mySites.guru

We Are Not the Only Ones Auditing Joomla Extensions

Two Joomla extension flaws went public via the Joomla CNA: a SQL injection in JoomCCK and a stored XSS in ChronoForms. Neither was ours. Update now.

Phil E. Taylor | 28 August 2026

mySites.guru has reported a run of Joomla extension vulnerabilities over the past month, enough of them that you could be forgiven for thinking we are the only people auditing this code. We are not, and this week proved it cleanly. Two more Joomla extension vulnerabilities were published by the Joomla CNA, both rated High, and neither of them was ours.

We are writing about them anyway, for two reasons. Our customers run both extensions, so the update matters to them whoever found the flaw. And it is worth saying out loud that a healthy security ecosystem has more than one set of eyes in it.

Two more extension flaws, and neither was ours

Here are the two, exactly as the Joomla CNA published them:

CVE	Extension	Flaw	Fixed in	Reported by
CVE-2026-49048	JoomCCK (JoomCoder)	Unauthenticated SQL injection	6.4.1	Kamil Soltanov
CVE-2026-58148	ChronoForms (ChronoEngine)	Unauthenticated stored XSS	8.0.53	Italo Almeida

CVE-2026-49048: SQL Injection in JoomCCK

CVE-2026-49048 is an unauthenticated SQL injection in JoomCCK, the content construction kit by JoomCoder. A front-end controller task builds two SQL statements by concatenating a request parameter straight into the query, with no escaping or parameterisation, so an anonymous visitor can bend the query and read from the database. Every version up to and including 6.4.0 is affected, and it is fixed in 6.4.1. The Joomla CNA scored it CVSS 4.0 8.7 (High), and it rates 9.8 (Critical) on the older CVSS 3.1 scale. It was reported by Kamil Soltanov.

8.7
CVSS 4.0

HIGH **CVE-2026-49048 (JoomCCK), scored by the Joomla CNA**

Unauthenticated SQL injection reachable over the internet with no login, ending in read access to the site database. It also rates 9.8 (Critical) on the older CVSS 3.1 scale. Fixed in JoomCCK 6.4.1.

No login needed | Exploitable over the internet | Database read | 9.8 Critical on CVSS 3.1

CVE-2026-58148: Stored XSS in ChronoForms

CVE-2026-58148 is an unauthenticated stored cross-site scripting flaw in ChronoForms, the long-established form builder by ChronoEngine. An anonymous visitor can store markup that later executes in another user's browser. Every version up to and including 8.0.52 is affected, and it is fixed in 8.0.53. It too was scored CVSS 4.0 8.7 (High), and was reported by Italo Almeida.

8.7
CVSS 4.0

HIGH

CVE-2026-58148 (ChronoForms), scored by the Joomla CNA

Unauthenticated stored cross-site scripting reachable with no login, where injected markup later executes in another user's browser. Fixed in ChronoForms 8.0.53.

No login needed | Unauthenticated stored XSS | Runs in a visitor's browser

Credit where it is due. Kamil Soltanov found the JoomCCK injection and Italo Almeida the ChronoForms XSS, both credited by name in the public CVE records. Neither is connected to us. That is the whole point of this post: independent researchers are auditing the same extensions we are, and that is a good thing.

It has kept happening since. In August 2026 the Joomla CNA published an unauthenticated remote code execution flaw in Fabrik, CVE-2026-66915, scored a maximum CVSS 10.0 and reported by Moe Khalilov of LeetProtect Research. Also not ours.

Why a busy month of disclosures is good news

It is easy to read a run of extension CVEs as a sign that Joomla itself is falling apart, and it is worth being precise that these are flaws in third-party extensions, not in Joomla's own core. The truth is closer to the opposite of falling apart: flaws that are found and disclosed get fixed. The ones that should worry you are the ones nobody is looking for, sitting in code that never gets a second pass. More researchers looking at widely used extensions means more of those quiet flaws get pulled into the light and patched.

The catch is the part nobody puts on the CVE record. Every one of these fixes is one more update your sites need, and the fix only protects the sites that actually take it. A disclosure is only half the job. The rollout is the half that decides whether any of it protected anyone, and the rollout is exactly the part that gets skipped when checking every site by hand is a chore that grows with every client you take on.

But doesn't publishing just tip off hackers?

