mySites.guru

RSFiles! Fixes an Unauthenticated File Upload RCE

RSFiles! for Joomla (com_rsfiles) up to 1.17.11 had an unauthenticated file upload flaw that let anyone drop a PHP file and run code. Fixed in 1.17.12, update now.

Phil E. Taylor | 10 July 2026

RSFiles! is a widely used file-manager and download component for Joomla, installed as `com_rsfiles` and used to store, organise and serve downloads on business and membership sites. Up to and including version 1.17.11, its frontend upload had a serious flaw: the task that received an upload could be reached by anyone, with no login and no CSRF token, and the code that actually wrote the file to disk performed no check on its type. An attacker could upload a `.php` file into the public downloads folder and then run it, which is unauthenticated remote code execution, the worst outcome a web flaw can have.

RSJoomla fixed it in version 1.17.12, released on 10 July 2026. If you run Joomla sites with RSFiles! on them, update every one to 1.17.12 or later right now. This is not a wait-for-a-maintenance-window update. Any site still on 1.17.11 or earlier is reachable by anyone on the internet, so treat it as urgent and check it for tampering the moment it is patched. 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.

> **Unauthenticated and remotely exploitable, update immediately**
>
> This flaw needs no account on the site and ends in code running on your server. Update every RSFiles! install to 1.17.12 without waiting for a maintenance window, then check each patched site for stray files in the downloads folder and for unexpected admin accounts, because a site on an old version could already have been hit.

RSJoomla's Own Advisory

RSJoomla published a clear, unambiguous security advisory alongside the fix, which is exactly what you want a vendor to do. Their post, Unauthenticated File Upload fixed in RSFiles! version 1.17.12 - update NOW!, spells out the risk in plain terms rather than hiding it under a routine "Fixed" line.

FROM THE RSJOOMLA ADVISORY, POSTED BY OCTAVIAN CINCIU, 10 JULY 2026

Unauthenticated File Upload fixed in RSFiles! version 1.17.12 - update NOW!

A critical flaw in the RSFiles! upload function allows unauthenticated file uploads without enforcing any file extension.

What this means is that any attacker, without having an account on your website, can upload a .php file in your /downloads directory and execute it.

The bottom line is: update immediately to RSFiles! 1.17.12 which fixes this along some other not-reported, but less critical security issues.

TL;DR

TL;DR: The frontend upload in RSFiles! (`com_rsfiles`) up to and including 1.17.11 (fixed in 1.17.12) could be reached by anyone with no authentication and no CSRF token, and the method that actually wrote the file performed no permission check and no file-type check. Because it trusted the caller's filename and the default downloads folder sits inside the web root with PHP execution on, a `.php` upload landed in a public directory and could be executed, which is unauthenticated remote code execution (CWE-434). We found it during a source-code audit, reproduced the full chain on a local Joomla install, and disclosed it privately to RSJoomla, who responded the same day, shipped 1.17.12, and published an advisory telling site owners to update now. A CVE has been requested and is pending assignment, crediting Phil Taylor as the reporter. No proof of concept has been made public. Update every RSFiles! install to 1.17.12 or later immediately, then check exposed sites for stray files and unexpected admin accounts. If you manage more than a handful of Joomla sites, mySites.guru lists every RSFiles! install in your account in seconds and pushes the update.

How This One Was Found

This one did not start with an abuse report or a hacked site. 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. RSFiles! came up in that rotation. Reading through the upload path, the security checks were in the wrong place: the method that decides whether you are allowed to upload, and what file types are permitted, is a separate pre-flight step from the method that actually writes the file, and only the first one holds the checks. The second one, the one that writes to disk, could be called directly.

We reproduced the whole chain end to end on a local Joomla install: an anonymous request, no login at any step, uploaded a `.php` file into the downloads folder, and requesting that file back ran it. From there it was a private email to RSJoomla. They replied the same day, shipped the fix in RSFiles! 1.17.12, and published an advisory the same day telling site owners to update immediately. A CVE has been requested and is pending assignment, 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 vendor the same day, with a clear public advisory and no technical detail released. No exploit has been published. The reason to write about it is to get sites updated, not to show anyone how it worked.

What Was Actually Wrong in RSFiles!?

The flaw was an unauthenticated file upload where the security checks and the file write were in two different places, and only one of them was guarded. RSFiles! splits its upload across two separate frontend tasks:

- A pre-flight check that holds the permission gate (can this user upload at all) and the extension allow-list (the permitted file types, by default images, text and PDFs). This method decides yes or no. It writes nothing.
- The write method that actually receives the file and saves it to disk. This method performed no permission check and no extension check. It read the target

filename straight from the request and handed the upload to Joomla's bundled upload handler.

