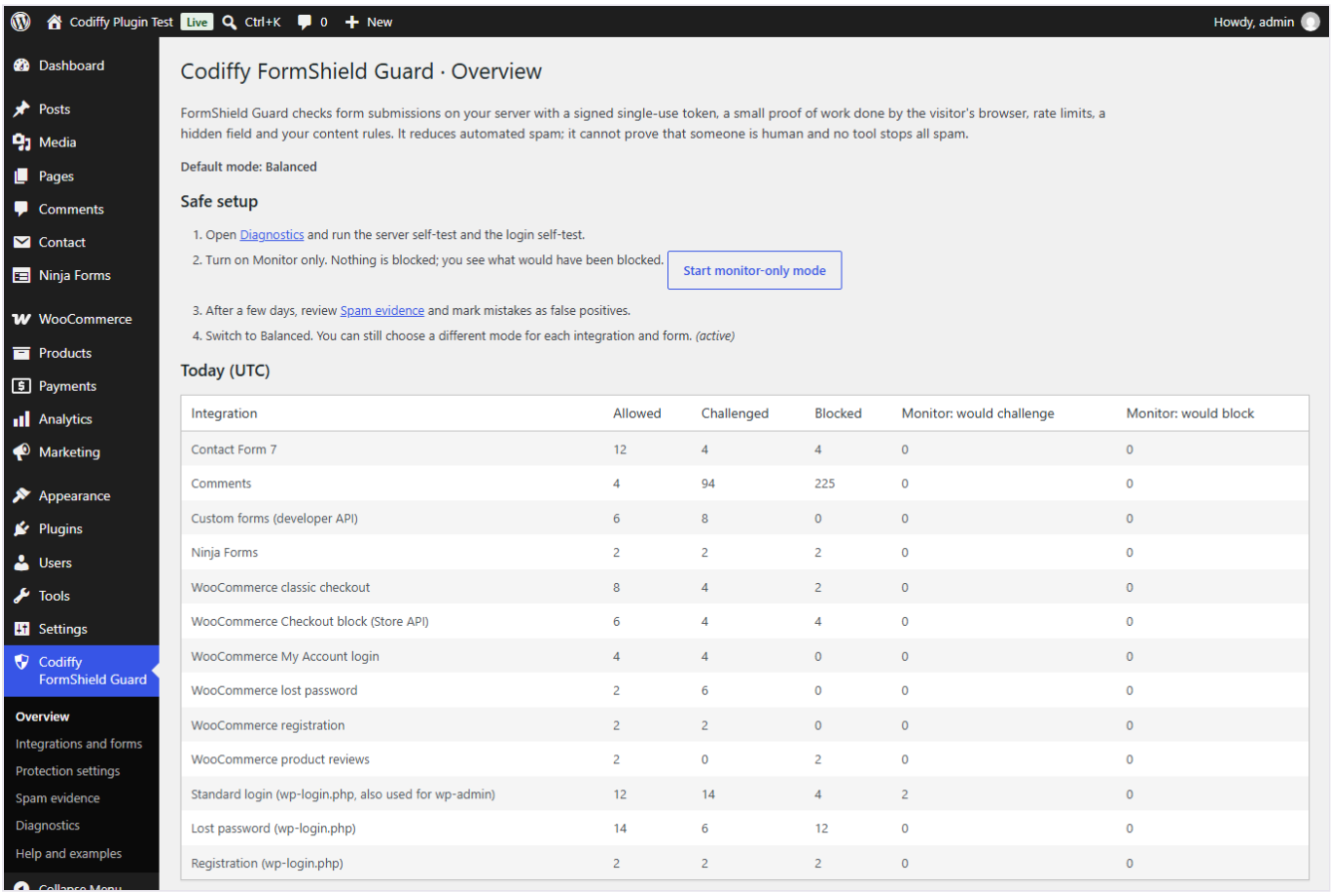
Codiffy FormShield Guard documentation

Item	Details
Plugin	FormShield Guard 1.0.0
Author	Haroon, Codiffy (https://codiffy.com/codiffy-teams/muhammad-haroon-ur-rasheed/)
Requires	WordPress 6.5+, PHP 7.4+ (tested: WordPress 7.1.4, PHP 8.3.33)
Optional	WooCommerce 8.0+ (tested 11.2.1), Contact Form 7 (tested 6.2.4.1), Ninja Forms (tested 3.15.7), Gravity Forms (tested 3.2.1)
External services	None

