mySites.guru

Phoca Download 6.1.3 Fixes an Authenticated Upload RCE

Phoca Download for Joomla (com_phocadownload) up to 6.1.2 let a logged-in member upload a PHP file and run code on the server. Fixed in 6.1.3, update now.

Phil E. Taylor | 26 August 2026

Phoca Download is a long-established download-manager component for Joomla, installed as `com_phocadownload` and used to store, organise and serve files on business, membership and community sites. Up to and including version 6.1.2, its frontend member-upload feature had a flaw: the file-type allow-list that the extension enforces on its other upload paths was skipped for member uploads. A logged-in user could upload a file type that should have been rejected, such as a `.php` script, into the public user-upload folder and then run it. That is authenticated remote code execution.

Phoca fixed it in version 6.1.3, released on 10 July 2026. If you run Joomla sites with Phoca Download on them, update every one to 6.1.3 or later. This post explains what the flaw was, how it surfaced, what the fix does, and how to check whether any of your sites were touched, without publishing anything an attacker could copy.

> **Authenticated and feature-gated, but still update promptly**
>
> The Joomla CNA scored this Critical (CVSS 4.0 9.0), because it ends in code execution. Unlike the recent no-login Joomla upload flaws, though, it needs a registered account and needs the Phoca Download user-upload feature switched on, so far fewer sites are reachable in practice. If your Phoca Download sites let members upload files, treat it as a priority: update to 6.1.3, then check the user-upload folder and your admin accounts. If member uploads are off (the default), you are not exposed to this path, but you should still update.

TL;DR

TL;DR: The frontend member-upload feature in Phoca Download (`com_phocadownload`) up to and including 6.1.2 (fixed in 6.1.3) skipped the file-type allow-list that the extension enforces on its other upload paths. Because of an internal upload-mode mismatch, the admin-configured list of permitted types was never consulted for member uploads, so a logged-in user could write a `.php` file into the public user-upload folder and execute it, which is authenticated remote code execution (CWE-434). It needs a registered account and needs the non-default user-upload feature enabled with a category granting upload rights, which is why the Joomla CNA scored it 9.0 (Critical) on CVSS 4.0 rather than the maximum 10.0 the recent unauthenticated flaws

hit. We found it during a source-code audit, reproduced the full chain on a local Joomla install, and disclosed it privately to Phoca, who fixed it the same day in 6.1.3. It is now tracked as CVE-2026-57828, assigned via the Joomla project's CNA and crediting Phil Taylor of mySites.guru. No proof of concept has been made public. Update every Phoca Download install to 6.1.3, then check any site that had member uploads enabled. If you look after more than a handful of Joomla sites, mySites.guru lists every Phoca Download install in your account in seconds and pushes the update.

What Was Actually Wrong in Phoca Download?

Phoca Download lets a site optionally accept files from logged-in members through the frontend, so a community or membership site can let its users submit downloads. That feature is off by default. When it is switched on, uploads should be checked against the same allow-list of permitted file types that the extension uses on its other upload paths, the list an administrator configures to allow documents, archives and images but not executable code.

The problem was that the member-upload path used a different internal upload mode from the one the allow-list check was written for. The check only applied to one mode, and the member-upload page ran under another, so the configured allow-list was never consulted. On a default configuration the practical result was that the file-type restriction that should have blocked a `.php` upload simply did not run for member uploads.

From there the rest of the chain is short. The file lands in Phoca Download's user-upload folder, which sits inside the web root and serves and executes PHP. A logged-in member could upload a file the extension was supposed to reject, then request it in a browser, and the server would run it. That is code execution on your server, chosen by whoever uploaded the file.

We are not publishing the exact request an attacker would send, the task names, or the parameters involved. The shape of the bug is enough to understand the risk and to check your own sites. The working details stay private until sites have had time to update.

How This One Was Found

This did not start with a hacked site or an abuse report. It started with reading the code.

Part of what we do is audit the Joomla extensions that turn up most often across the sites we manage, on a local test install where we can prove a flaw safely and put the box back exactly as it was. Phoca Download came up in that rotation. Reading through the member-upload path, the allow-list check was guarding one upload mode while the member-upload page ran under a different one, so the check never fired where it mattered most.

