Exposed Customer Invoices, Order Forgery and SQL Injection in EasyStore for Joomla found by mySites.guru

Before EasyStore 2.0.2, any logged-in customer could read every other customer's invoice by editing one URL. mySites.guru found this and two unauthenticated flaws. Update now.

Phil E. Taylor | 23 July 2026

Active Joomla Extension security alerts: Sixteen and counting this month • JCE • Quix • SP Page Builder

EasyStore is JoomShaper's ecommerce component for Joomla, the shopping-cart engine behind a growing number of Joomla shops. Here is the finding that will make any shop owner wince: before EasyStore **2.0.2**, any logged-in customer could pull up every other customer's order and printable invoice, with full name, email address, billing and shipping addresses, phone number and complete purchase history, **by changing one number in the URL**. On a shop with open customer registration, which is most shops, "any logged-in customer" means anyone who signs up. mySites.guru found that during routine security research, along with two more issues that need no login at all, and reported all three to JoomShaper before disclosing anything publicly.

If you run EasyStore on any Joomla site, update to 2.0.2 now. If you manage more than a handful of shops, read on for how to find every affected one at once, and for how JoomShaper chose to ship the fix.

TL;DR

- mySites.guru found **three critical vulnerabilities** in EasyStore 2.0.1 and reported them to JoomShaper
- **Cross-customer invoice and order exposure.** The order and invoice views only checked that *someone* was logged in, never that the order belonged to them, so **any customer could read every other customer's order and invoice** by changing the id in the URL. Names, emails, addresses, phone numbers, the lot
- **Unauthenticated order forgery.** An anonymous request to the order-repayment endpoint could mark any order **paid** with no login, no token, and no contact with any payment gateway. Goods released without payment, and the same flaw rewrites any order's fields by changing an id
- **Unauthenticated SQL injection.** The product-list sort parameter placed request input straight into the database query's `ORDER BY`, so an **anonymous visitor could**

read the entire database: Joomla accounts, password hashes, the site secret, and every customer record

- **Update to EasyStore 2.0.2 immediately.** Affected: 2.0.1 and every earlier version
- Reported privately on 21 July 2026, fixed the next day in **2.0.2**, but shipped with **no security advisory and no credit**. mySites.guru has applied for a CVE through the Joomla CNA
- mySites.guru already flags every connected Joomla site still running a vulnerable version, so you do not have to check each shop by hand

> mySites.guru discovered these issues and reported them to JoomShaper before publishing details. We withheld the exact requests 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.

How JoomShaper shipped the fix: quietly

We reported all three privately on 21 July 2026. JoomShaper replied on 22 July, confirmed they had reproduced and verified every one, said they were “treating these issues with the highest priority”, thanked us, and said they looked forward to continuing to work with us. The next day, 23 July, they shipped EasyStore **2.0.2**, and it closes all three.

To be fair to the code, we checked. We downloaded 2.0.2 and read the exact paths we had reported: the repayment endpoint now requires a token, checks that the order belongs to the caller, and takes the payment status from the stored order rather than from the request; the product sort is restricted to a fixed `ASC` / `DESC` allow-list; and the order and invoice views now verify that the order belongs to the person viewing it. The fix is correct and complete, and we have no complaint about the code.

The handling is another matter. The 2.0.2 release carries no security advisory, the changelog never says “security release”, and the three fixes sit among routine feature

additions and cosmetic bug fixes, each written in language a shop owner would read straight past. Here is the changelog as JoomShaper published it:

23 July 2026

Updates

- Added customization options for product specification display. (Pro only)
- Added tracking URL support in the order detail summary header for admin and site.
- Added custom class support with default classes for all addons in settings. (Pro only)
- Improved range slider validation and value mapping.

Fixes

- Updated rating descriptions and added language strings.
- Resolved coupon condition issues.
- Resolved addon settings class undefined issue on product list. (Pro only)
- Product gallery thumbnail and main image now update correctly on variant selection.
- Fixed unauthenticated order state manipulation in the checkout repay task.
- Added strict allowlist sanitization for product list sorting options.
- Enforced user ownership verification on order details and invoice views.

