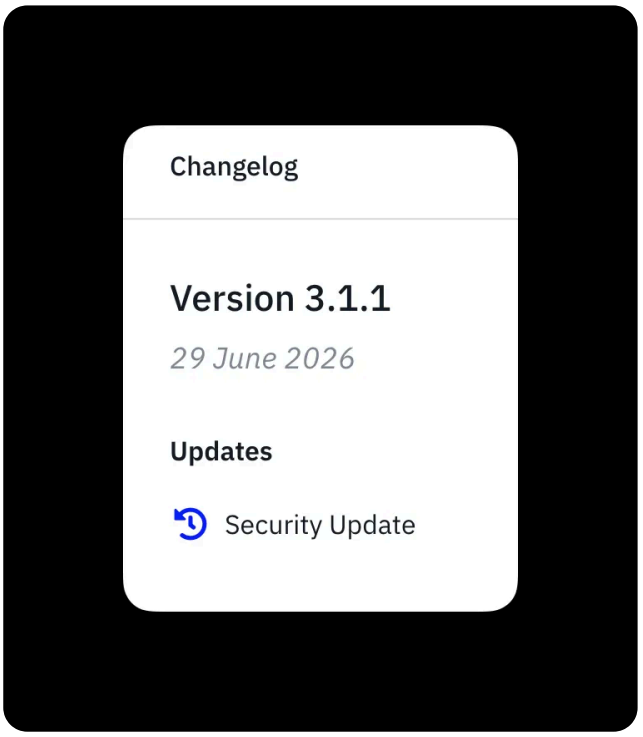
Phil E. Taylor | 29 June 2026

Active Joomla security alerts: Helix3 File Write • JCE Profiles Hack • PageBuilder CK RCE • iCagenda Vulnerability • SP Page Builder Zero Day

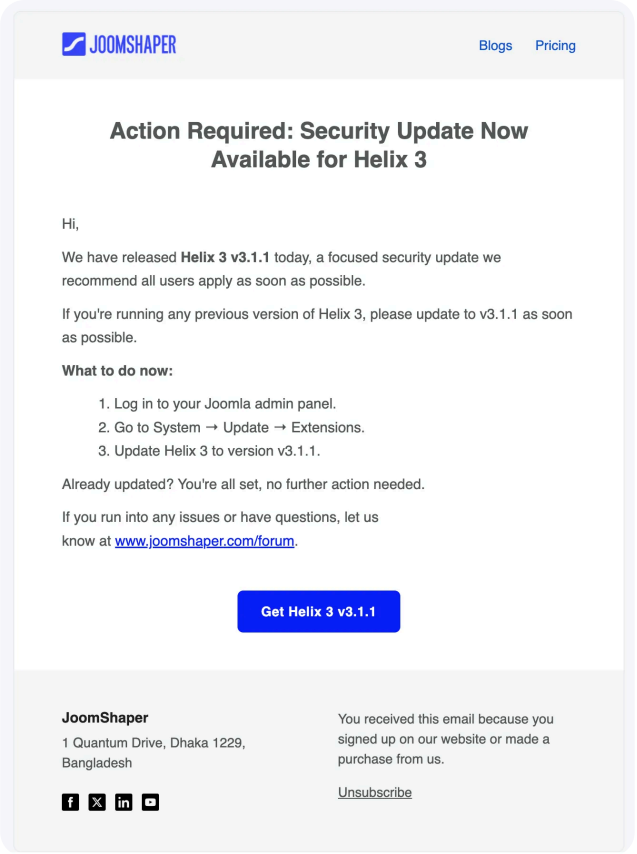
JoomShaper shipped Helix3 3.1.1 on 29 June 2026. The update fixes a serious security flaw: an attacker with no login at all could write files into your template, delete arbitrary files on your server, and overwrite your template settings, all through a single unauthenticated request. We found these issues while investigating a hacked mySites.guru customer site, and reported them privately under responsible disclosure.

