mySites.guru

Unauthenticated SQL Injection in DPCalendar found by mySites.guru

mySites.guru found and reported an unauthenticated SQL injection in the DPCalendar Joomla extension's public events feed. Fixed in 10.11.2 and 8.19.4.

Phil E. Taylor | 27 August 2026

Updated 27 August: DPCalendar 10.12.0 is out, and it fixes two further issues

Digital Peak released **10.12.0** and **8.19.5** on 27 August 2026, closing a blind SQL injection and a stored cross-site scripting flaw reported by an independent researcher. Those are separate from the flaw described below, and both need an authenticated user rather than an anonymous one. The fix on this page is still the one to apply first if you are running anything older, because this flaw needed no login at all. See our breakdown of the 10.12.0 release.

DPCalendar is one of the most widely installed calendar and events components for Joomla, chosen for mature, professionally run sites right up to the highest levels of the Joomla world itself. During routine security research on the extensions our customers rely on, **mySites.guru discovered an unauthenticated SQL injection vulnerability in DPCalendar, and reported it to the developers before disclosing anything publicly.** The fix is now available in Digital Peak's security release: DPCalendar **10.11.2** for Joomla 4.4.4 to 6.x, and **8.19.4** for the Joomla 3 branch.

If you run DPCalendar on any Joomla site, update to 10.11.2 now (or 8.19.4 if you are still on Joomla 3). If you manage more than a handful of sites, read on for how to find every affected one at once.

TL;DR

- mySites.guru found an **unauthenticated SQL injection** in DPCalendar and reported it to the vendor
- The flaw dates back to DPCalendar **8.18.0** and is present in every release since, on both the current line and the Joomla 3 branch, across all editions (Free, Pro, Business); fixed in **10.11.2** (Joomla 4.4.4 to 6.x) and **8.19.4** (Joomla 3)
- The public front-end events feed placed an **author filter parameter straight into a database query** without turning it into a number, so an **anonymous visitor could read data from any table**: user accounts, password hashes, the site secrets from

any database table such as draft content, API keys, personal data, ecommerce orders - everything in the database!

- **Update to DPCalendar 10.11.2 immediately** (or 8.19.4 if you are still on Joomla 3)
- We score it **CVSS 4.0 8.7 (High)**, and Digital Peak's own advisory rates it High too. It is tracked as CVE-2026-57831, assigned through the Joomla CNA
- Credit to Digital Peak: they shipped the fix within 24 hours, over a holiday break, and publicly credited us for the find
- mySites.guru already flags every connected Joomla site still running a vulnerable version, so you do not have to check each site by hand

mySites.guru discovered this security issue and reported it to Digital Peak before publishing details. We withheld the exact request and any proof-of-concept until a fix was available and site owners had a reasonable window to update. This is how we handle every vulnerability we find: fix first, publish second.

What is the vulnerability?

DPCalendar exposes a front-end events feed that anonymous visitors can reach without logging in. It is the same feed the calendar itself uses to draw events, reached at `index.php?option=com_dpcalendar&view=events&format=raw`. That feed accepts an author filter, `filter_created_by`, and when it is supplied as a plain value the extension placed it into the events query **without casting it to an integer first**, the textbook shape of a SQL injection.

Joomla's standard text filter, which the extension applied to that parameter, strips HTML but does nothing to neutralise SQL syntax. Quotes, parentheses and keywords all survive. It is the same public-endpoint-meets-unchecked-input pattern behind a long run of recent Joomla extension vulnerabilities.

So an attacker could supply a crafted author filter that turned the query into one reading from any table in the Joomla database, and infer the result from the response. No account needed, no CSRF token, nothing stolen first.

In practice that means an unauthenticated attacker could read:

- Joomla user accounts, including usernames and password hashes
- The Joomla configuration secret and other stored data
- Event, content and extension data

We are not publishing the exact request or a working proof-of-concept. The point of this post is to get people updated, not to hand attackers a recipe.