That bundled handler accepts any file type unless the calling code tells it otherwise, and RSFiles! never told it otherwise, so its default was to accept everything, including `.php`. Because the write task could be called directly, an attacker skipped the pre-flight check entirely and posted to the write task, which never consulted the allow-list or the permission gate. The frontend controller had no site-wide CSRF token and no access check, so the whole thing ran for an anonymous visitor.

The final piece is where the file lands. RSFiles!'s default downloads folder sits inside the web root, and the protective `.htaccess` that would stop PHP running there is an opt-in admin setting that is off by default. So an anonymous upload of a `.php` file went into a directory that serves and executes PHP, and requesting it in a browser ran it. No account on the site was needed at any point.

The important detail for site owners is the "no login" part. Plenty of upload bugs need an editor or admin account first, which limits the blast radius to people you already trust. This one did not. Anyone on the internet who could reach the site could reach the upload.

We are deliberately not publishing the exact request an attacker would send. The shape of the bug is enough to understand the risk and to check your own sites; the working details stay private.

Why an Upload Flaw Is as Bad as It Gets

An unauthenticated upload that lands executable code is the top of the severity scale, and it is worth being clear about why. Once an attacker can run PHP of their choosing on your server, they are no longer limited to your site. They can read your `configuration.php` and the database credentials in it, create a hidden Joomla Super User, plant a backdoor that survives an extension update, pivot to other sites in the same hosting account, or add the server to a spam or malware network. The uploaded file is rarely the goal in itself; it is the foothold.

This is the same lesson as the AJAX and frontend endpoints blind spot we keep coming back to. An endpoint that Joomla will run for anonymous visitors is exposed to the entire internet, so every input it touches has to be treated as hostile. A single allow-list check on the file that is actually written, "is this one of the types this folder is allowed to hold", would have stopped this outright.

What the RSFiles! 1.17.12 Fix Does

RSJoomla's fix closes the anonymous route into the upload. Version 1.17.12 requires a valid CSRF token on the upload task and runs the permission and allow-list checks before the file is written, so a visitor with no account is refused before anything reaches the disk. The changelog records the same release also restoring the missing CSRF token on uploads and blocking file-preview access when permissions are disabled. Their advisory notes it fixes some other, less critical issues that were not separately reported.

The practical takeaway is simple: 1.17.12 shuts the door that let an anonymous visitor upload and run code. Every RSFiles! install on 1.17.11 or earlier is on the wrong side of that door until it is updated. Because the flawed routine had been in the code for years, "my site is old and stable" is a reason to update sooner, not later.

How mySites.guru Caught This Without a Signature

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

That generic detection is the same logic that catches Page Builder CK, iCagenda, Balbooa Forms and JCE upload attempts. An attacker dropping a `.php` file through a public endpoint 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.

Signature-based tools only see an attack once someone has described it. Behaviour-based detection sees it the first time it happens.

If a hostile file does land, the suspect content 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 RSFiles! 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 RSFiles! 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 RSFiles! install across your sites

Open your Extension Inventory

Search for RSFiles! across every connected Joomla site and filter for anything on 1.17.11 or earlier to find the installs that still need updating. Not a subscriber? Sign up free and connect your sites.

Bulk Updating RSFiles! Across Every Site

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. The same routine covers any Joomla extension, plugin, or core update, so the workflow you set up once works for the next security release too. You can also switch on automatic updates for any Joomla extension so future RSFiles! releases land without you lifting a finger.

For agencies managing dozens of Joomla sites

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

If a particular site cannot be updated right away, unpublishing the RSFiles! frontend and any menu items that expose its upload takes the vulnerable handler out of reach while you schedule a maintenance window. A web application firewall such as RSFirewall! or Admin Tools can also blunt the attack, but neither is a substitute for the update.

How Do I Check an RSFiles! Site for Tampering?

Updating closes the door. It does not tell you whether anyone walked through it first. Because the flaw allowed an anonymous visitor to drop a file and run it, a site that was on 1.17.11 or earlier could already have been touched, and you should check before you assume it is clean.

Three checks, in order of value:

1. Look in the downloads folder. RSFiles! stores files under a downloads directory inside the web root. Anything there that is not a genuine download, 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 Extensions Are a Recurring Joomla Attack Surface

If you manage Joomla sites, treat every extension that accepts or serves files as a piece of your attack surface that deserves regular attention. The exact flaw fixed here, an unauthenticated upload of a file with a dangerous type, 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. Page Builder CK, iCagenda, Balbooa Forms, and the SP Page Builder zero-day were all the same shape: a frontend endpoint that took a file from an anonymous visitor without a strict check on what was actually written. When a category of extension keeps producing the same kind of bug, the sensible response is not to distrust one vendor, it is to monitor the whole category and to patch fast when a fix lands.