JoomShaper did tell people an update was available. They emailed users and they wrote a changelog entry. Credit where it is due, plenty of vulnerabilities get patched in total silence. The trouble is what those two announcements actually said.

Here is the Helix3 3.1.1 changelog on JoomShaper's own site, in full:



Two words. "Security Update." The email went a little further:



“A focused security update we recommend all users apply as soon as possible.” That is better than the changelog, and the “Action Required” subject line is the right instinct. But put both side by side and the same gap remains: neither tells you what was fixed, how severe it is, or whether it is being exploited. If you manage Joomla sites for clients, you still cannot answer the only question that matters: do I patch tonight, or does it wait for the monthly maintenance window? That is the problem this post is really about.

TL;DR

TL;DR: Helix3 3.1.1 patches an unauthenticated file write, an unauthenticated arbitrary file delete, and an unauthenticated template-parameter overwrite in the Helix3 ajax plugin. The handler is reachable through Joomla’s `com_ajax` dispatcher and ran several actions before any token or permission check. It affects current Joomla (4, 5 and 6), not just legacy installs. JoomShaper announced the update by email and changelog, but neither says what was fixed or how serious it is. Update every Helix3

site to 3.1.1 now. If you manage more than a handful of Joomla sites, mySites.guru can list every Helix3 install in your account in seconds and push the update.

How to find every Helix3 site you manage with mySites.guru

The first question after any framework vulnerability is the hard one: which of my sites actually run this? A vendor email helps only if it reaches the right inbox and you connect it to the right sites. You still have to know where Helix3 is installed.

mySites.guru keeps a live inventory of every extension, template and framework installed on every Joomla and WordPress site in your account. You search for Helix3 once, and you get back every affected site, the version each one is running, and whether the 3.1.1 update is available. No logging into 40 admin panels one at a time. No guessing.

View all your Helix3 installations

Open Helix3 Extension Search

Lists every installed version across all your connected Joomla sites. Filter for anything below 3.1.1 to find the installations that still need updating.

From that same list you can push the update to every affected site, so a vague vendor notice becomes a five-minute job across your whole account instead of an evening of manual work. If you are not a customer yet, run a free security audit on one site and see the extension inventory for yourself, or read how we manage multiple Joomla sites from one screen.

That is the practical answer. The rest of this post is the detail JoomShaper's changelog left out, so you can judge the severity yourself.

What Helix3 is, and what it is not

JoomShaper ships two products with confusingly similar names, and getting them mixed up will send you patching the wrong thing.

Helix3 is the original template framework. It installs as a system plugin (`System – Helix3 Framework`) plus a companion ajax plugin, `plg_ajax_helix3` . This is what 3.1.1 patches. Despite the name and its age, it is not a Joomla 3 relic: the 3.1.1 update manifest declares compatibility with Joomla 4.0 through 6.5, and the patched code carries explicit Joomla 5 and 6 branches. It runs on current Joomla.

Helix Ultimate is a separate, newer template with its own codebase and version numbers. It is not affected by this issue. If you run sites on both, only the Helix3 installs need the update.

The vulnerable code is the `onAjaxHelix3` handler inside the Helix3 ajax plugin.

What the bug actually did

Joomla has a built-in dispatcher called `com_ajax` that lets plugins expose an endpoint at a predictable URL. It is a normal, useful feature. The catch is that `com_ajax` will happily call a plugin's handler for a completely anonymous visitor. Authorisation is the plugin's job, not Joomla's.

In Helix3 before 3.1.1, the handler took an `action` parameter and ran the matching code. Most of those actions did their work before checking a session token or a user permission. So an unauthenticated request could reach them. Three of them are dangerous:

- `save` writes attacker-controlled content to a file under the active template. The filename is forced to end in `.json` , but the path was not sanitised, so directory traversal let the file land outside the intended layout folder.
- `remove` deletes a file. It applied no extension constraint and no path validation at all, so path traversal turned it into an arbitrary file delete: anything the web server user could reach.
- `import` overwrites the stored parameters of any template style in the database.