Every time we post one of these, someone on Facebook tells us we should keep quiet, that writing it up only hands attackers a map, or that we are doing great harm to the Joomla community by making the flaw public at all. It is a fair thing to raise, and it deserves a real answer rather than a shrug.

A blog post like this one is the end of the process, not the start of it. Behind it sits weeks, sometimes months, of work nobody sees: finding the flaw, proving it on a test install, writing it up for the developer, disclosing privately, waiting, testing the fix they send back on their own behalf, coordinating the release, waiting for a CVE to be assigned, and waiting again. By the time we publish, the fix has already shipped and the CVE record already exists. We are the last car in that convoy, not the first.

> **"Malicious actors do not read blog posts to work out what to exploit. They are already on it."**

That is the part the objection misses. Sophisticated threat actors, some of them state-backed, are already on these extensions, continuously and constantly, probing widely used code for exactly this kind of flaw, often before anything is public at all. Keeping a vulnerability secret does not stop them finding it, it only stops the defenders knowing. That is how zero-days come to exist in the first place. Where a fix already exists, they are watching the signals that ship long before any write-up. The moment a vendor ships a release labelled “security”, someone diffs it against the previous version and finds the changed line in minutes.

This is not unique to extensions, and it is worth spelling out because it is exactly how Joomla’s own core security releases work. The moment the Joomla project ships a core security update, malicious actors are already poring over the diff, isolating the changed lines, and building tools to exploit every site that has not updated yet. By releasing the fixed code at all, Joomla has itself invited that critique, because a security patch is a map straight to the bug it closes. Nobody sensible argues the project should therefore stop shipping fixes. The release has to be public to protect anyone, and the release is what starts the clock, not a blog post written after it.

The moment a CVE record is created, it is indexed and scraped. Exploitation starts at the release and the CVE, not at the write-up, and it would start there whether or not we ever wrote a word. Some vendors we have given a ninety-day window have simply shipped a patched version without telling us, and declined to coordinate at all. The fix was already public, the diff was already there to read, and our silence would have protected nobody.

The CVE record is not only fuel for attackers, it is the reason defenders stand a chance of keeping up. The CVE program exists to standardise how each vulnerability is named and tracked, so a security team can identify the weakness, scan their own estate for it, and prioritise the fix from one shared reference rather than a hundred vendor blogs in a hundred different formats. A flaw fixed with no error, with no CVE, still helps the attacker who reverse-engineers the patch, and helps nobody else. A published CVE is what lets everyone else catch up. It is the record mySites.guru consumes to tell you which of your sites are affected, which is the whole point of assigning one.

A silent argument never wins. Keeping quiet does not slow an attacker who is already working from public information, it only leaves the people running the software in the dark. The scale settles it anyway. Wordfence alone published more than two hundred WordPress plugin and theme vulnerabilities in the past week, and more than eight hundred this month. We have published fewer than fifteen, a vanishingly small share of what is already out in the open, and the idea that our handful of posts is what moves the needle does not hold up next to those numbers. What our fifteen do have is some of the most critical flaws we have ever found, which is exactly why the people running that code deserve to hear about them.

Put the choice the way it actually presents itself. Search Google for “hacked by AntonKill” and you are shown a sea of defaced Joomla sites whose owners never applied the update that would have saved them. Would you rather read a blog post we wrote about the flaw and how to close it, or be one more entry in that list of a million sites that never bothered? A write-up is the version of the story where the reader still gets to act first.

Then there is Bob the Builder

The “just stay quiet” argument never answers the obvious question: if not a blog post, then how is anyone meant to find out at all?

Open source has no mailing list. There is no register of everyone running a given Joomla extension, and no button a developer can press to reach them. Commercial extensions are barely better off. Most vendors do not hold an accurate list of who actually runs their product, and the lists they do hold are full of addresses that unsubscribed long ago and never see the alert.

So picture Bob. Bob is a builder, on site all day, up a ladder. His website was put together one weekend by his teenage son, who was bored and dodging his homework. Bob is never going to read a security advisory. He will not check a changelog, and he will not find his way to a blog post about an SQL injection in an extension he could not name if you asked him. Telling Bob to follow the developer’s announcements is telling him nothing. He does not even know which extensions went into his site in the first

place, and he has never had a reason to. What Bob can do is pay someone to watch on his behalf, which is the whole reason a service like mySites.guru exists.