1. What this plugin does



The Codiffy FormShield Guard Overview screen with the safe-setup steps and statistics

FormShield Guard protects your site's forms from automated abuse: fake logins and accounts, password-reset floods, comment and review spam, contact-form spam and fake or card-testing orders. It works with several layers that run on your own server:

- **FormShield Verify**: every protected form gets a signed, single-use token from your site. The visitor's browser solves a small calculation (a "proof of work", usually well under a second) before the form is

sent. Bots have to pay that cost for every attempt; suspicious visitors are asked for a harder one.
- **Server-side checks** inside each form plugin's own validation, before e-mails, accounts, orders, payments or webhooks happen.
- **Rate limits**, a **hidden "leave empty" field** and **content rules** (links, domains and phrases you choose, repeated text).

It reduces spam; it does not prove that someone is human, and no tool stops all spam. A person (or a well-funded bot) using a real browser can still get through, and content rules only catch what they are told to look for.

2. Who it is for

Site owners who want fewer spam submissions without sending visitors to a third-party CAPTCHA, and developers who want one consistent protection layer for many kinds of forms, including their own.

3. Key features

- WordPress: wp-login.php (which also protects access to wp-admin), front-end `wp_login_form()`, registration, lost password, comments.
- WooCommerce: My Account login, registration and lost password, classic checkout, Checkout block (Store API, including direct API calls), product reviews. Compatible with HPOS and legacy order storage.
- Contact Form 7, Ninja Forms and Gravity Forms, each form separately.
- Your own forms through a small PHP/JavaScript API.
- Modes per integration and per form: Disabled, Monitor only, Balanced, Strict.
- Works with full-page caching; works without JavaScript through a verification code (where allowed).
- Evidence dashboard with reasons, export, deletion, false-positive marking, retention limits and privacy tools.
- Safe login setup: a self-test in your browser before login protection can be enforced, plus server-side recovery.

4. Installation

1. Plugins → Add New → Upload Plugin → choose `codiffy-formshield-guard-1.0.0.zip` → Install → Activate. Activation creates three database tables and changes nothing else: every integration starts Disabled.
2. Open **Codiffy FormShield Guard** → **Diagnostics**. Click **Run server self-test** (all checks should say PASS) and **Run login self-test** (it should say "Passed").
3. On **Overview**, click **Start monitor-only mode**. Forms now show the widget and every decision is recorded, but nothing is blocked.
4. After a few days, look at **Spam evidence**. If real people appear, adjust the settings or mark them as false positives.
5. Click **Switch to Balanced** (or set modes per integration).
6. If your site is behind a proxy, load balancer or CDN, add its addresses under **Protection settings** → **Trusted proxies** (Diagnostics shows when forwarding headers are present).

