
WOOCOMMERCE · STRIPE

Driftproof

*Finding where your store and Stripe
have stopped agreeing*

WHAT MAKES THIS CLASS OF BUG EXPENSIVE

nothing errors. no order turns red. the webhook returned **200**

What Driftproof does

WooCommerce and Stripe each keep a copy of the truth, and nothing in WordPress checks that they still match. Driftproof compares them and prices whatever disagrees.

It lists every relevant Stripe record for a window — about ten requests for a thousand-order month — joins them against your orders and subscriptions in memory, and runs twelve checks over the result. What comes back is a list sorted worst first, with the money at stake beside each line rather than a severity badge and a shrug.

Why the failures are silent

Almost nothing in this class announces itself. A webhook is missed, retried, or matched to nothing. The request returned 200. No order turns red, no email is sent, and the money is somewhere it should not be. You find out weeks later from a customer, a chargeback, or a month-end that will not reconcile.

Scanning changes nothing. It is read-only, it is free forever, and no amount of scanning can alter an order, a charge or a subscription. Repair is a separate, paid, deliberately awkward thing — covered in chapter six.

Getting started

Install it, open WooCommerce → Driftproof, and press Scan. There is nothing to configure first.

About your credentials

Driftproof does not ask for an API key. It reads the credentials your existing Stripe gateway already stores and proves they work against Stripe before claiming to have detected anything. Rotate your key and it follows, because no second copy is kept.

If you would rather narrow what it can reach, enter a Stripe restricted key instead. It will tell you up front which repairs that rules out, rather than failing halfway through one.

WooPayments is detected and then reported as unscannable. It never exposes Stripe API access, so there is nothing for Driftproof to compare your orders against. Saying so is more useful than quietly reporting no findings for that account.

How much a scan costs you

About ten Stripe requests for a thousand-order month. Driftproof lists each resource once across the window and joins in memory rather than looking up each order individually, which is what makes it usable on a store with tens of thousands of orders rather than only on a small one.

The findings screen

Everything found, grouped by what it means rather than by which check produced it, with a total you can act on.

Driftproof

Findings
Schedule
Connection

AT STAKE
AUD 5,011.00
Sum of every finding below

FINDINGS
11

LAST SCAN
7 seconds ago
Looking back 30 days

Export CSV

Last 30 days ▼
Scan again

WooCommerce Stripe Payment Gateway TEST acct_1Qi5nj2Ns42bcMb1

Money is moving, or about to be lost 8

● Order #11 has 2 identical charges; AUD 1,200.00 looks owed back to the customer.	AUD 1,200.00
● Order #11 was charged AUD 1,200.00 but totals AUD 44.70, a difference of AUD 1,155.30.	AUD 1,155.30
● Order #13 was charged AUD 99.99 but totals AUD 1,200.00, a difference of AUD 1,100.01.	AUD 1,100.01
● Order #13 has 2 identical charges; AUD 99.99 looks owed back to the customer.	AUD 99.99
● Order #15 is disputed for AUD 44.70 at Stripe but still reads as pending here.	AUD 44.70
● Order #15 has 2 identical charges; AUD 44.70 looks owed back to the customer.	AUD 44.70
● Order #11 is not marked paid, but Stripe collected AUD 44.70 for it.	AUD 44.70
● Order #15 was charged AUD 44.70 but totals AUD 42.50, a difference of AUD 2.20.	AUD 2.20

A customer is affected, or will be 3

● Order #11 shows no record of the AUD 1,230.00 partial refund Stripe has already returned.	AUD 1,230.00
● Order #15 shows no record of the AUD 44.70 full refund Stripe has already returned.	AUD 44.70
● Order #13 shows no record of the AUD 44.70 full refund Stripe has already returned.	AUD 44.70

▶ 7 checks ran, 5 could not

The line under the list — **7 checks ran, 5 could not** — is not a footnote. It is the difference between a clean store and a scan that could not look.

The twelve checks: money

All twelve run on the free plan. Each one comes from a failure documented against a real store, not from a list of things that could theoretically go wrong.

S1 Authorization about to lapse

A hold expires after about seven days and the money cannot then be recovered. WooCommerce does not track the expiry.

S1 Paid at Stripe, unpaid here

The webhook returned 200 and the order never moved.

S1 Charged twice for one order

A redirect racing a webhook, a retried renewal, a migrated server.

S1 Dispute not recorded

A chargeback with a response deadline, on an order still reading as a completed sale.

S1 Charged a different amount

Including the hundredfold case, where a zero-decimal currency is handled as though it had decimals.

S2 Refunded there, not here

Your reports, totals and tax figures are overstated until it is recorded.

The twelve checks: subscriptions

The half nothing else covers. A failed renewal is a customer who churns without ever complaining.

S1 No saved card

Active, renewing, and nothing to charge. It fails at the renewal.

S1 Card detached at Stripe

The id is saved; Stripe no longer holds that card for that customer.

S1 Out of step with Stripe

Cancelled at Stripe while you keep fulfilling, or cancelled here while Stripe keeps charging.

S2 Renewal overdue, nothing scheduled

The quietest of all. The date sits in the past and the subscription stops earning.

And two about configuration

S1 Test and live keys mixed up

A live store holding a test key rejects every real payment. The reverse charges real cards.

S2 Customer no longer exists

A subscription linked to a Stripe customer that was deleted, merged, or lost in a migration.