We reproduced the whole chain end to end on a local Joomla install: a registered member, with the user-upload feature enabled, uploaded a file that the allow-list should have rejected, and requesting that file back ran it. Then it was a private email to Phoca. Jan Pavelka, the author, responded and shipped the fix in Phoca Download 6.1.3 the same day, which is exactly the response you want. The flaw is now tracked as CVE-2026-57828, assigned through the Joomla project's CNA and crediting Phil Taylor of mySites.guru as the reporter.

> **This is what responsible disclosure looks like**
>
> Found in a code audit, confirmed live on a local install, reported privately, and fixed by the author the same day. No exploit has been published, and none will be until sites have had a chance to update. The reason to write about it is to get sites patched, not to show anyone how it worked.

Critical on Paper, Gated in Practice

It is worth being precise about severity, because it is easy to lump every "file upload to RCE" bug into the same panic. The Joomla CNA scored this Critical, and rightly so on paper: it ends in code execution, which compromises the whole server. But it is not the internet-wide emergency the recent unauthenticated Joomla flaws were, because getting to that code execution is gated behind preconditions those flaws did not have.

Three things have to be true before this is exploitable, and none of them is the default:

- The site must have the Phoca Download user-upload feature switched on. This means an administrator has added a User Upload menu item, set “Enable User Upload” to Yes, and is letting members submit files. Out of the box, all of this is off.
- A Phoca Download category must grant upload rights to registered users.
- The attacker must have a registered account on the site that those upload rights apply to.

Compare that with the RSFiles unauthenticated upload RCE, the Page Builder CK flaw, or the Balbooa Forms flaw, all of which needed no account at all and were reachable by anyone on the internet in a single request. Those scored the maximum 10.0. This one is scored 9.0 (Critical) on CVSS 4.0, a notch lower, because the login and non-default feature it needs are reflected in the vector (`AT:P`, `PR:L`). The practical difference is exposure: the unauthenticated flaws are reachable by anyone on the internet, while this one needs a registered account and a feature most sites never switch on. Where the preconditions are met, though, the outcome is identical: a member gets to run code on your server, so it still needs patching.

What the Phoca Download 6.1.3 Fix Does

Version 6.1.3 applies the file-type allow-list to member uploads, the same way it always applied it elsewhere. The member-upload path now runs the check that decides whether a given file type is permitted, so an upload of a type the administrator has not allowed, such as a `.php` script, is rejected before anything is written to disk. The only way to skip the allow-list now is for an administrator to choose an “allow all file types” setting, which is an explicit choice to disable a safety net rather than a silent default.

The practical takeaway is simple: 6.1.3 closes the gap that let a member upload a file type the site was supposed to reject. Every Phoca Download install on 6.1.2 or earlier, with member uploads enabled, is on the wrong side of that gap until it is updated.

How Do I Get Phoca Download 6.1.3 Right Now?

The fix is Phoca Download 6.1.3. A brand-new release takes a short while to propagate everywhere, so at the time of writing you may see the earlier 6.1.2 in a couple of places while 6.1.3 catches up:

- Phoca published 6.1.3 on its official GitHub releases on 10 July 2026. That is the authoritative source for the patched package right now.
- The phoca.cz download portal and the in-Joomla update channel may still be offering 6.1.2 for a short period as the new release rolls out.

So the one thing to check is the version number. Whether you update from the Joomla updater, the phoca.cz portal, or GitHub, confirm you are actually installing 6.1.3 or later and not 6.1.2. If your usual update route still shows 6.1.2, either grab 6.1.3 from Phoca's GitHub releases or give it a little time and re-check. Phoca is expected to publish its own release notes alongside the update.

How mySites.guru Catches This Without a Signature

If you already run sites through mySites.guru, this is the part worth knowing. We did not need a Phoca-specific rule to catch this class of activity. mySites.guru watches for the behaviour, an unexpected executable file being written into a public folder, rather than a fingerprint of one particular extension's bug.

That generic detection is the same logic that catches RSFiles, Page Builder CK, iCagenda and SP Page Builder upload attempts. An executable file dropped through an upload looks the same regardless of which extension left the door open, so we can catch new bugs of this shape before anyone has written a rule naming them. This is the same lesson as the AJAX and frontend endpoints blind spot: an endpoint that accepts input from users has to treat every input as hostile.

If a hostile file does land, the suspect content tool and hacked-file detection and the backdoor scanning find it across every connected site, matching it against known malware hashes and thousands of code patterns. Anything flagged can be sent for AI-powered malware analysis that explains in plain English what the file does.