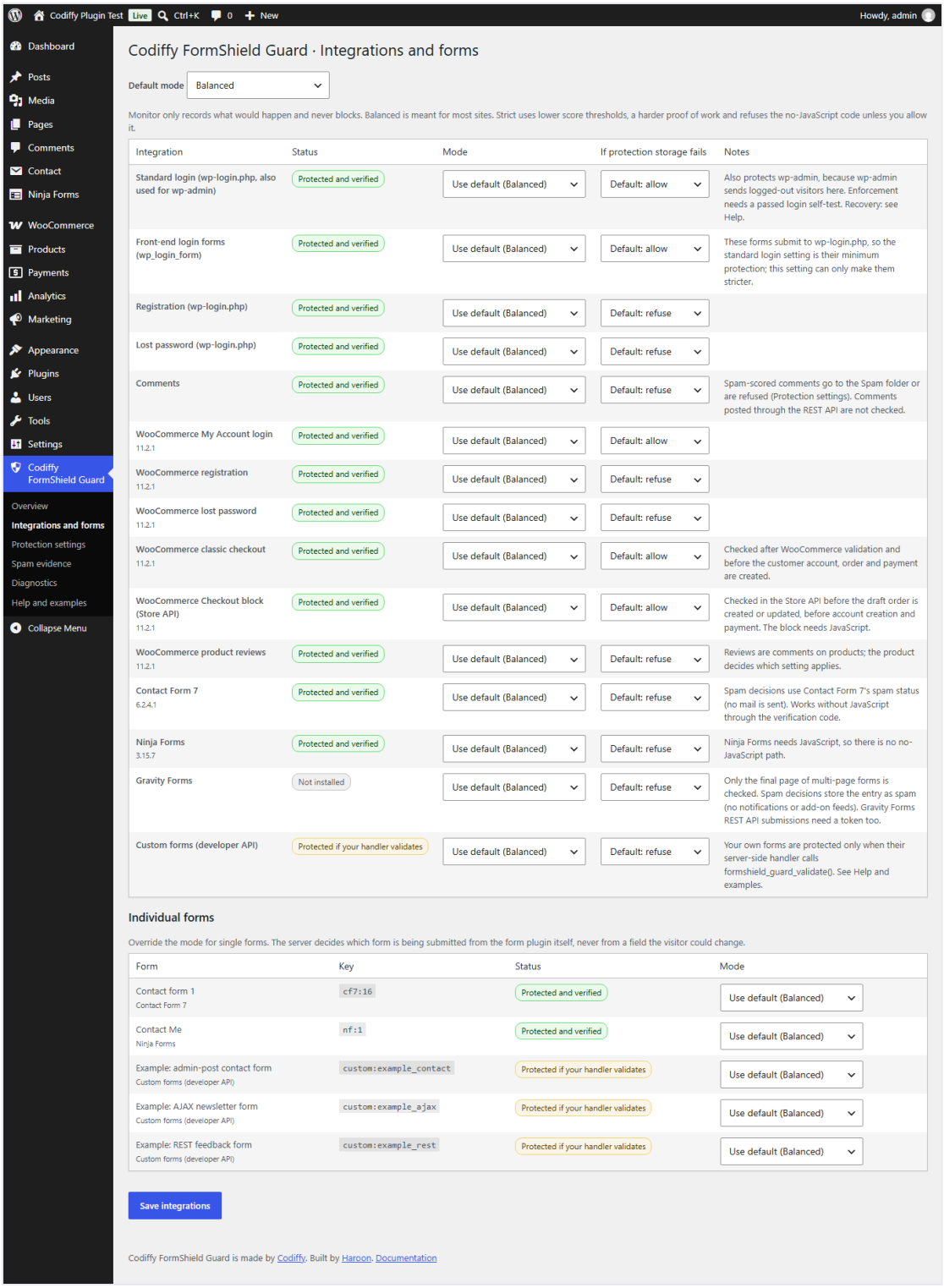
The ZIP contains everything; no Composer or Node.js is needed on the server.

5. Setting it up

Modes

Mode	What happens
Disabled	No widget, no checks
Monitor only	Widget and checks run; decisions are recorded as "would challenge" / "would block"; every submission continues
Balanced	Missing or bad tokens are refused; risk score 40+ asks for a harder proof of work, 80+ blocks
Strict	Score 25+ asks for a harder check, 60+ blocks; base proof of work is 4× harder; the no-JavaScript code is refused unless allowed

Integrations and forms



Integrations and forms: a mode for each integration and for single forms

Each integration has a mode (or "Use default") and a choice for **If protection storage fails**:

Integration	Default when storage fails	Why
Logins (wp-login.php, front-end, WooCommerce)	Allow	Avoid locking everyone out; WordPress still checks passwords

Integration	Default when storage fails	Why
Checkout (classic and block)	Allow	Avoid losing orders; WooCommerce and your gateway still validate
Registration, lost password, comments, reviews, form plugins, custom	Refuse ("try again in a few minutes")	Avoid spam floods while protection is down

Storage failures are shown on Diagnostics and recorded with the reason `infra_storage_error`; they are never counted as spam.