It is a blind injection, and that still means full database read

There are two broad flavours of SQL injection. In the loud kind, the query result comes straight back in the page, and an attacker dumps a table in one request. This one is the quieter kind: a **blind** injection. The response does not print the data. Instead, the attacker asks the database true-or-false questions one at a time (is the first character of the admin hash greater than `m`?) and reads the answer from whether the events feed changes.

Do not let “blind” read as “limited”. Blind injection is slower to exploit, not smaller in what it exposes. Automated tooling walks a database out one bit at a time in seconds of wall-clock. The end state is identical to the loud kind: the full contents of every table an anonymous request can reach, including the credentials table. Blind is a throughput detail for the attacker, not a safety margin for you.

If you have not met blind SQL injection before, here is a short explainer from Indusface that walks through the exact mechanics described above. Yes, it is someone else’s video. Just like we always do, copying things from other people all the time:

Watch video: https://www.youtube-nocookie.com/embed/8N7Z1aHFSIQ

One character per request is not a safety margin

Digital Peak's advisory is candid about the mechanics, and one point in it is worth addressing directly. It notes that the data comes out "one character per request", and frames that as making it impractical to pull entire tables. That describes exactly how a blind injection works. It does not describe a limit on the damage.

The character-at-a-time loop is the definition of blind SQL injection, not a cap on it. Every blind injection ever found has worked this way, and every one has still ended with the full database in the attacker's hands. The arithmetic is not reassuring: a bcrypt password hash is about sixty characters, a binary search reads each character in roughly seven requests, so a complete Super User hash is a few hundred requests, and a live server answers those in a couple of minutes. "A large number of requests" sounds like friction. At machine speed it is a coffee break.

The severity does not move with the request count. Whether the database is read in one request or one million, the end state is identical: every user account, every password hash and other Joomla API secrets, all readable by an anonymous visitor. Confidentiality is either intact or it is gone, and after a blind injection it is gone. That is why both we and Digital Peak rate this High rather than shrugging it off with **<sarcasm>This is fine.</sarcasm>**

You have never needed AI to weaponise blind SQL injection

The advisory also mentions that AI-assisted tools can speed the extraction up. That is true, and it is also beside the point, because the speed-up it describes arrived in 2006, not in the 2020s.

Fully automated, character-by-character blind SQL injection has been free, off-the-shelf tooling for about twenty years. sqlmap, the standard open-source tool, has automated boolean-based and time-based blind extraction and dumped entire databases since July 2006. Point-and-click extractors put the same power in front of absolute beginners even earlier: Absinthe demonstrated automated blind dumping at

Black Hat in 2004, and Havij made it a one-button affair by 2010. The underlying technique was documented publicly by Chris Anley in 2002 and popularised in a dedicated SPI Dynamics whitepaper the following year. All of it predates modern AI by fifteen to twenty years.

A blind injection a human would find tedious is a single sqlmap command that has been public since before a lot of today's extension code was even written. AI did not make blind SQL injection practical. sqlmap did that two decades ago. Casting AI as the thing that turns "one character per request" into a real threat gets the timeline backwards.

Joomla knows this class of bug intimately, and not only from its extensions. Its own core has shipped serious SQL injection flaws, and they did real damage long before anyone had an AI assistant:

- **October 2015, `com_contenthistory` (CVE-2015-7857 and related).** An unauthenticated SQL injection in Joomla core, in every release from 3.2.0 to 3.4.4, let an anonymous attacker steal a live Super User session and take the whole site over. It was error-based rather than blind, which sharpens the point rather than softening it, and it was being exploited in the wild within 24 hours of the fix, against a component that shipped enabled by default on millions of sites.
- **May 2017, `com_fields` (CVE-2017-8917).** An unauthenticated SQL injection in Joomla 3.7.0, through an ORDER BY parameter that public exploits drove both blind and error-based, exposing session tokens and password hashes. It was weaponised with public exploit code within days.

None of those needed AI. SQL injection has been fully compromising Joomla sites, core and extensions alike, for well over a decade, with tools any teenager could download for free. The stakes were never raised by AI. They were always this high, and the only reliable answer has always been the same one: update the affected code.