The three critical security fixes are the last three lines of the Fixes list, sitting directly under a coupon bug and a product gallery thumbnail glitch, with nothing to mark them out as different. Here is what we reported, set against the exact words JoomShaper used to describe each fix:

What we found and reported	How JoomShaper described it in the 2.0.2 changelog	Severity (CVSS 4.0)
Any logged-in customer could read every other customer's order and invoice by	*"Enforced user ownership verification on order details and*	**7.1, High**

What we found and reported	How JoomShaper described it in the 2.0.2 changelog	Severity (CVSS 4.0)
changing the id in the URL	*invoice views"*	
An anonymous request could mark any order paid, and rewrite any order's fields, with no login	*"Fixed unauthenticated order state manipulation in the checkout repay task"*	**8.7, High**
An anonymous visitor could read the entire database through the product sort parameter	*"Added strict allowlist sanitization for product list sorting options"*	**8.7, High**

Only one of those three lines even contains the word "unauthenticated". There was no coordinated disclosure date, no heads-up that the release was going out, no CVE requested, and no credit to the finder. A shop owner scanning that changelog has no way to know that 2.0.2 is the most urgent update EasyStore has shipped this year, or that staying on 2.0.1 leaves every customer's invoice one URL edit away from any logged-in visitor.

Because JoomShaper did not request one, mySites.guru has applied for a CVE for these issues through the Joomla CNA, crediting the finder, so there is a public, citable record that does not depend on a changelog line reading like a slider tweak.

> Treat EasyStore 2.0.2 as an urgent security release, whatever its changelog implies. Two of the three flaws need no login, and the third exposes your entire customer base's personal data to anyone who can register. If a vulnerable version was live on a shop that handles real customers, this is a personal-data breach you may be obliged to report.

The three flaws in detail

EasyStore is a full ecommerce component: products, a cart, checkout, orders, invoices, coupons and payment plugins. That is a large front-end attack surface, because a shop

has to accept requests from anonymous visitors to function. Most of it is built carefully. Three places were not.

1. Every customer's invoice, one URL edit away

The flaw that leads this post is a broken access control, the kind that quietly exposes personal data. EasyStore's order detail page and its printable invoice only checked that *someone* was logged in. They never checked that the order being viewed actually belonged to the person viewing it. The query that loads the order (`OrderManager.php:917–938`) filtered on the order id alone, with no customer-ownership condition.

Because order ids are sequential integers, any logged-in customer could walk through them one by one and read every other customer's order and printable invoice: full name, email address, billing and shipping addresses, phone number, and complete purchase history. Change `id=1041` to `id=1042` and you are looking at the next customer's invoice. On a shop with open customer registration, "any logged-in customer" means anyone. That is a serious personal-data exposure and, under the UK GDPR and equivalents, a reportable breach if it happened on a live shop.

The tell here is that EasyStore already gets this right elsewhere. Guest orders are protected by an unguessable token emailed to the customer, and the order-cancel path does check ownership. The detail and invoice views simply missed the same check. The fix (which we confirmed in 2.0.2) adds the ownership condition so a customer can only ever load their own orders.

> Data exposure like this leaves no trace. No file changes, no new admin user, nothing for an integrity scanner to catch. If a vulnerable version was ever live on a shop with real customers, you have to assume the customer list and its invoices were reachable, and handle it as a potential data breach.

2. Marking any order paid, for free, with no login

The most direct of the three, and it needs no account. EasyStore has an order-repayment path, meant to let a customer retry payment on an order. The endpoint took the order details straight from the request body as JSON, including the **payment status**, and for any enabled manual payment method (Cash on Delivery is enabled by default and qualifies) it wrote that status to the database.

The root cause was in the checkout controller, which built the order object from client JSON and called `updateOrder()`, ending in a blind write at `OrderController.php:299` (`$orm->update('#__easystore_orders', $data, 'id')`). Nothing in that path checked a login, a CSRF token, ownership of the order, or, crucially, whether any payment actually happened. There was no call to a payment gateway anywhere in it.

So an anonymous attacker could send one crafted request and flip any order to `paid`. On a shop that ships on payment status, that is goods dispatched for free. Because the order id is attacker-chosen and sequential, the same flaw also let an attacker rewrite arbitrary fields on **any** order in the shop, addresses, customer id and discount included.

We proved it on our own EasyStore test install: a single anonymous request turned a test order from `unpaid` to `paid`, and we then restored the test data. The fix (confirmed in 2.0.2) ties the repayment path to the real customer, requires a token, and only ever sets payment status from the stored order or a verified gateway response.

3. Unauthenticated SQL injection through the product sort

The product listing lets shoppers sort results, and the sort option arrives in a request parameter, `filter_sortby`. EasyStore split that value into a column and a direction, and the direction was passed through to the database query's `ORDER BY` clause **without being checked against a list of allowed values**.