There were more unguarded actions beyond those three (`load`, `resetLayout`, `updateFonts`, `fontVariants`), all reachable without authentication.

Is the file write remote code execution?

Not automatically, and this is worth being precise about because overclaiming helps nobody. The write is constrained to a `.json` extension. On a hardened host that is a serious integrity problem but not a guaranteed shell.

It becomes code execution in two realistic situations. First, on server configurations that pass a file like `name.php.json` to the PHP handler (some misconfigured Apache setups do exactly this). Second, the arbitrary-delete primitive can be chained with other upload paths to remove the protections, like a `.htaccess` or an `index.html`, that would otherwise stop a planted file from running. The arbitrary delete and the template-parameter overwrite are damaging in their own right on every host, regardless of whether you ever reach code execution.

The rest of the release: everything else 3.1.1 fixed

The unauthenticated file write is the headline, but it is not the whole release. Our review of the Helix3 ajax plugin turned up several more issues, and 3.1.1 closes all of them. None of the extra fixes are described anywhere public either. If you are weighing how urgent this update is, the full list is the answer: this is a security release with several distinct holes closed, not a single bug fix.

Here is everything 3.1.1 changes, beyond the `save`, `remove` and `import` actions already covered.

Image upload accepted any file type. The old upload action took the file extension straight from the uploaded filename, with no allow-list and no content check, then saved it. So a logged-in user with the common `core.create` permission, an Author or Editor on many Joomla sites, could upload a `.php` file and run it. That is a privilege escalation to code execution from a low-trust account. The new code restricts uploads

to `jpg`, `jpeg`, `png`, `gif` and `webp`, runs the name through `File::makeSafe`, and verifies the file is actually an image with `getimagesize()` before it is written.

The image delete action had the same traversal as `remove`. It took a `src` parameter and deleted it with no path validation, so it was a second route to arbitrary file deletion. The new code forces the path under the site's `images` folder and resolves it with `realpath()` before deleting anything.

Output was not escaped, which is cross-site scripting. The font picker output, the uploaded-image markup, and the `data-` attributes the framework builds were all emitted without escaping. An attacker who could influence any of those values could inject script into the admin's browser. The new code runs them through `htmlspecialchars`, and sanitises attribute keys.

A live Google Fonts API key was hardcoded in the source. The old `updateFonts` action shipped with a real API key embedded in the plugin file, readable by anyone with the code. The new version removes it and reads the key from a template parameter instead. A leaked credential is a leaked credential, regardless of how the bug was found.

The article-rating action could be spammed. Voting had no rate limit and no check that the article being rated existed or was published, so it could be hammered against arbitrary content IDs. The new code verifies the article, tracks votes per session, and rate-limits to five votes a minute.

Add it up and 3.1.1 closes an unauthenticated file write, an arbitrary file delete (twice over), a template-settings overwrite, an authenticated-upload route to code execution, a stored XSS, a leaked API key, and a ratings-abuse hole. "Security Update" covers all of that in two words.

How 3.1.1 closes it

The fix is the right shape, which is reassuring. 3.1.0 trusted the request. 3.1.1 verifies it, and it does so consistently across every action. Here is the complete set of

hardening in the release and what each piece addresses.