How serious is it?

Serious enough that we treated it as a priority disclosure. The two things that decide real-world impact are:

1. **Whether the site has a web application firewall that filters SQL.** Because the payload is literally SQL, a WAF often recognises and blocks it before it reaches DPCalendar. Sites running bare, with no such protection, are the exposed ones.
2. **Whether the calendar has any publicly visible events.** The blind oracle in this flaw reads against the events the feed returns, so a live, in-use calendar (which is every real one) gives an attacker exactly what they need. An empty calendar has nothing to read against, though that is not a state any working site stays in.

One more thing worth saying: this is not obscure code running on neglected sites. DPCalendar gets chosen because it is mature and well-regarded, and it runs on some of the most reputable, carefully managed Joomla installations there are, up to and including the Joomla Project's own official infrastructure. One of the Project's own `joomla.org` sites was itself running an affected version, shielded from this exact flaw only by the web application firewall in front of it, doing the SQL-filtering job described above. That is a double lesson: it shows how trusted and well-regarded DPCalendar is, chosen for Joomla's own properties, and it shows that being trusted is not the same as being safe. If the Joomla Project itself can be caught running a vulnerable copy of an extension this well-regarded, anyone can. "We only use trusted, well-maintained extensions" is a good instinct, and on its own it is not a defence against a flaw in one of them. The most buttoned-up site in the world is still exposed if it is a version behind on the extension that has the hole.

A WAF is mitigation, not a fix. The only reliable remedy is to update DPCalendar. Or leave it a version behind and hand your entire database to the first anonymous stranger who asks. Sure. Go ahead. (Or is that *Is Not Passive-Aggressive Enough*?)

> This is the case for defence in depth. A SQL-aware firewall in front of a vulnerable extension can catch this class of attack before it reaches the code, buying you time to update. It does not mean you can skip updating. Treat the firewall as the seatbelt and the patch as not crashing the car: you want both.

Would a long, strong .htaccess have protected me against this?

Almost certainly not. A big hardened `.htaccess` is the security blanket a lot of Joomla owners reach for, and it guards a different kind of attack than this one. The exploit here is an ordinary, allowed request to `index.php` (`option=com_dpcalendar&view=events&format=raw`), the exact request your `.htaccess` exists to let through so the site works at all. The dangerous part is not the URL, it is the SQL hidden inside one query-string value, and a stock `.htaccess` never looks there.

To block this at the `.htaccess` layer you would need explicit rules that pattern-match SQL syntax in the query string and return a 403. Hardly anyone runs those, they are brittle, and they tend to block real visitors as often as attackers. Inspecting request input for SQL is really the job of a web application firewall with SQL injection filtering switched on. Even then it only helps if that filtering is actually enabled: a WAF with its SQLi rules turned off, or left in log-only mode, waves this through as happily as no WAF at all. Protection you have to remember to configure is not the same as code that is not vulnerable in the first place.

The short version applies here directly. Update DPCalendar.

Which versions are affected?

Digital Peak's advisory states the flaw has existed since **DPCalendar 8.18.0**, so every release from 8.18.0 up to the current 10.11.0 is affected, and so is the **Joomla 3 branch** (the 8.x line still shipped for Joomla 3 sites). The vulnerable code sits in the component's shared front-end model, which is identical across editions, so **the Free, Pro and Business editions are all affected**. This is not a Free-only issue. The fix is in **10.11.2** for Joomla 4.4.4 to 6.x, and **8.19.4** for Joomla 3, both casting the author filter to an integer before it reaches the query so a crafted value can no longer change the SQL. If you are on the 9.x line, Digital Peak's upgrade path is 9.x to 10.0.0, then 10.6.0, then 10.11.2.

If you run any version of DPCalendar earlier than 10.11.2 (or earlier than 8.19.4 on Joomla 3), on any edition, assume your site is affected and update now. Do not wait to confirm exploitation: by the time you can confirm it, the data is already gone.