Notes per integration:

- **Standard login** also covers wp-admin: logged-out visitors to wp-admin are sent to wp-login.php. It cannot be enforced until the login self-test passed in the last 7 days.
- **Front-end login forms** (`wp_login_form()`) submit to wp-login.php. The standard login setting is their minimum; the front-end setting can only make them stricter, because a bot could always post to wp-login.php directly.
- **Comments and reviews**: reviews are comments on products, so the product decides which setting applies.
- **Classic checkout** is checked after WooCommerce's own validation and before the customer account, the order and the payment. "Update totals" requests are not checked.
- **Checkout block** is checked in the Store API (`POST /wc/store/v1/checkout`, also through `/batch`) before WooCommerce creates or updates the draft order, before account creation and before payment. WooCommerce may already hold a draft order from an earlier step of the same visit; a refused checkout never turns it into a real order, takes payment or sends order e-mails. WooCommerce removes old drafts on its own schedule.
- **Contact Form 7**: problems a visitor can fix appear as a message next to the form; spam decisions use Contact Form 7's own "spam" status (no mail is sent; Flamingo, if installed, keeps it as spam).
- **Ninja Forms** needs JavaScript to show forms, so the no-JavaScript code is not available there.
- **Gravity Forms**: only the final submission of a multi-page form is checked (moving between pages and "save and continue" are not). Problems a visitor can fix appear with Gravity Forms' validation message; spam decisions store the entry with Gravity Forms' Spam status, so no notifications are sent and no add-on feeds (webhooks, CRMs, payments) run. Gravity Forms still fires its `gform_after_submission` action for spam entries, so custom code hooked there should check the entry status. Normal, AJAX and `admin-ajax.php` submissions are covered; submissions through the Gravity Forms REST API need a token too.

Individual forms (Contact Form 7, Ninja Forms, Gravity Forms and registered custom forms) can override the integration's mode.

Protection settings

Setting	Default	Notes
Token lifetime	120 minutes	Open forms refresh their token before it expires
Fast-submit signal	3 seconds	Weak signal (password managers are fast too); 0 = off

Setting	Default	Notes
Hidden field	On	Not reachable by keyboard or screen readers; filled = strong but not conclusive signal
No-JavaScript code	Allowed in Balanced/Monitor, refused in Strict	See section 6
Maximum links	3	More links ask for a harder check; more than twice as many, together with any other signal, block
Repeated text	On	Same text to one form more than 3 times an hour
Domains to flag	empty	Links or e-mail addresses on these domains (and subdomains); one match blocks
Phrases and patterns to flag	empty	Phrases, or `/regex/flags`; one match asks for a harder check, two block
Spam-scored comments	Save to Spam	Or refuse them; comments failing token checks are always refused
Rate limits	see table on the page	Per network (IPv4 address or IPv6 /64) with soft (harder check) and hard (refuse) limits; sessions get 60%
Trusted proxies / header	none / X-Forwarded-For	Forwarding headers are read only from these addresses
Keep events	30 days, 50,000 events	Cleanup runs hourly
IP addresses	Do not store	Or truncated, or full
Content samples	Off	Allowlist of field names, redacted, max 500 characters each
Delete data on uninstall	Off	See section 12

Content rules never judge language or alphabet and never open submitted links.

6. What your visitors see

Usually nothing: the check runs in the background while they fill in the form. If they submit before it finished, the form waits a moment ("Checking your browser before sending…") and then sends itself. Messages are short and never say which check failed:

- "We could not verify this submission. Please wait a moment and try again…" (missing or expired token)
- "One more verification step is needed. Please wait a moment and submit again." (harder check requested)
- "Too many attempts. Please wait N minutes and try again."
- "This form was already submitted…" (double submit)
- "Your submission could not be accepted…" (spam decision)
- "We cannot verify submissions right now…" (storage problem, refuse mode)

Without JavaScript (or when the browser cannot run the check), visitors see a link "Get a verification code (opens a new tab)". That page shows a button; after about three seconds it gives a 12-character code that works once, for about 15 minutes, in the same browser and network. This path is weaker than the automatic check (a scripted bot can fetch codes too), which is why Strict mode refuses it unless you allow it.

Keyboard, screen-reader, touch, autofill and password-manager users are supported: the hidden field is never focusable, nothing needs a mouse, dragging, colours or fast reactions, and status messages are announced politely.

7. Shortcodes

`[formshield_guard form="your_key"]` prints the widget for a registered custom form (see section 10). No other shortcodes or blocks are needed: supported form plugins get the widget automatically.

