

GAJWAF · PLAYBOOK

Taming false positives

Stop that rule firing on that traffic — and nothing else

Confirm. Fix narrowly. Verify both ways.

A false positive is legitimate traffic the WAF flagged or blocked. This playbook walks you from the event to the narrowest fix that works, proves the fix didn't open a hole, and leaves a change record behind.

4 stages

Each with a go / no-go check

4 rungs

From narrowest to broadest fix

1 record

For your change log

AT A GLANCE

The method

GOLDEN RULE

Fix a false positive with the **narrowest exclusion that works** — not by lowering the whole site's sensitivity or disabling a feature.

A broad relaxation to silence one page opens every other page to the same attack.

THE FIX LADDER

Start at the top. Stop at the first rung that works.

▲ NARROWEST — PREFER

a	"Not an attack?" → scoped exclusion	Default — the right fix for most false positives
b	Purpose-built field / path setting	When a feature has its own allow-list
c	Custom allow rule	A well-defined trusted source or path
d	Raise the blocking threshold	Only when the whole site is too strict

▼ BROADEST — LAST RESORT

✗ DON'T

- **Don't** turn a whole detection category off to silence one page.
- **Don't** lower the global threshold as a first move — it weakens the entire site.
- **Don't** add a broad **allow** custom rule matching more than you mean to — it becomes a bypass.
- **Don't** mark events "solved" as a fix — that's triage bookkeeping, not an exclusion; the rule still fires.

IN THIS PLAYBOOK

Contents

1	Confirm it really is a false positive	3
2	Apply the narrowest fix	4
3	Verify the fix	5
4	Roll the fix to every node.....	5
✓	Change record	6

WHO **Admin** to add exclusions; **analyst** to switch modes if needed.

STAGE 1 OF 4

1 Confirm it really is a false positive

SITE MODE **BLOCK — DROP TO DETECT IF UNSURE** WHO Analyst

Open the event in **Events** — filter by site, then by the user, IP or path that's affected — and read the detail:

Ask	Look at	What it tells you
What was the request?	The request line, user and client	A real user action (a form post, an upload, an API call your own app makes) — or something that genuinely looks like an attack?
Which finding fired?	Finding rule ID and category	Which rule to scope the fix to.
Score vs. threshold	The event's score and action	Was it blocked outright, or did it accumulate score across several findings?

IMPORTANT

- If the request is genuinely malicious, it is **not** a false positive — leave it blocked.
- If you're unsure, drop the site to **Detect** while you investigate so nobody is blocked meanwhile. Detect never blocks.

COMMON GENUINE FALSE POSITIVES ON REAL APPLICATIONS

Symptom	Cause	Fix
hidden-field-tampered on Matomo / analytics	Matomo reuses module / action as both hidden form fields <i>and</i> AJAX parameters.	Add module, action to Detection → "Fields your JavaScript changes" .
API / mobile / webhook clients challenged	Non-browser clients can't solve a browser check.	List their paths under Bot check → "Never check these paths" .
bot-biometrics Medium everywhere	Benign — a human navigated without moving the mouse.	Informational; it does not ban. No action needed.
A WYSIWYG / rich-text field blocked as XSS or SQLi	The field legitimately contains markup.	A scoped exclusion on that field / path (rung a).

✓ MOVE ON WHEN

You have confirmed the request is legitimate, and noted its **Finding rule ID, category and path**.

STAGE 2 OF 4

2 Apply the narrowest fix

SITE MODE **DETECT OR BLOCK** WHO Admin

Work down the ladder and **stop at the first rung that fixes it**. Each rung down is broader than the one above.

a "Not an attack?" → scoped exclusion DEFAULT CHOICE

WHERE Event detail → "Not an attack?". Manage them all under **Edit configuration → Exclusions**.

The WAF proposes an **exclusion scoped to that finding and path**: it stops *that rule* firing for *that matching request*, leaving every other rule and path fully protected. Review it and save.

TIP

This is the standard fix — the right answer for the large majority of false positives.

b A purpose-built field / path setting WHEN ONE EXISTS

Some features have a dedicated allow-list that is cleaner than a generic exclusion:

Setting	Config key	Use it for
Detection → "Fields your JavaScript changes"	<code>hidden_fields.exempt</code>	Form fields your own scripts rewrite.
Bot check → "Never check these paths"	<code>challenge.exempt_paths</code>	APIs, webhooks, health checks, mobile endpoints.
Sign-in → "Public paths (no sign-in)"	<code>auth_gateway.exempt</code>	Pages that must stay open without a login.

c A custom allow rule SPECIFIC CASES

WHERE Edit configuration → Custom rules, with action **allow**.

For a well-defined trusted source or path. **Keep the match tight** — a loose allow rule becomes a bypass.

d Raise the blocking threshold LAST RESORT

WHERE General → "Blocking sensitivity (score)" — higher = more lenient.

Only when the site **as a whole** scores legitimate traffic too aggressively. Raise it a little, never to paper over one page — and re-review afterwards.

✓ MOVE ON WHEN

A fix is saved at the **highest rung** (narrowest scope) that resolves the false positive.

STAGE 3 OF 4

3 Verify the fix

SITE MODE **DETECT → BACK TO BLOCK** WHO Admin + analyst

1. If you dropped the site to **Detect**, keep it there for one more cycle.
2. **Re-run the action that was blocked** — replay the request, or ask the user to retry. Confirm it now passes.
3. **Check Events — both ways.** The finding should no longer fire for that traffic, **and** it should still fire for an actual attack against the same endpoint. That proves you didn't over-broaden. (The *GAJWAF demo playbook* has safe test attack requests.)
4. **Return the site to Block.**
5. Optionally **Mark solved** on the handled events.

NOTE

Mark solved is triage bookkeeping only — it records that you dealt with the events. It does **not** change what the WAF blocks.

✓ **MOVE ON WHEN**

- The legitimate request passes.
- A real attack on the same endpoint is still caught.
- The site is back in **Block**.

STAGE 4 OF 4

4 Roll the fix to every node

SITE MODE **UNCHANGED** WHO Admin

Exclusions are **configuration**. If you run several GAJWAF nodes, roll the updated config to **every** node — use the console's **Export / Import** — or the false positive persists on the nodes that didn't get it.

✓ **MOVE ON WHEN**

Every node carries the same exclusions. (Single node? Nothing to do.)

FOR YOUR CHANGE LOG

 Change record

Site	
Event / Incident ID	
Finding rule ID & category	
Path / field affected	
Fix applied (rung a–d)	☐ a ☐ b ☐ c ☐ d (see the fix ladder, page 2)
Legit request now passes	☐ Yes
Real attack still caught	☐ Yes
All nodes updated	☐ Yes ☐ Single node
Site back in Block	☐ Yes
Done by / date	

Related documents

Document	Use it for
Onboard a site safely (playbook)	Reviewing false positives <i>before</i> you enforce a new site.
GAJWAF Troubleshooting & Support Manual	Chapter C — "Legitimate traffic is being blocked", with fixes by finding family.
GAJWAF Features & Configuration Guide	The Detection, Bot check, Exclusions and Custom rules tabs (chapters 4, 7, 10, 11).