Credit where it is due: a fast fix over a holiday

We gave Digital Peak the full standard responsible-disclosure window to fix this in their own time. They did not need it. We reported the issue on 12 July; the fix was built, tested and released within 24 hours, over a holiday break while the developer was away, and outside Digital Peak's normal Thursday release schedule.

That is a good response, and worth saying out loud. Plenty of vendors sit on a report for weeks. Digital Peak did the opposite, and DPCalendar users are safer for it. They even credited mySites.guru publicly in their own advisory and recommended the platform, a gracious touch we did not ask for. Our thanks to Allon Moritz and the Digital Peak team.

Digital Peak is not the exception this month either. Most of the developers we have responsibly disclosed to during this run have responded the same way: grateful, quick to thank us, and shipping a fix within 24 hours. Only two out of the whole set have not. When a vendor gets this right, it deserves to be said in public as as the flaw itself.

The right thing for site owners to do in return is simple: apply the update they shipped.

SQL injection is a recurring theme right now

This DPCalendar flaw is not an isolated find. It is one of several unauthenticated SQL injections we have uncovered in widely used Joomla extensions in a single research run, and they all share one root cause: a public front-end endpoint that takes anonymous request input and hands it to a database query without turning it into a safe value first.

- AcyMailing (CVE-2026-56292, CVSS 8.7). The same class of unauthenticated SQL injection in one of the most popular newsletter extensions for Joomla, and because it shares a codebase, the WordPress plugin too. Fixed in 10.11.1.
- The month of Joomla disclosures. DPCalendar is one entry in a larger run of vulnerabilities we found and reported across popular Joomla extensions in a single month, several of them this exact SQL injection shape.
- Why AJAX endpoints are a CMS security blind spot. The underlying pattern, and why front-end feeds like DPCalendar's keep producing this bug.

Two more disclosures from the same research run are still working through coordinated disclosure, redacted until their vendors ship a fix. We will publish each in full the moment it is patched, and add it to the roundup. Naming an unpatched flaw is just publishing an exploit, so until then we describe only the risk, not the extension.

How do you update DPCalendar safely?

1. **Take a backup first.** Before any extension update on a production Joomla site, back up the database and files. If you use mySites.guru, trigger a snapshot or a full backup so you have a clean starting point.
2. **Update through Joomla's Extensions manager, or in bulk from mySites.guru.** On a single site, open the Joomla administrator, go to System, then Update, then Extensions, and let Joomla pull DPCalendar 10.11.2 (or 8.19.4 if the site still runs Joomla 3). If it does not appear, use Find Updates, or download the latest package from your Digital Peak account and install it over the top. On the older 9.x line, Digital Peak's upgrade path is 9.x to 10.0.0, then 10.6.0, then 10.11.2. If you manage more than one site, use the mySites.guru mass update feature to upgrade DPCalendar across every affected site from a single dashboard. You can even enable auto-updates for any Joomla extension so the next fix like this reaches your sites without you lifting a finger.
3. **Confirm the version.** After updating, open Components, then DPCalendar, and check the version shown is 10.11.2 (or 8.19.4 on Joomla 3) or later.

4. **Clear caches.** Clear Joomla's cache and any CDN or page cache so stale front-end assets do not linger.

Updating closes the door. It does not undo any data an attacker may already have read, so if you handle sensitive data and were on a vulnerable version for a long time, treat credentials and any exposed data as potentially known.

How do I find every DPCalendar site I manage?

The first question after any extension security release is the awkward one: which of my sites actually run this? Up to about ten sites, you can log in to each Joomla admin and check the installed extensions list. Past that, you need a single view.

mySites.guru keeps a live inventory of every extension, template and framework on every Joomla and WordPress site in your account. You search for DPCalendar once and get back every connected site running it, the version each one is on, and whether an update is available. No logging into forty admin panels one at a time.

View every DPCalendar install across your sites

Open your Extension Inventory

Search for DPCalendar across every connected Joomla site and filter for anything on 10.11.0 or earlier to find the installs that still need updating. Not a subscriber? Sign up free and connect your sites.