8. FAQ

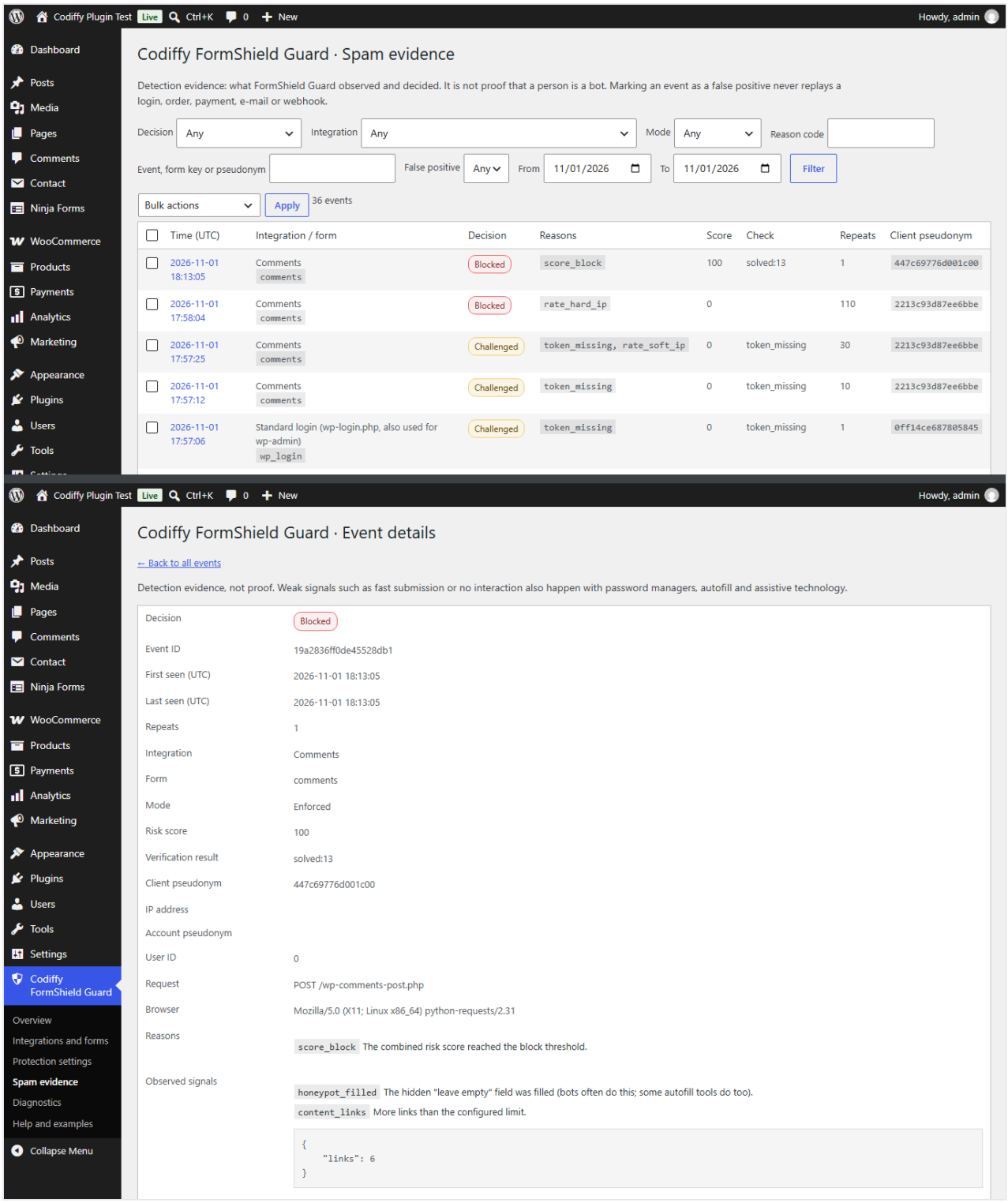
Does it contact Google or any other service? No.

Will it lock me out? Login enforcement needs a passed self-test. Accounts are never locked; at worst your network has to wait for the rate-limit window (15 minutes by default) or solve a harder check. See section 9 for recovery.

Does it work with caching plugins and CDNs? Yes. Pages contain no tokens. The challenge endpoint and code page send no-store headers; make sure your CDN does not cache `POST` requests or `admin-post.php`.

What is not protected? REST API logins and application passwords, XML-RPC, WP-Cron, payment-gateway callbacks, webhooks, other plugins' own login or registration forms, multisite `wp-signup.php`, WooCommerce "pay for order" retries, comments posted through the REST API, and any form whose handler does not call FormShield Guard.