The direction helper (`FilterHelper.php:741`) returned the request's direction unchanged, and the model (`ProductsModel.php:923`) concatenated it straight into the query. A sort direction is one of the few places you cannot simply quote a value safely, which is exactly why the surrounding code has to restrict it to `ASC` or `DESC`. This path did not, while the sibling brand and collection listings did. That gap was the bug.

The consequence is a textbook SQL injection reachable by an anonymous visitor: an attacker could read any table in the Joomla database, from user accounts and password hashes to the site secret and every customer's stored order data. We confirmed it on our test install with a time-based proof, a crafted sort value that made the database pause for a measurable few seconds versus an instant baseline, which shows the injected code ran. We did not extract real data. The fix (confirmed in 2.0.2) restricts the direction to a fixed allow-list before it reaches the query.

> A database read writes nothing. No file changes, no new admin user, no web shell, nothing for an integrity scanner to find. If a vulnerable version was ever live on a busy shop, treat the database as potentially copied: rotate the Joomla secret and any stored API keys, and assume password hashes are known.

Fair credit: the PayPal side was built correctly

It would be easy to read three findings and conclude the whole component is careless. It is not. The part that most often goes wrong in ecommerce extensions, the payment gateway callback, is built properly here. EasyStore's PayPal webhook verifies the notification's cryptographic signature, checks the merchant email, and reads the payment status back from PayPal's own signed response rather than trusting request parameters. The order total is computed server-side from the database, not taken from the client. That is the right way to do it, and it is worth saying so.

The order-forgery flaw was not a weakness in PayPal handling. It was a separate, manual-payment code path that skipped the checks the gateway path applied. The lesson is a familiar one: security has to hold on **every** path that changes state, not just the obvious one.

Which versions are affected?

EasyStore **2.0.1**, the current release at the time of disclosure, and **every earlier version** are affected by all three issues. JoomShaper fixed them in EasyStore **2.0.2**, released 23 July 2026.

> If you run any version of EasyStore earlier than 2.0.2, assume your shop is affected and update now. Two of the three flaws need no login, so there is no second precondition for an attacker to satisfy: if the shop is online, the endpoints are reachable.

How do you update EasyStore 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 first.
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 EasyStore 2.0.2. If you manage more than one shop, use the mySites.guru mass update feature to upgrade EasyStore across every affected site from one dashboard.
3. **Confirm the version.** After updating, check EasyStore reports 2.0.2 or later.
4. **Clear caches.** Clear Joomla's cache and any CDN or page cache.

Updating closes the door. It does not undo any data an attacker may already have read, so if you handle customer data and were on a vulnerable version for a while, treat credentials and exposed data as potentially known, and review recent orders for any that were marked paid without a matching payment.

How do I find every EasyStore shop 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. 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 EasyStore once and get back every connected site running it, the version each one is on, and whether an update is available.

View every EasyStore install across your sites

Open your Extension Inventory

Search for EasyStore across every connected Joomla site and filter for anything earlier than 2.0.2. Not a subscriber? Sign up free and connect your sites.

Why ecommerce extensions carry extra weight

A shop is not a brochure. When a page builder has a bug, the exposure is the site's data. When an ecommerce component has one, the exposure is the site's data plus its customers' personal and order details, plus, in the order-forgery case, real money. That raises the stakes on exactly the code that has to accept anonymous requests to work: the cart, the checkout, the payment callbacks.

We have found the same shape of issue across the Joomla extension ecosystem this year, in page builders, form builders, calendars and newsletter systems: a public endpoint that reaches a sensitive operation with too little checking in between. The lesson is not "avoid these tools". The reputable ones are usually well maintained, and the EasyStore fix itself is solid. What let this one down was not the code but the handling: a set of critical flaws quietly patched with no security notice, which leaves the shop owners who most need to update fastest with the least reason to think they must. Keeping these extensions updated is security-critical, not just feature maintenance, and that is far easier when something is watching the versions for you.

Stay ahead of the next one

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 these. 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, which is harder still when a critical fix ships without a word about security.

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.

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

Two of these three flaws are reachable by an anonymous visitor over the network with no privileges: the order forgery (a broken access control that changes order state and payment status, CWE-862 and CWE-639) and the SQL injection (CWE-89). The third, cross-customer order and invoice disclosure, is an insecure direct object reference (CWE-639) reachable by any authenticated customer.