Hardening in 3.1.1	What it fixes
Action allow-list, split into admin actions and public actions	Unknown actions get a 403; only `voting` is treated as public
`requireAuthorisedAdminRequest()` checks `core.admin` / `core.manage` and a valid CSRF token on every admin action	Stops unauthenticated guests reaching `save` / `remove` / `import` / font / upload / delete
`requireValidToken()` on the public `voting` action	CSRF protection on the one guest-facing action
`sanitizeLayoutName()` (`basename`, strict `a-z0-9_-` pattern, strips `.json`) plus `getLayoutFilePath()` with a `realpath()` containment check	Kills path traversal in the `save` / `remove` / `load` layout file operations
`getSafeMediaImagePath()` rejects null bytes and `..`, forces an `images/` prefix, and requires the resolved path to stay under the site images folder	Kills the arbitrary file delete in `remove_image`
Upload validation: `File::makeSafe`, `basename`, an extension allow-list (`jpg` / `jpeg` / `png` / `gif` / `webp`) and a `getimagesize()` content check	Stops uploading `.php` or other non-image files (the code-execution vector)
`import` now verifies the template ID exists with `client_id = 0`, validates and round-trips the JSON, and caps it at 1 MB	Stops arbitrary template-styles tampering and oversized payloads
Size caps on layout content (1 MB) and import settings (1 MB)	Denial of service and abuse
`htmlspecialchars` escaping on font output, image output and `data-` attributes, with attribute keys sanitised	Fixes stored and reflected cross-site scripting in the rendered output
Hardcoded Google Fonts API key removed; the key now comes from a template parameter	Removes a leaked credential
Voting rewritten: checks the article exists and is published, tracks votes per session, rate-limits to five votes a minute	Stops ratings spam and votes against arbitrary content IDs

The real issue: telling people to update is not telling them why

We did not go looking for this. We found it while investigating a hacked mySites.guru customer site, traced the compromise to the Helix3 ajax plugin, confirmed the flaw on a clean test install, reported it privately, and asked for the standard 90-day coordinated disclosure window. JoomShaper replied the next morning, were professional and gracious about it, and told us the fix had already shipped. They emailed users an "Action Required" notice and they wrote a changelog entry. On the process, they did more than many vendors do, and they deserve credit for a fast turnaround.

So this is not a story about a vendor who said nothing. It is a story about a vendor who said "update" without saying why, which is a more common and more subtle failure.

Look again at what a Helix3 user actually received. The changelog: "Security Update." The email: "a focused security update we recommend all users apply as soon as possible." Both true. Neither tells you:

- whether this is a minor hardening tweak, or an unauthenticated remote file write
- whether it is being actively exploited right now
- whether you need to patch before you go to bed, or whether it can wait two weeks
- which component is even affected

Put yourself in the seat of someone running 30 client sites on Helix3. "Apply as soon as possible" is what every release email says. With no severity attached, it competes for attention with every other "recommended" update in the queue, and busy people triage by deferring the ones they cannot size. So a critical, unauthenticated file-write fix gets treated like a routine point release, and every deferred site stays exploitable while the patched code sits in public for anyone to diff.

This is not unique to JoomShaper, and that is exactly why it is worth saying out loud. We have written about the same root-cause pattern of AJAX endpoints that check a token but not authorisation more than once, in Novarain, in Astroid, in SP Page Builder,

in PageBuilder CK. The vulnerabilities get found and fixed. The communication keeps landing short.

A security notice does not need to be a full CVE write-up. It needs four things: what was affected, how serious it is, whether exploitation is known, and whether you should act immediately. “Security Update” and “apply as soon as possible” give you only the last one, and a weak version of it. The Joomla ecosystem runs on thousands of small extension and template developers, and most of them are doing their best, but the bar for how a fix gets communicated has to be higher than “trust us, update.” We would like to start that conversation, openly, with the developer community.

What you should do right now

1. **Find every Helix3 install you run.** If you use mySites.guru, search your extension inventory for Helix3 and you have the list in seconds. If you do not, you will need to check each site’s installed extensions by hand.
2. **Update every one of them to Helix3 3.1.1.** The update is live on JoomShaper’s update server and shows in the normal Joomla updater.
3. **Do not confuse Helix3 with Helix Ultimate.** Only Helix3 needs this update.
4. **Check exposed sites for tampering.** Because the bug allowed file writes and deletes, a site left unpatched could already have been touched. Look for unexpected files under your template, missing protective files like `.htaccess`, and unexplained changes to your template style parameters. Our guides on finding hacked files and backdoors and fixing a hacked Joomla site walk through this.