What it could not check

A check that could not read the data it needs is not evidence your store is clean. Reporting it as a pass is the one thing this plugin must never do.

Every scan states how many checks ran and how many could not, with the reason for each. The screenshot in chapter three shows five that could not — the four subscription checks and the orphaned-customer check — because WooCommerce Subscriptions is not installed on that store.

That is the correct answer. The alternative is a clean bill of health for four checks that never looked at anything, on a store whose subscriptions might be quietly failing.

Ran, found nothing

Genuine evidence. This part of your store agrees with Stripe.

Could not run

Named, with the reason. Not evidence of anything.

Failed

The check errored. Also named, and also not a pass.

Repairing, and the dry run first

Finding it is free. Fixing it is the paid tier, and every repair shows you its plan before it will do anything.

The dry run is not a button you have to remember to press. It is already on the screen: each repairable finding prints exactly what running it would do, in plain words, before you have decided anything. A rehearsal nobody has to ask for is one everybody reads.

Three classes, and they are treated differently

A — safe

Changes data in WooCommerce only. Backfills a missing card reference, records a refund Stripe already made.

B — emails

Marks an order paid, which may send the customer a receipt. Confirmed separately, with the email named.

C — moves money

Captures or refunds at Stripe. Reached only through a link, then a second screen that states the amount and the word irreversible.

Nothing irreversible ever runs on a schedule, and there is no bulk control on the findings screen. A loop that captures or refunds, interrupted halfway by a browser timeout, leaves a store nobody can reason about.

Every repair is written down

Including the ones it declined.

The repair log records what changed, when, which finding it came from, and who ran it — a username where there is one, `wp-cli` otherwise. Not a user id: the log outlives the user row, and "user 4" is useless once user 4 has been deleted.

A declined repair is logged too. Knowing that Driftproof considered a repair and refused it is more useful than silence, because the next person to look will otherwise wonder whether it was ever offered.

Applied once, and only once

Behind Stripe's own idempotency keys sits a second check: the log itself. Some repairs never reach Stripe at all — a metadata backfill has no idempotency key to rely on — and running one twice on a subscription somebody has since corrected by hand would overwrite their correction with stale data.

Scanning on a schedule

A scan you have to remember to run is a scan that happens once. The paid tier runs it for you and emails when something new turns up.

Set a frequency and Driftproof queues a recurring scan through Action Scheduler, fanning out to one run per account so that a store with two gateways gets two independent scans and one failing key does not cost you the other account's findings.

The schedule never repairs. It scans and it emails. Everything that changes anything still needs a person, on the screen, confirming it.

What the email says

Only what is new. A finding that was in yesterday's scan is not announced again, so a check that has been failing since Monday does not re-announce itself every morning until you start filtering the sender.

Worst first, then most money — the first line of an email is the only line guaranteed to be read, on a phone, in a notification preview.

If the licence lapses

The queue is cleared. A recurring action left behind with no callback runs forever at whatever the interval was; tidying up after a feature it can no longer perform is the plugin's job, not yours.

Getting it out

Findings and the repair log both export as CSV, from the screen or the command line.

The two are separate downloads because they answer different questions. Findings are what is wrong now; the repair log is what has been done about it. Somebody reconciling a month wants the second, and somebody triaging a Monday wants the first.

```
wp driftproof export > findings.csv
```

Byte for byte what the screen hands out, byte-order mark included.

A redirect, deliberately, rather than a `--file` option: a plugin writing to a path chosen by an argument is a plugin directory finding.

Both carry a byte-order mark so Excel opens them as UTF-8 rather than mangling every currency symbol and every name with an accent in it.

If something looks wrong

It says no gateway was found

Driftproof reads credentials from a Stripe gateway plugin. If yours is not one it recognises, or the store has since removed it, use the Connection tab to enter a restricted read-only key. The same tab is the route for a store that needs last quarter reconciled after the gateway is gone.

Several checks say they could not run

That is usually correct rather than broken. The four subscription checks decline when WooCommerce Subscriptions is not installed, because they have nothing to look at. The reason is printed next to each one.

A finding looks wrong to me

Open it. Every finding carries the Stripe id it came from and the order it matched, so the claim can be checked against Stripe directly rather than taken on trust. If it is wrong, that is a bug worth reporting with those two ids in it.

A repair did not appear for a finding

Not every finding has one. Some describe a state Driftproof can see but should not change — a dispute, for instance, is answered at your gateway and not here. The screen says so rather than leaving a button that would do nothing.

About this guide

Driftproof is a WooCommerce plugin by Allan Niñal. It runs entirely inside your WordPress installation and talks only to Stripe.

No third-party service sits in the path, and no copy of your gateway credentials is stored — the key is read at the moment it is needed, so rotating it takes effect immediately.

Requires WordPress 6.5 or later, WooCommerce, PHP 8.1 or later.

Free covers all twelve checks, every scan, every finding priced, read-only and forever.

Pro adds repair, scheduled scanning, email alerts and CSV export. Solo \$19, Store \$49, Agency \$99 per month, each with a 14-day trial. Annual is ten months for twelve.

Online www.allanninal.dev/driftproof/

The screenshot in chapter three is a real scan of a live test account — real Stripe ids, real amounts, and eleven findings the plugin actually produced. It is not a mock-up, and the five checks it reports as unable to run really could not run.