Once you know which sites need it, the mass updater handles the rollout: tick the sites on an old version, push the update to all of them from one screen.

For agencies managing dozens of Joomla sites

The patch is the easy bit. Knowing which client sites run DPCalendar, and getting the update onto all of them, is the work. See how mySites.guru manages multiple Joomla sites from one screen.

Why calendar and events extensions are a quiet attack surface

This is a pattern, not a one-off, and it is why the class of bug matters more than this single instance. Calendar, form and newsletter extensions make attractive targets for the same reason they are useful: they expose front-end endpoints that anonymous visitors are meant to reach. Event listings, subscribe forms, iCal feeds, location lookups, none of that can require a login.

Every one of those public endpoints is attack surface. When the code behind them trusts request input, or hands it to a database query without escaping or casting, an anonymous visitor becomes an anonymous attacker. We have seen the same class of issue across form builders like Balbooa Forms, page builders like PageBuilder CK, template frameworks like Novarain, newsletter tools like AcyMailing, and editors like JCE: a public task that reaches a dangerous operation with too little checking in between.

The lesson for site owners is not “avoid calendar extensions”. They are fine, and the reputable ones are well maintained. The lesson is that **keeping them updated is security-critical, not just feature maintenance**. An extension two versions behind is not a cosmetic problem.

Stay Ahead of the Next One

This is one extension on one day. There will be another, because Joomla runs on thousands of third-party extensions and the ones that accept input from anonymous visitors keep producing bugs like this. The hard part is never the update itself. It is knowing a fix exists, knowing which of your sites are affected, and getting to them before an attacker does, across every extension on every site you look after.

That is the job mySites.guru does for you. It keeps a live inventory of every extension on every Joomla and WordPress site in your account, flags the ones with a known vulnerability, and lets you push the update to all of them from one screen. When

something like this DPCalendar flaw is disclosed, you see exactly which sites are exposed in seconds, without opening a single admin panel to check.

Get free email alerts when a Joomla vulnerability breaks

We email a plain-English alert the moment a serious flaw like this one is disclosed, with the affected versions and what to do. No charge, unsubscribe any time.

Subscribe to security alerts

Want the alerts and the tooling to act on them? Start with a free audit on one site and see your full extension inventory, or sign up for mySites.guru to get vulnerability alerts and one-click updates across every site you manage.

What to do right now

- Update every Joomla site running DPCalendar to **10.11.2** or later (or **8.19.4** on Joomla 3), one at a time or in bulk from one dashboard
- If you manage multiple sites, let mySites.guru show you which ones are still exposed rather than checking by hand
- Take a backup before updating, and clear caches afterwards
- If you were on a vulnerable version for a long time and handle sensitive data, treat any exposed data as potentially read, and if you suspect a breach, find any hacked files and backdoors and follow our guide to fixing a hacked Joomla or WordPress site

Disclosure and Severity

This flaw is CWE-89, SQL injection, reached by an anonymous visitor over the network in a single request, with no privileges and no user interaction. It is a read-only injection, so it stops short of the maximum score: an attacker reads the database but cannot

write to it or run code through this flaw. Reading the database is bad enough, because what leaks is every user record, every password hash, and other Joomla API secrets.

8.7
CVSS 4.0

HIGH **Our own assessment, pending the official vector**

Unauthenticated, exploitable over the internet with no user interaction, ending in full read access to the site database. It falls short of 10.0 only because it is a read, not a write: high confidentiality impact, but no integrity or availability impact.

- No login needed
- Exploitable over the internet
- No user interaction
- Full database read
- Password hashes exposed

This vulnerability is tracked as CVE-2026-57831, assigned through the Joomla CNA, the body that assigns identifiers for Joomla and its extensions, crediting Phil Taylor of mySites.guru as the reporter.