If you would rather not do any of that by hand across a portfolio of client sites, that is precisely the job mySites.guru exists for. One inventory, every site, every framework version, one click to update. Run a free audit and see what is actually installed across your sites.

Disclosure timeline

- **Before disclosure** - While investigating a hacked mySites.guru customer site, we trace the compromise to the Helix3 ajax plugin and confirm the flaw on a clean test install.
- **28 June 2026** - We privately report the unauthenticated file write, arbitrary file delete, and template-parameter overwrite in the Helix3 ajax plugin to JoomShaper, with the proof of concept withheld and a 90-day window requested.
- **29 June 2026** - JoomShaper replies that the issue is already addressed and a patched version (3.1.1) has been released. They note a temporary outage had delayed their usual security announcement.
- **29 June 2026** - JoomShaper emails users an “Action Required: Security Update Now Available for Helix 3” notice and publishes the 3.1.1 changelog. Both recommend updating; neither describes the vulnerability. We forward the exchange to the Joomla security team.

CVE Record

A CVE for this vulnerability is pending, crediting Phil Taylor of mySites.guru as the reporter. We will add the identifier here once it is assigned. In the meantime, the fixed version is the authoritative reference: if you are on Helix3 3.1.1 or later, you have the patch.

Further Reading

- JoomShaper Helix3 framework - the vendor’s product page.
- Joomla com_ajax documentation - how the unauthenticated endpoint works.
- Joomla Vulnerable Extensions List - the community-run list of known Joomla extension issues.
- OWASP: Path Traversal - background on the file-path class of bug.

Frequently Asked Questions

What does Helix3 3.1.1 fix?

A lot more than the changelog suggests. It closes an unauthenticated file write, an arbitrary file delete (via two separate actions), and an unauthenticated template-parameter overwrite in the Helix3 ajax plugin (the onAjaxHelix3 handler). It also blocks the image upload from accepting non-image files like .php, removes a hardcoded Google Fonts API key, adds output escaping to stop cross-site scripting, and rate-limits the article-rating action. We reported these issues to JoomShaper privately, and 3.1.1 is the release that fixes them.

Is Helix3 the same as Helix Ultimate?

No. Helix3 is JoomShaper's original template framework, shipped as a system plugin plus the plg_ajax_helix3 ajax plugin. Helix Ultimate is a separate, newer template with its own codebase and versioning. This vulnerability is in Helix3, not Helix Ultimate. If you run sites on both, only the Helix3 install needs the 3.1.1 update.

Does this affect Joomla 4, 5 and 6, or just Joomla 3?

The Helix3 3.1.1 update manifest declares compatibility with Joomla 4.0 through 6.5, and the patched code contains explicit Joomla 5 and 6 compatibility branches. This is a current-Joomla problem, not a legacy-only one. Any live site running the Helix3 framework should update.

Has this been exploited in the wild?

We found this while investigating a hacked mySites.guru customer site and traced the compromise to the Helix3 ajax plugin, so treat it as a live risk, not a theoretical one. We reported it privately to JoomShaper under coordinated disclosure and they had already shipped a fix. Now that the bug is patched in public, anyone can compare the old and new code to understand it, so the window between the public patch and wider scanning is short. Treat it as urgent.

Why is a two-word changelog a problem?

JoomShaper did announce the update, by email and in their changelog, so this is not a case of a silent patch. The problem is that neither announcement says what was fixed. The changelog reads, in full, 'Security Update', and the email says only 'a focused security update we recommend all users apply as soon as possible'. No severity, no affected component, no indication of whether the issue is being exploited. Thousands of site owners

are left unable to judge whether to patch tonight or wait for their next maintenance window. For a framework that runs unauthenticated code, that is not enough information to act on.

How do I find every Helix3 install across the sites I manage?