At the other end sits the agency with a thousand client sites, for whom “just update it” is the same small job multiplied by a thousand logins. They do not need telling that a flaw exists so much as they need one place to see which of their sites are affected and fix them together. It is the same need Bob has, at a different scale.

What almost 100,000 live sites tell us

It is easy to forget, in an argument about whether a blog post helps or harms, what sits behind these posts. mySites.guru is not a side project. It has thousands of paying subscribers and watches close to 100,000 live websites every single day. That scale is the whole point: our view of Joomla extension security comes from the real world at volume, not from a lab or a single test install on a laptop.

We see the trends in that data before the Joomla project itself hears about them. We can watch which extensions are suddenly being probed, which exploits are being tried, and which sites have been compromised, because we are looking at real production websites rather than a hypothesis. The sites we watch are every kind there is:

- the village bakery
- the local butcher
- the school-project site a kid built five years ago and left running on a Raspberry Pi, long since compromised and now hosting a crypto-miner and a backdoor for someone else
- and, at the other end, the sites behind government bodies in countries all around the world, hundreds of them

That range is exactly why this work matters. A flaw in a Joomla extension is never an abstract CWE number to us. It is a specific list of real sites, run by real people who mostly have no idea the extension they installed once has a hole in it. When we tell you

to update JoomCCK or ChronoForms, it is because we can see who is running them, and what tends to happen to the sites that do not.

How to make sure you actually get the fixes

This is the job mySites.guru is built for, and it does not care who found the flaw. We track published Joomla extension CVEs, add the affected version ranges to our vulnerability database, and flag every connected site running a vulnerable version. Whether the finder was us, Kamil Soltanov, Italo Almeida, or anyone else, the outcome for you is the same: the affected sites light up, and you update them.

> **"In our experience there are only two kinds of site owner: the ones who patch the day a fix ships, and the ones who never patch at all."**

We cannot make anyone patch. What we can do is take away every excuse not to. mySites.guru puts the affected sites, the available update, and the button that applies it in one console, so “I did not know” and “keeping up across that many sites is too much work” stop being reasons.

Find every JoomCCK and ChronoForms install across your sites

Open your Extension Inventory

Search once and see every connected Joomla site running either extension, the version each is on, and whether an update is waiting. Not a subscriber? Sign up free and connect your sites.

So the short version is the same as it always is:

- Update JoomCCK to **6.4.1** or later

- Update ChronoForms to **8.0.53** or later
- If you manage more than a handful of sites, let mySites.guru show you which ones run either extension rather than checking each admin by hand, then **push the update to all of them from one dashboard**

We will keep auditing Joomla extensions and reporting what we find. It is reassuring to be reminded, in the same week, that we are in good company doing it.

Further Reading

- A month of Joomla security disclosures - the run of extension vulnerabilities we found and reported over the same period.
- Why AJAX endpoints are a CMS security blind spot - the front-end endpoint pattern behind the JoomCCK injection and a long line of similar flaws.
- CVE-2026-49048 and CVE-2026-58148 - the public CVE records for the two flaws.
- The Joomla CNA - the body that assigns CVEs for Joomla and its extensions.

Frequently Asked Questions

Did mySites.guru find these two Joomla extension vulnerabilities?

No. CVE-2026-49048 in JoomCCK was reported by Kamil Soltanov, and CVE-2026-58148 in ChronoForms by Italo Almeida. Both researchers are credited in the public CVE records assigned by the Joomla CNA. We are writing about them because our customers run both extensions, not because we found them.

Which versions do I need to update to?

Update JoomCCK to 6.4.1 or later and ChronoForms to 8.0.53 or later. Both flaws are unauthenticated and were scored CVSS 4.0 8.7 (High) by the Joomla CNA, and the JoomCCK SQL injection rates 9.8 (Critical) on the older CVSS 3.1 scale, so treat them as priority updates rather than routine ones.

How does mySites.guru know about vulnerabilities it did not discover?

We track published Joomla extension CVEs from the Joomla CNA and other sources, add the affected version ranges to our vulnerability database, and flag every connected site running a vulnerable version. That happens regardless of who reported the flaw, so your sites get flagged whether the finder was us, one of these researchers, or anyone else.

Doesn't publishing these vulnerabilities just help attackers?