Field	Detail
CVE	CVE-2026-57831 (assigned via the Joomla CNA)
Component	DPCalendar for Joomla (`com_dpcalendar`), all editions (Free, Pro, Business)
Vendor	Digital Peak (digital-peak.com)
Type	Unauthenticated SQL injection (CWE-89)
CVSS 4.0	8.7 (High), `AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N` (our own assessment)
CWE	CWE-89 (Improper Neutralization of Special Elements used in an SQL Command)
Impact	Anonymous read access to any database table, including user accounts and password hashes
Finder	Phil Taylor, mySites.guru
Affected versions	8.18.0 up to and including 10.11.0 on the current line, plus the Joomla 3 (8.x) branch, all editions

Field	Detail
Fixed in	10.11.2 (Joomla 4.4.4 to 6.x) and 8.19.4 (Joomla 3)

The disclosure ran on a short cycle from audit to fix, and the full timeline is at the foot of this post.

Timeline

2026-07-12

We find the injection and report it privately

Routine security research on the extensions our customers rely on turns up the unauthenticated SQL injection in DPCalendar. We confirm it against the code, prove it on a local install that reads database contents back through a single anonymous request, and disclose it to Digital Peak with all public detail withheld.

2026-07-13

Digital Peak ships the fix within 24 hours

DPCalendar 10.11.2 for Joomla 4.4.4 to 6.x and 8.19.4 for Joomla 3, casting the author filter to an integer, released over a holiday break and outside their normal Thursday schedule. They publish a security advisory crediting mySites.guru. We verify the patched build and flag every connected site below the fix.

2026-07-15

The Joomla CNA assigns CVE-2026-57831

Crediting Phil Taylor of mySites.guru as the finder.

Further Reading

- This DPCalendar flaw was one of a month of Joomla extension vulnerabilities we found and disclosed - the full roundup of the run.

- Our AcyMailing SQL injection disclosure - the same class of bug, in a different extension, found in the same period.
- Why AJAX endpoints are a CMS security blind spot - the recurring public-endpoint pattern behind flaws like this one.
- How to check your Joomla database security - what to review and rotate if a vulnerable extension was ever live.
- CWE-89: SQL Injection and OWASP: SQL Injection - the canonical references for this weakness class.
- Digital Peak's own security advisory - the vendor's write-up of the fix, crediting mySites.guru.
- DPCalendar official site and downloads - the vendor's product page, where 10.11.2 (and 8.19.4 for Joomla 3) is available.

Frequently Asked Questions

What is the DPCalendar SQL injection vulnerability?

It is an unauthenticated SQL injection in DPCalendar, the calendar and events component for Joomla by Digital Peak. The public front-end events feed took an author filter parameter and placed it into a database query without turning it into a number first, letting an anonymous visitor inject conditions and read data from any table in the site's database. mySites.guru discovered the issue and reported it to the vendor. It is fixed in DPCalendar 10.11.2 for Joomla 4.4.4 to 6.x, and 8.19.4 for Joomla 3.

Do I need to log in to Joomla for an attacker to exploit this?

No. The affected events feed is reachable by anonymous visitors, with no account, no stolen credentials, and no CSRF token. That is what makes it serious. The main real-world limits are whether the site sits behind a web application firewall that filters SQL, and whether the calendar has any publicly visible events for the attack to read against.

Which DPCalendar versions are affected?

The flaw was introduced in DPCalendar 8.18.0 and is present in every release since, on both the current line (up to and including 10.11.0) and the Joomla 3 branch. The vulnerable code lives in the shared front-end model, so all editions (Free, Pro and Business) are affected, not just the Free edition. Digital Peak fixed it in 10.11.2 for Joomla 4.4.4 to 6.x, and 8.19.4 for Joomla 3. If you run any earlier version, update now.

How do I know if my site is affected?

If you run DPCalendar on a version earlier than 10.11.2 (or 8.19.4 on Joomla 3), assume you are affected and update. mySites.guru flags every connected Joomla site running a vulnerable version automatically, so you get told which sites need attention rather than checking each one by hand.

Was this exploited in the wild before the fix?

We have no evidence of exploitation before disclosure. Because the payload is SQL, a web application firewall that filters SQL blocks it. We reported the issue to the vendor and withheld the exact request and any proof-of-concept, so the details were not public before the fix shipped.