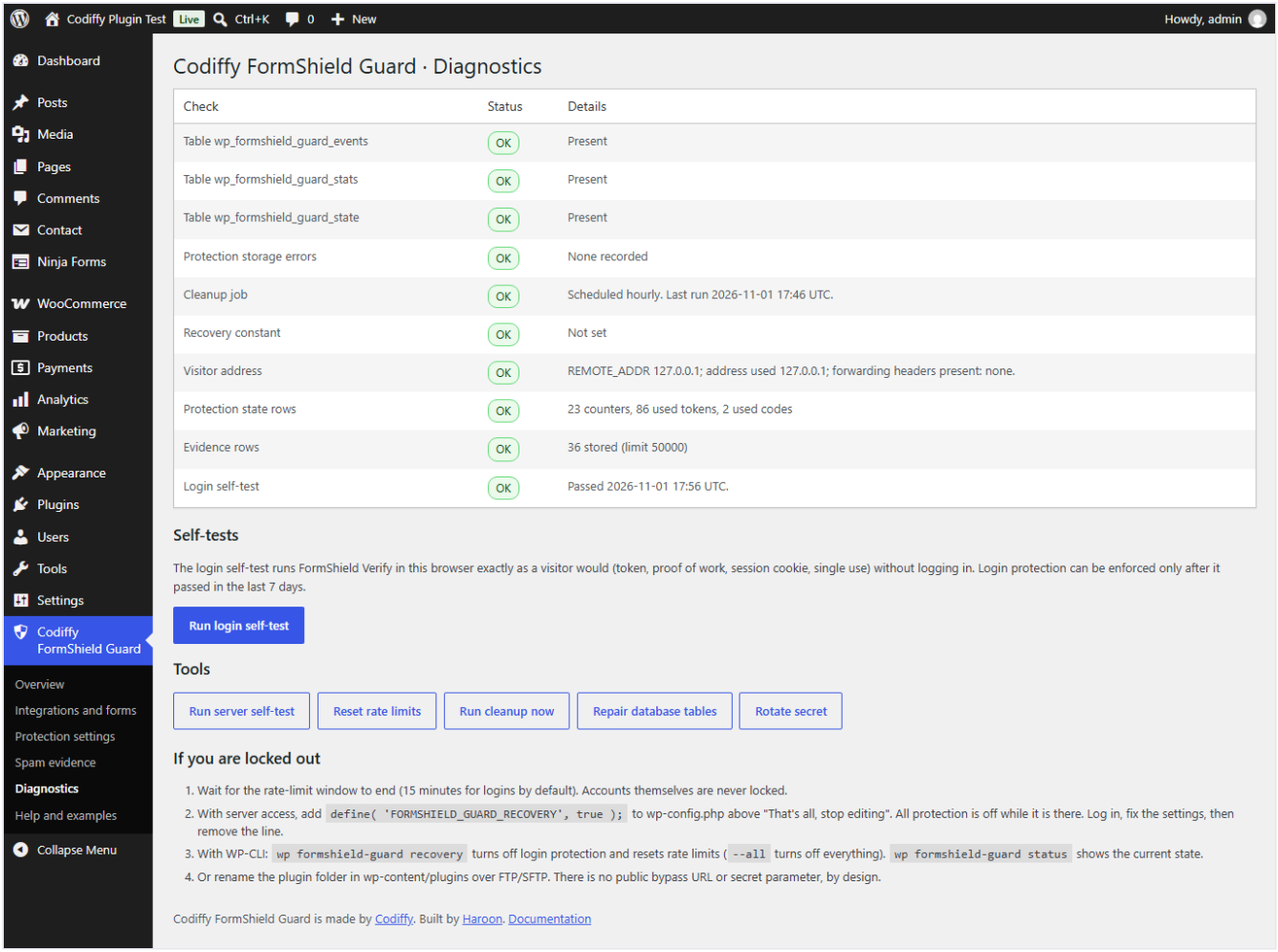
Why did a real person get blocked? Open the event under Spam evidence: each reason is explained. Weak signals (fast submit, no interaction) alone never block in Balanced mode. Mark the event as a false positive and ask the person to try again; nothing is replayed automatically.