How Do I Find Every Phoca Download 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 Phoca Download once and get back every connected site running it, the version each one is on, and whether an update is available. It is the same workflow as finding every site running a vulnerable JCE: search once, see everything, no logging into forty admin panels one at a time.

View every Phoca Download install across your sites

Open your Extension Inventory

Search for Phoca Download across every connected Joomla site and filter for anything on 6.1.2 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 and push the update to all of them from one screen. You can also switch on automatic updates for any Joomla extension so future Phoca Download releases land without you lifting a finger.

How Do I Check a Phoca Download Site for Tampering?

Updating closes the gap. It does not tell you whether anyone walked through it first. Because the flaw allowed a member to drop a file and run it, a site that had member uploads enabled and was on 6.1.2 or earlier could already have been touched, so check before you assume it is clean.

Three checks, in order of value:

1. Look in the Phoca Download user-upload folder. Anything there that is not a real member upload, and above all anything ending in `.php`, is a red flag. On a single site you can run `find` over that folder for `.php` files via SSH.
2. Check for rogue administrator accounts. Code execution is a fast route to a hidden Super User. In your Joomla Users list, sort by registration date and treat any administrator account you do not recognise as suspect.
3. Hunt for modified and unfamiliar files. A foothold is a perfect moment to plant persistence elsewhere. Look for recently changed PHP files and for executable files sitting where uploads should not contain code.

mySites.guru runs all three of these across every connected site at once. The suspect content tool and hacked-file detection surface the stray PHP files, the extension and user inventory sorts every account across every site by registration date, and real-time alerting tells you the moment a new file appears or an unfamiliar admin logs in, rather than at the next manual check.

If any of that turns something up, the Joomla hacked recovery guide and the how to fix a hacked site walkthrough cover the cleanup, and fix.mysites.guru is the done-for-you option if you would rather hand it over.

File Uploads Are a Recurring Joomla Attack Surface

If you manage Joomla sites, treat every extension that accepts files as a piece of your attack surface that deserves regular attention. The exact weakness fixed here, an upload of a file with a dangerous type slipping past a check that should have stopped it, is CWE-434, and it shows up across the whole category rather than in one vendor's code.

The recent run of Joomla disclosures makes the point. RSFiles, Page Builder CK, iCagenda, Balbooa Forms and the SP Page Builder zero-day were all variations on the same theme: an endpoint that took a file without a strict check on what was actually written. Phoca Download's version of it needed a login and a non-default feature, which makes it less exposed, but the underlying lesson is identical. A single allow-list

check on the file that is actually written, applied on every path that writes one, is what stops this.

Credit where it is due: Phoca did the thing that matters most, which is fix it the same day it was reported. Phoca Download has no CVE of its own on record before this, and its history of security reports is short, so this is not a pattern of problems. It is one bug, found by reading the code, and closed quickly.

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 files 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 Phoca Download flaw lands, you see exactly which sites run it in seconds instead of logging into forty admin panels to find out. And because the monitoring watches for the behaviour rather than a signature, it catches new upload attacks of this shape before anyone has written a rule naming them.

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.

Disclosure and Severity

This flaw is CWE-434, unrestricted upload of a file with a dangerous type. It is reachable over the network by a registered user, with no user interaction, on a site that has the non-default Phoca Download user-upload feature enabled, and it ends in remote code execution. The Joomla CNA scored it Critical (CVSS 4.0 9.0); the account and non-default-feature preconditions are what keep its real-world exposure below the unauthenticated 10.0 flaws.

9.0
CVSS 4.0

CRITICAL

Critical score, gated by a login and a non-default feature

The Joomla CNA scored it 9.0 (Critical) because it ends in remote code execution, which compromises the whole server. It stops just short of the 10.0 the recent RSFiles, Page Builder CK and Balbooa Forms flaws hit, because it needs a registered account and needs the user-upload feature switched on rather than being reachable by anyone with no login. Those preconditions are why far fewer sites are exposed in practice.

- Registered account required
- Needs the user-upload feature enabled
- No user interaction
- Remote code execution

This flaw is tracked as CVE-2026-57828, assigned through the Joomla project's CNA and crediting Phil Taylor of mySites.guru as the reporter.