We score every issue under CVSS 4.0. The two unauthenticated flaws come out at **8.7, High**, which is the top of the High band. CVSS 4.0 caps a single maximal impact reached with no privileges there rather than in the Critical range, but a full-database read or a free-order forgery is a total compromise of what it touches, and we treat it with the urgency that implies.

8.7
CVSS 4.0

HIGH

Highest of three findings: unauthenticated order forgery and unauthenticated SQL injection, 8.7 each

Reachable over the internet with no login. One flaw marks any order paid and rewrites arbitrary order records; the other reads the entire site database, password hashes included. A third lets any logged-in customer read every other customer's invoice.

No login needed | Exploitable over the internet | Free orders and order tampering | Full database read

Each finding, scored under CVSS 4.0:

Finding	Vector (`CVSS:4.0/...`)	Score	Severity
Unauthenticated order/payment forgery	`AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:H/VA:N/SC:N/SI:N/SA:N`	8.7	High
Unauthenticated SQL injection (full DB read)	`AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N`	8.7	High
Cross-customer order/invoice IDOR	`AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N`	7.1	High

We scored the SQL injection on confidentiality alone, because it is a read through the sort clause. A site where it could be chained into a writeable context would score higher. We reported all three to JoomShaper as Critical; the customer-invoice IDOR scores 7.1 under CVSS 4.0 because it needs a login, but its practical impact, the whole customer base's personal data, is why it leads this post.

Field	Detail
Component	EasyStore for Joomla (`com_easystore`)
Vendor	JoomShaper (joomshaper.com)
Type	Cross-customer IDOR data exposure (CWE-639); unauthenticated order forgery / broken access control (CWE-862, CWE-639); unauthenticated SQL

Field	Detail
	injection (CWE-89)
CVSS 4.0	8.7 (High) for both unauthenticated flaws; 7.1 (High) for the customer-invoice IDOR
CVE	Applied for via the Joomla CNA (pending assignment); JoomShaper did not request one
Impact	Cross-customer personal-data and invoice disclosure; free orders and arbitrary order tampering; anonymous read of the whole database including password hashes
Finder	Phil Taylor, mySites.guru
Affected versions	2.0.1 and all earlier versions
Fixed in	EasyStore 2.0.2
Reported	21 July 2026 (privately to JoomShaper)
Acknowledged	22 July 2026 (JoomShaper reproduced and verified all three)
Fixed	23 July 2026, EasyStore 2.0.2 (shipped with no security advisory)

Further Reading

- Our Quix Page Builder SQL injection disclosure and the wider month of Joomla security disclosures - the same class of bug across the ecosystem.
- Why AJAX endpoints are a CMS security blind spot - the recurring public-endpoint pattern behind flaws like these.
- How to check your Joomla database security - what to review and rotate if a vulnerable extension was ever live.
- CWE-639: Insecure Direct Object Reference and CWE-89: SQL Injection - the canonical references for these weakness classes.
- JoomShaper - the vendor's site, where EasyStore 2.0.2 is available.

Frequently Asked Questions

What was the worst EasyStore vulnerability?

Before EasyStore 2.0.2, any logged-in customer could read every other customer's order and printable invoice, with full name, email, billing and shipping address, phone number and full purchase history, simply by changing the order id number in the URL. On a shop with open customer registration, which is most shops, that means anyone could walk the whole customer base's personal data. mySites.guru found and reported this, along with two unauthenticated flaws: an order-forgery bug that marked orders paid for free, and a SQL injection that read the entire database. All three are fixed in EasyStore 2.0.2.

Do I need to log in to exploit these?

The invoice and order exposure needs any logged-in customer account, which on a shop with open registration is trivial to obtain. The other two need no login at all: the order-forgery and the SQL injection are both reachable by anonymous visitors with no account and no stolen credentials. That is what makes this set serious for any live EasyStore shop.

Which EasyStore versions are affected?

EasyStore 2.0.1 and earlier are affected. JoomShaper fixed the issues in EasyStore 2.0.2, released 23 July 2026. If you run any earlier version, assume you are affected and update now.

Did JoomShaper release this as a security update?

No. JoomShaper shipped the fixes in EasyStore 2.0.2 the day after acknowledging our report, but the release carries no security advisory and the changelog never says security release. The three fixes are worded as routine maintenance lines. We downloaded 2.0.2 and confirmed in the code that all three holes are properly closed, and because JoomShaper did not request a CVE, mySites.guru has applied for one through the Joomla CNA so there is a public, citable record.

How do I know if my shops are affected?