Spam evidence: the event list with filters, and one event with every reason explained

Does it slow my site down? Pages are not changed except for a small script (about 10 KB) on pages with protected forms. On the test system a protected form submission took a few milliseconds and a few queries more than without the plugin (see TEST_REPORT.md).

9. Troubleshooting and recovery



Diagnostics: self-tests, tools and the steps to follow if you are locked out

Problem	What to do
Locked out of wp-admin	Wait 15 minutes, or add `define( 'FORMSHIELD_GUARD_RECOVERY', true );` to wp-config.php (above "That's all, stop editing"), log in, fix the settings and remove the line
Have WP-CLI	`wp formshield-guard recovery` (login protection off, limits reset), `--all` turns everything off, `wp formshield-guard status` shows the state, `wp formshield-guard mode wp_login monitor` sets one integration
No server access to wp-config or WP-CLI	Rename `wp-content/plugins/codiffy-formshield-guard` over FTP/SFTP; settings stay in the database
Everyone behind one office network gets "Too many attempts"	Raise the limits for that profile, or check Trusted proxies: behind a proxy without it, all visitors share the proxy's address
Diagnostics shows "Protection storage errors"	Click **Repair database tables**; check the database user's permissions
A pattern stopped working	Diagnostics names patterns that failed to run (too complex); simplify them

Problem	What to do
Forms show "The automatic check could not run"	The browser blocked the script or the REST API is blocked for visitors; the widget tries admin-ajax.php next. Check security plugins and CDN rules

There is no public bypass URL or secret parameter. Recovery always needs file, database or shell access.

10. For developers

See `docs/custom-forms.md` for the full API. In short:

```
formshield_guard_register_form( 'acme_contact', array( 'content_fields' => array( 'message' ) ) ); // on init
formshield_guard_render( 'acme_contact' );                                                          // inside <form>
$decision = formshield_guard_validate( 'acme_contact' );                                            // first in the handler
if ( ! $decision->allowed() ) { /* stop */ }
```

Hooks: `formshield_guard_decision` (action), `formshield_guard_capability`, `formshield_guard_base_bits` (filters). WP-CLI: `wp formshield-guard status|recovery|mode|reset-limits|cleanup|selftest`.

11. Privacy and data

- Nothing is sent to other services. No telemetry.
- Cookie `formshield_guard_sid`: random, HttpOnly, SameSite=Lax, deleted when the browser closes; links a token to the browser session.
- Evidence (blocked, challenged and monitor-only "would" events): time, form, decision, reason codes, score, signals, rate-limit numbers, a keyed pseudonym of the network address, a pseudonym of the username/e-mail used (logins, resets, registration, checkout), user ID if logged in, request method and path (no query string) and the browser's user-agent string. Optional: truncated or full IP address, redacted samples of allowlisted fields.
- Never stored: passwords, auth cookies, authorization headers, reset links or keys, session secrets, tokens or proof-of-work answers, verification codes, payment details, uploaded files. Sample field names that look sensitive are refused even if listed.
- Pseudonyms are keyed hashes: they cannot be reversed without the site secret, but they are still personal data (someone with the secret can test a guessed address). Pseudonymisation is not anonymisation.
- Retention (default 30 days) and a row limit (default 50,000) are enforced hourly. During floods identical events are combined into one row with a counter.
- Tools → Export/Erase Personal Data includes FormShield Guard events linked to the person's user ID or the pseudonym of their username/e-mail.
- Settings → Privacy → Policy Guide contains suggested text describing this behaviour.
- Exports (JSON/CSV) need the plugin's capability and a nonce; CSV cells that could run as spreadsheet formulas are neutralised.

12. Uninstalling

- **Deactivate**: the hourly cleanup job stops; settings and evidence are kept.
- **Delete** with "Delete settings and all evidence…" off (default): the temporary-state table and the cron job are removed; settings, the secret and evidence stay, so a reinstall continues where you left off.
- **Delete** with that option on: all three tables, the settings, the secret, the state option and comment links are removed.

13. Changelog

See `CHANGELOG.md`. 1.0.0 (2026-10-31): first release.

14. Support and about Codiffy

FormShield Guard is built by Haroon at Codiffy. Contact: https://codiffy.com/contact-us/.