Field	Detail
CVE	CVE-2026-57828
Component	Phoca Download (`com_phocadownload`)
Vendor	Phoca (Jan Pavelka)
Type	Authenticated arbitrary file upload to remote code execution
CVSS 4.0	9.0 (Critical), `AV:N/AC:L/AT:P/PR:L/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H`
CWE	CWE-434 (Unrestricted Upload of File with Dangerous Type)

Field	Detail
Preconditions	Registered account, user-upload feature enabled, category granting upload rights
Finder	Phil Taylor, mySites.guru
Affected versions	Up to and including 6.1.2
Fixed in	6.1.3, released 10 July 2026

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

Timeline

2026-07-10

A source-code audit turns up the member-upload path

Phoca Download's member-upload path skips the file-type allow-list. We reproduce the full chain on a local Joomla install: a registered user with the user-upload feature enabled uploads a file that should be blocked and executes it.

2026-07-10

We disclose it privately, and Phoca ships 6.1.3 the same day

Jan Pavelka responds and releases Phoca Download 6.1.3, enforcing the allow-list on member uploads. A CVE is requested through the Joomla CNA. No proof of concept is published.

2026-07-11

CVE-2026-57828 is assigned

Crediting Phil Taylor of mySites.guru as the finder.

Further Reading

- This joined a dozen Joomla extension flaws we reported in a single month - the full roundup.
- Phoca Cart 5.2.4, 6.1.7 and 4.0.13 fix a front-end SQL injection - a separate Phoca extension, a separate flaw, and a reminder to check every Phoca component you run, not just this one.
- Phoca Download releases on GitHub - the authoritative source for the patched 6.1.3 package while it propagates to the download portal.
- CWE-434: Unrestricted Upload of File with Dangerous Type - the canonical definition of this weakness class from MITRE.
- OWASP File Upload Cheat Sheet - the developer's checklist for accepting uploads safely: allow-lists, signature checks, renaming, and storing outside the web root.
- PortSwigger Web Security Academy: File upload vulnerabilities - a neutral, in-depth reference on how these flaws work and how to prevent them.
- Securing Joomla extensions - Joomla's own guidance for developers, including safe file handling with `JFilterInput` and `JFile::makeSafe()`.

Frequently Asked Questions

What was the security flaw in Phoca Download?

Phoca Download's frontend member-upload feature skipped the file-type allow-list that the extension enforces everywhere else. Because of an internal upload-mode mismatch, the admin-configured list of permitted file types was never consulted for member uploads, so a logged-in user could upload a file type that should have been rejected, such as a .php script, into the public user-upload folder and then run it in a browser. That is authenticated remote code execution, classed as CWE-434, unrestricted upload of a file with a dangerous type.

Which version of Phoca Download is affected, and which one fixes it?

The flaw was present up to and including Phoca Download 6.1.2. Phoca fixed it in version 6.1.3, released on 10 July 2026. Update every Phoca Download install to 6.1.3 or later. When a release is brand new it takes a short while to reach the phoca.cz download portal and the in-Joomla update channel, so check you are actually installing 6.1.3 and not the earlier 6.1.2.

Is this as serious as the recent unauthenticated Joomla upload flaws?

It is rated Critical, but it is less exposed in practice. The Joomla CNA scored it CVSS 4.0 9.0 (Critical) because it ends in remote code execution, which compromises the whole server. What sets it apart from the recent RSFiles, Page Builder CK and Balbooa Forms flaws, all scored 10.0, is that those needed no account at all and were reachable by anyone on the internet in a single request. This one needs a registered account and needs the non-default Phoca Download user-upload feature switched on, with a category granting upload rights, so far fewer sites are reachable. Where those preconditions are met, though, any member gets code execution, so patch it promptly.

My site does not let members upload files. Am I affected?

If the Phoca Download user-upload feature is off, which is the default, the vulnerable path is not reachable and you are not exposed to this specific flaw. You should still update to 6.1.3 as a matter of hygiene, because a feature that is off today can be switched on later, and running the patched version removes the risk entirely.

How do I find every Phoca Download install across the sites I manage?

Manually you would log in to each Joomla site and check the installed extensions list. With mySites.guru you open the extension inventory, search for Phoca Download, and get every

connected site running it with its installed version on one screen. Anything on 6.1.2 or earlier needs the update, and you can push it to all of them from the same place.