If you run EasyStore on a version earlier than 2.0.2, assume you are affected and update. mySites.guru flags every connected Joomla site running a vulnerable version automatically, so you get told which shops need attention rather than checking each one by hand.

Could someone have read customer data or got free orders before the fix?

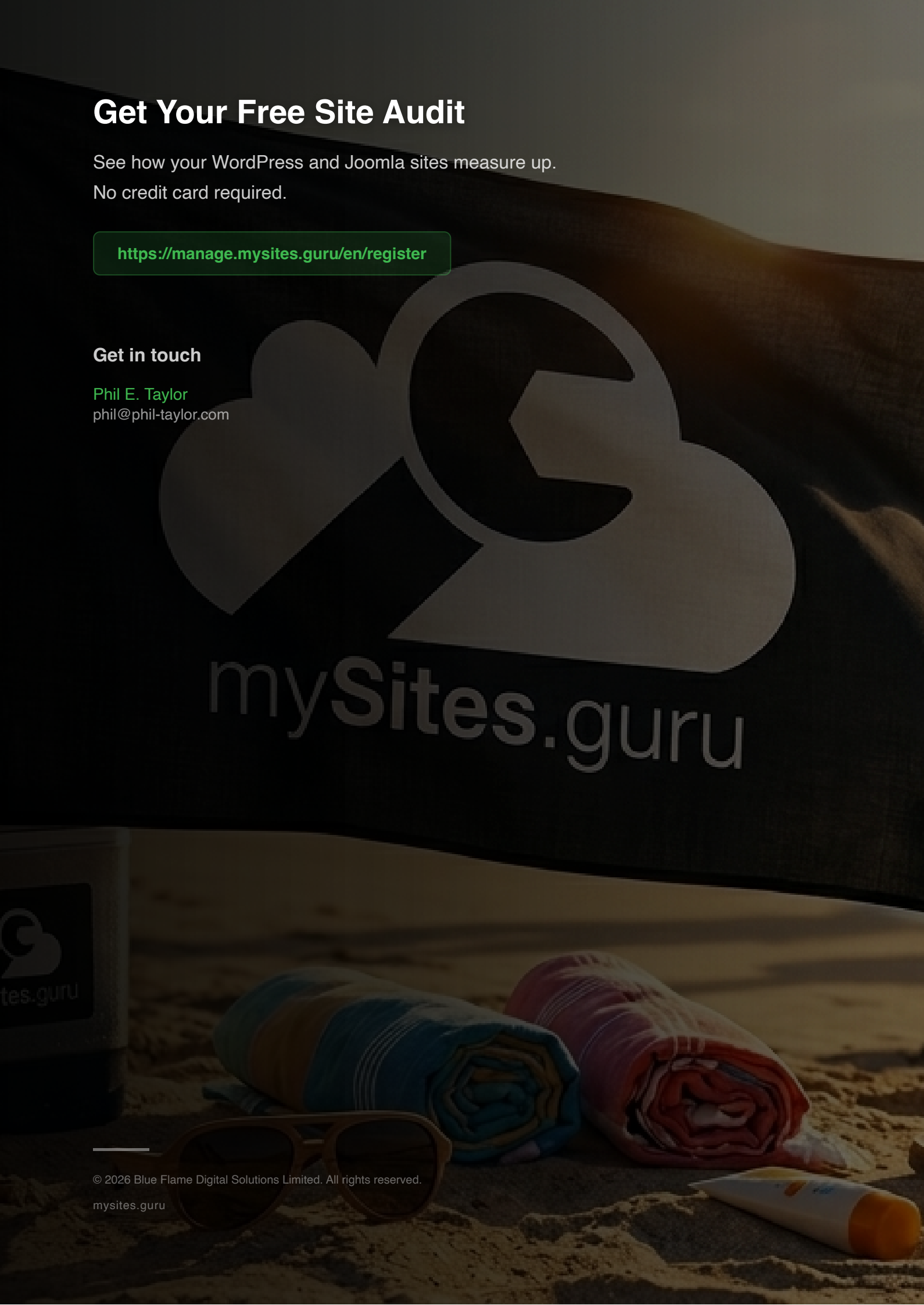
We have no evidence of exploitation before disclosure, and we withheld the exact requests and any proof-of-concept until the fix shipped. That said, all three flaws leave little trace, so if you ran a vulnerable version on a busy shop, treat every customer invoice as potentially read, review recent orders marked paid for a matching payment, and assume database contents including password hashes may be known.

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