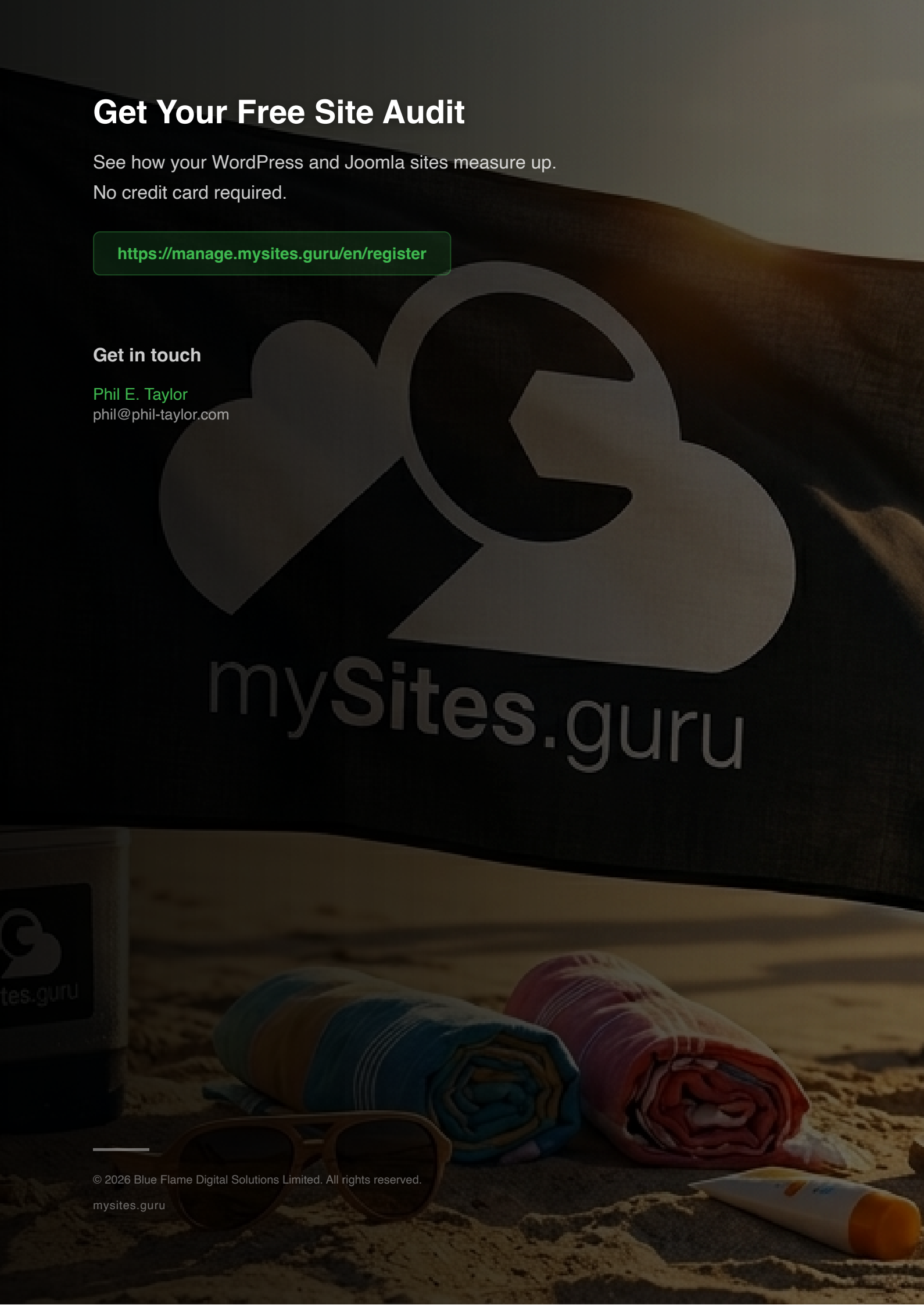
mySites.guru keeps a live inventory of every extension, template and framework on every Joomla and WordPress site in your account. Search for Helix3 once and you get every affected site, the installed version, and a one-click path to push the 3.1.1 update, without logging into each site by hand. Start with a free audit at /free-audit/.

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