It is worth keeping this in proportion for RSFiles! specifically. Its only previously recorded CVE is a directory traversal from 2007 (CVE-2007-4504) in the 1.0.2 release, nearly two decades ago. This upload flaw is a different and far more serious class of bug, and the response to it, a same-day fix and a clear advisory, is the more relevant signal about how the extension is maintained today.

Credit where it is due: RSJoomla did the two things that matter. They fixed it the same day it was reported, and they published a plain-spoken advisory that tells site owners to update now rather than burying it under a routine changelog line. That is not always how these go. We have written before about a vendor shipping a security fix under a bland "Security Update" changelog with nothing to tell owners how urgent it was. RSJoomla did the opposite, and it is the right call.

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 RSFiles! flaw lands, you see exactly which sites are exposed 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.

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, reached by an anonymous visitor over the network in a single request, with no privileges and no user interaction, and it ends in full remote code execution. That profile is the worst a web flaw can have.

10.0
CVSS 4.0

CRITICAL **The maximum possible score**

It is unauthenticated, remotely exploitable in a single request, needs no user interaction, and ends in full remote code execution, so every metric that makes a flaw dangerous is at its worst here. This is the same profile as the near-identical Page Builder CK and Balbooa Forms flaws, both of which scored 10.0 on CVSS 4.0.

No login needed | Exploitable over the internet | No user interaction | Full remote code execution

A CVE has been requested for this flaw and is pending assignment, crediting Phil Taylor of mySites.guru as the reporter. We will add the identifier here once it is issued.

Field	Detail
CVE	Requested, pending assignment
Component	RSFiles! (`com_rsfiles`)
Vendor	RSJoomla
Type	Unauthenticated arbitrary file upload to remote code execution
CVSS 4.0	10.0 (critical), `AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H`
CWE	CWE-434 (Unrestricted Upload of File with Dangerous Type)
Finder	Phil Taylor, mySites.guru
Affected versions	Up to and including 1.17.11
Fixed in	1.17.12, released 10 July 2026

The disclosure ran on a single-day cycle from audit to fix:

Date	Event
9 July 2026	During a source-code audit of commonly installed Joomla extensions, we find the unauthenticated upload in RSFiles! and reproduce the full chain end to end on a local

Date	Event
	Joomla install: an anonymous request uploads a `.php` file into the downloads folder and executes it.
10 July 2026	We disclose the flaw privately to RSJoomla. The vendor responds the same day, ships RSFiles! 1.17.12 closing the anonymous upload route, and publishes an advisory telling site owners to update now. A CVE is requested. No proof of concept is published.

Further Reading

- RSJoomla advisory: Unauthenticated File Upload fixed in RSFiles! 1.17.12 - the vendor's own notice, telling site owners to update now.
- 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 RSFiles!?

The frontend upload task in the RSFiles! component (com_rsfiles) could be reached by anyone, with no login and no CSRF token, and the code that actually wrote the file performed no check on the file type. Because the extension trusted the filename the visitor supplied, an anonymous attacker could upload a .php file into the public downloads folder and then request it in a browser to run their own code on the server. That is unauthenticated remote code execution, the most serious outcome a web flaw can have. It is classed as CWE-434, unrestricted upload of a file with a dangerous type.

Which version of RSFiles! is affected, and which one fixes it?

The flaw was present in RSFiles! up to and including 1.17.11, and the flawed upload routine had been in place for years. RSJoomla fixed it in RSFiles! 1.17.12, released on 10 July 2026. Update every RSFiles! install to 1.17.12 or later. If you are on 1.17.11 or earlier, treat the site as exposed until it is updated, and check it for tampering because an unauthenticated upload flaw leaves no login trail behind.

Is RSFiles! 1.17.12 an urgent update or can it wait?

It is urgent. The flaw needs no account on the site, works in a single request, and ends in code running on your server, so any RSFiles! install still on 1.17.11 or earlier is reachable by anyone on the internet. RSJoomla's own advisory tells site owners to update NOW. Do not wait for a maintenance window: update, then check the downloads folder and your admin accounts for anything that should not be there.

How do I find every RSFiles! 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 RSFiles!, and get every connected site running it with its installed version on one screen. Anything on 1.17.11 or earlier needs the update, and you can push it to all of them from the same place.