How do I know if a site was already exploited through this?

Updating stops the next attempt but does not undo one that already happened. Look in the Phoca Download user-upload folder for any file that is not a genuine upload, especially anything ending in .php. Check your Joomla user list for administrator accounts you do not recognise, and look for recently modified or unfamiliar PHP files across the site. mySites.guru runs all three of these checks across every connected site at once.

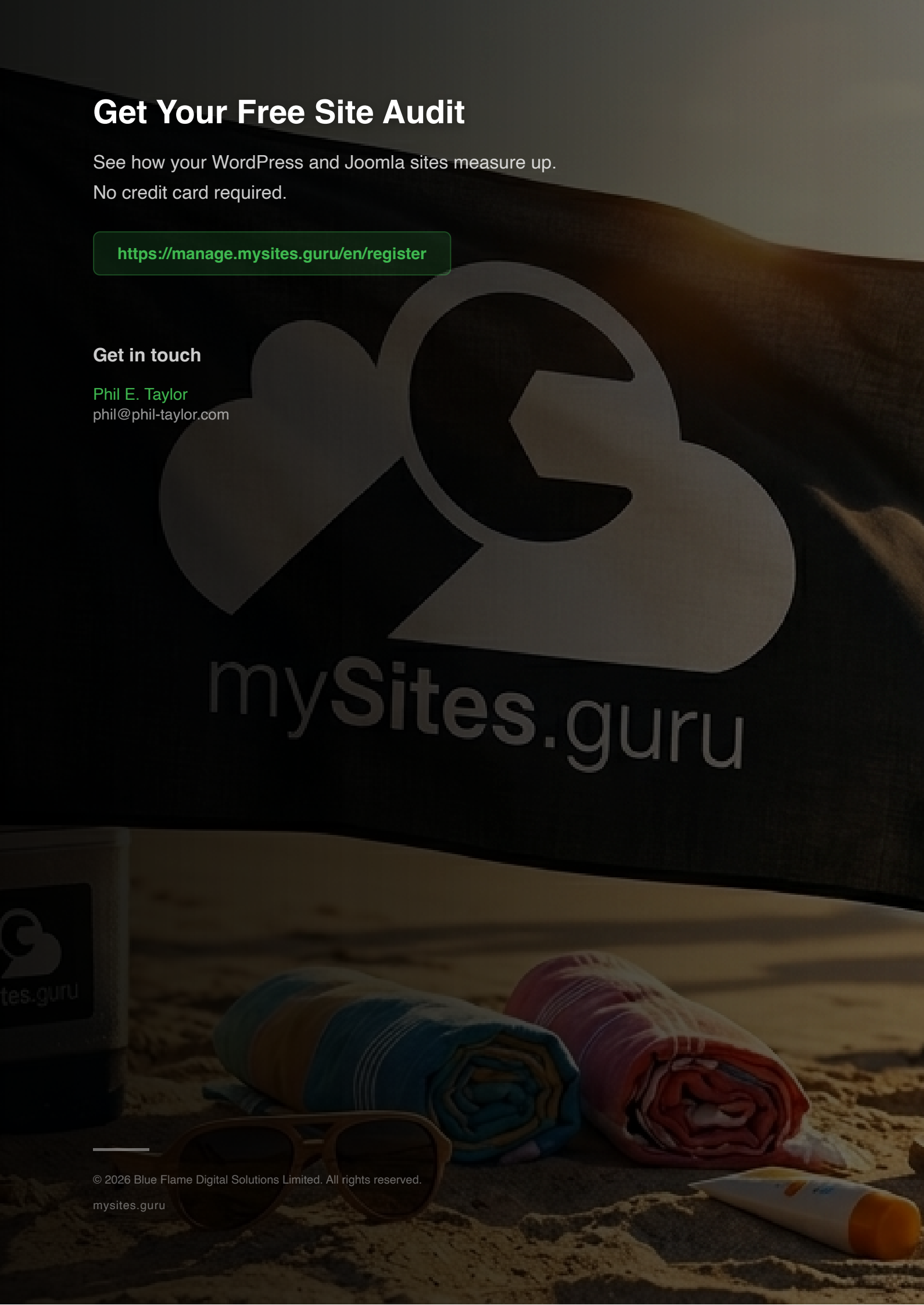
Written and edited by a human, with AI assistance. Our approach to AI

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.