No. By the time we publish, the fix has already shipped and the CVE record already exists, and attackers watch vendor releases and CVE records, not our blog. Exploitation of a flaw like this starts when a fix is released or a CVE is assigned, because anyone can diff the release or scrape the record, not when a write-up appears days later. Staying silent would only keep the people running the extension in the dark while attackers act on information that is already public.

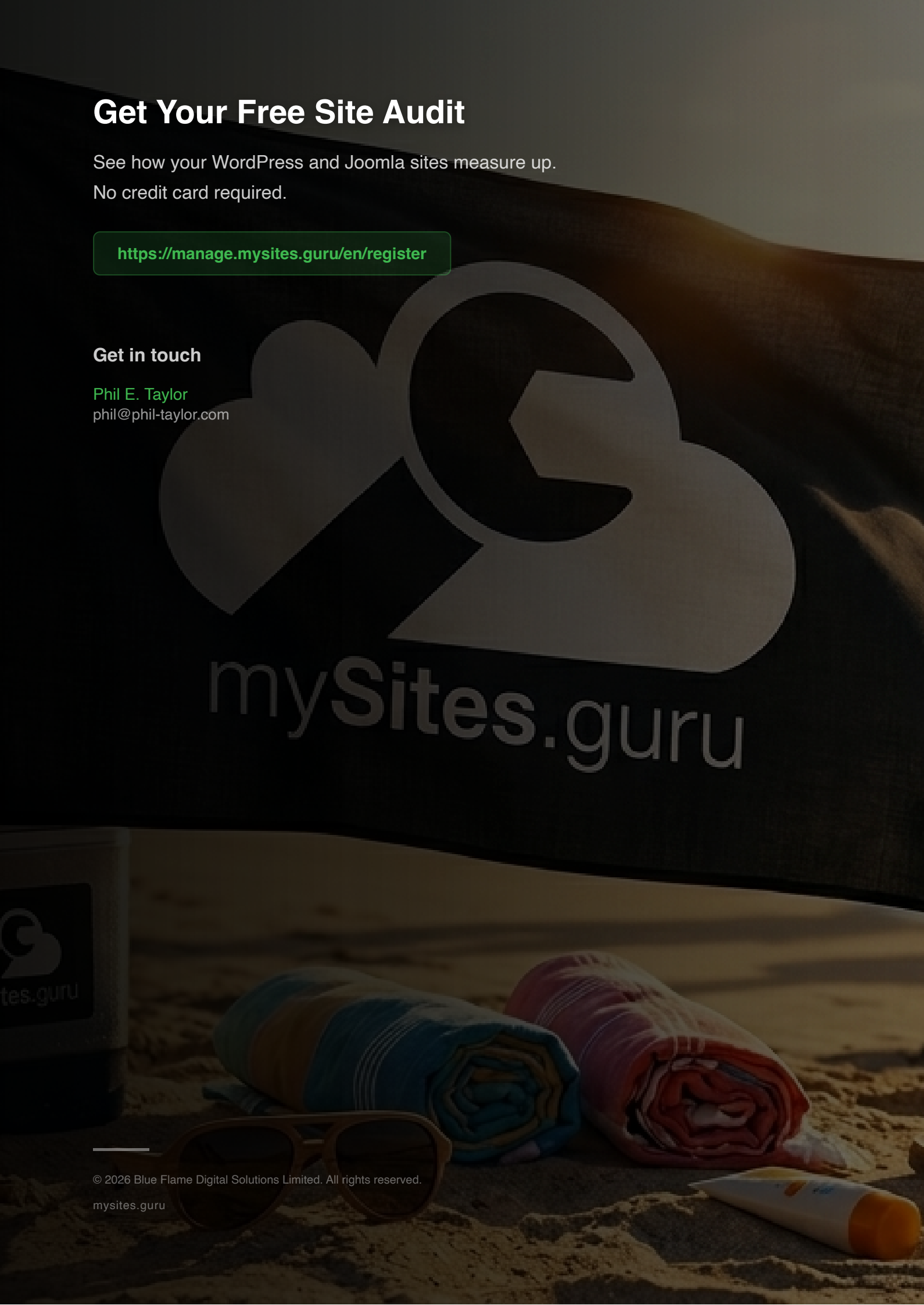
Written and edited by a human, with AI assistance. Our approach to AI

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.