My site runs RSFiles!. How do I know if it was already hit?

Updating stops the next attempt but does not undo one that already happened. Look in the RSFiles! downloads folder (by default a /downloads directory under the web root) for any file that is not a genuine download, 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 automates all three checks across every connected site.

Do file-manager and upload extensions get attacked often?

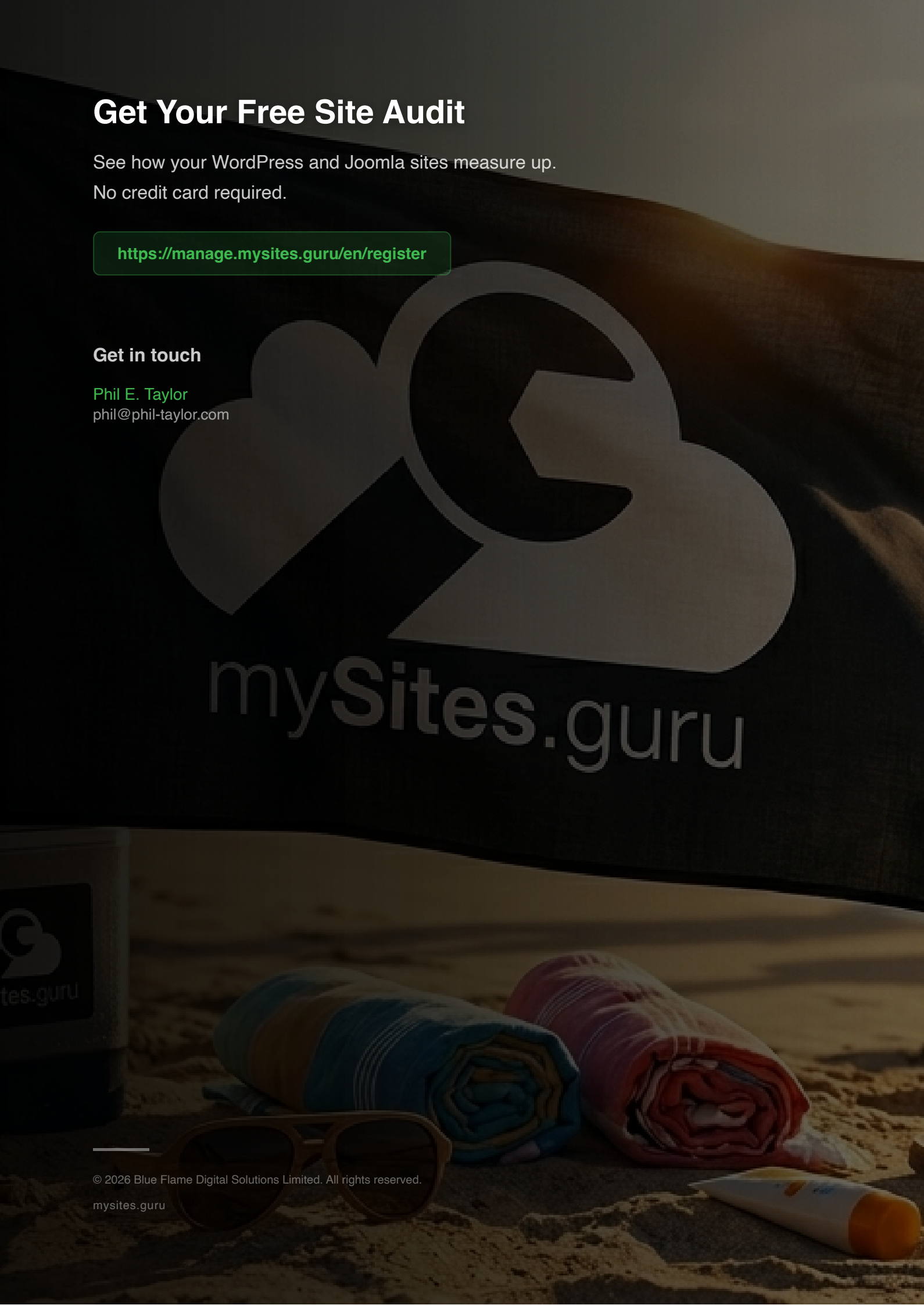
Yes. Any extension whose job is to accept or serve files is a natural target, because an upload that lands executable code is the fastest route from the public internet to a shell on the server. The same class of flaw, an unauthenticated file upload, was fixed recently in Balbooa Forms, iCagenda, Page Builder CK and JCE. Any extension that takes uploads needs a strict allow-list checked against the file that is actually written to disk, not the one the caller claims to be sending.

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