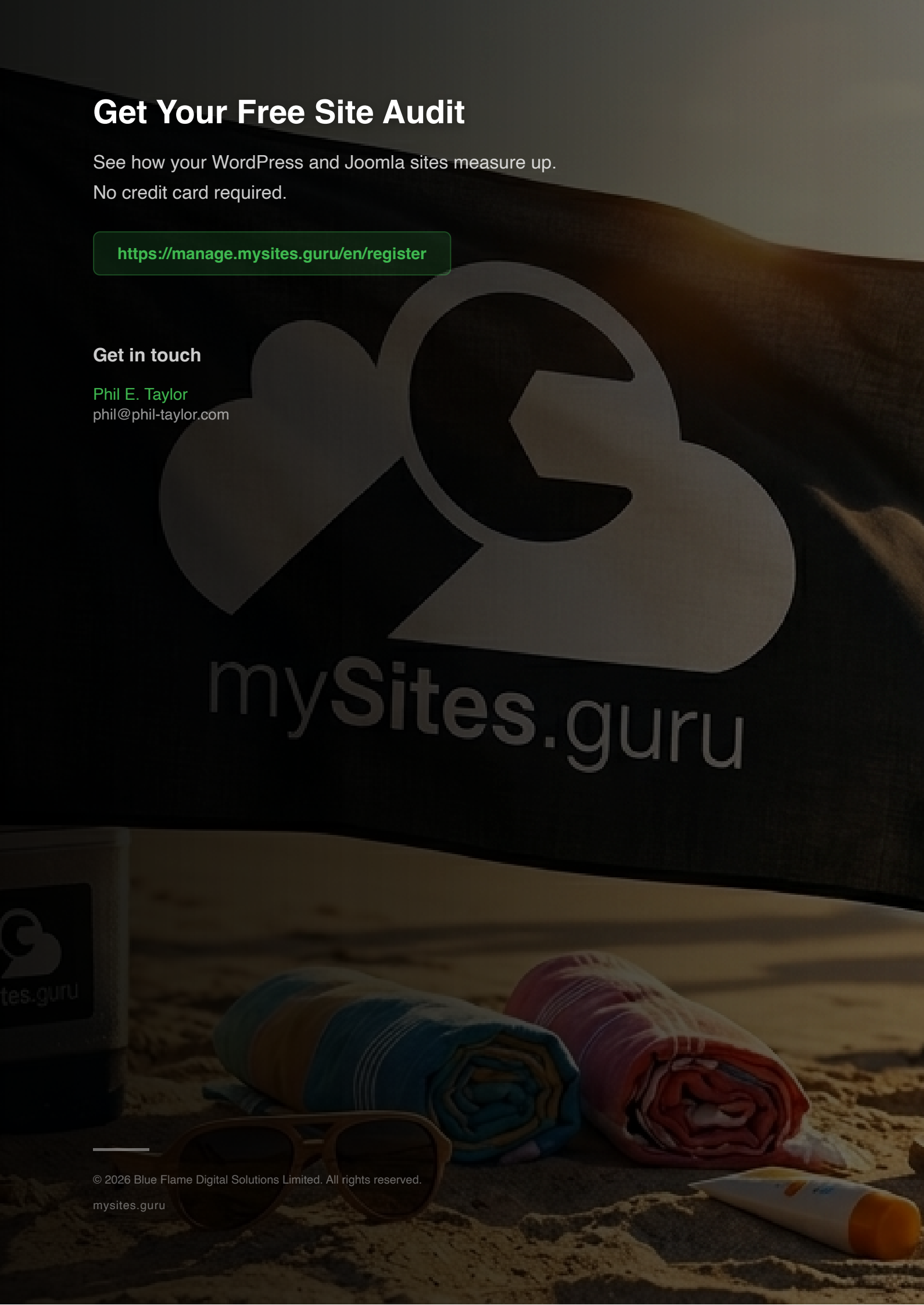
Written and edited by a human, with AI assistance. Our approach to AI

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.