

GAJSHIELD · WEB APPLICATION & API FIREWALL

GAJWAF

Troubleshooting & Support Manual

For GAJWAF administrators

Read what GAJWAF is telling you — and fix it

How to read GAJWAF's evidence, the problems that actually come up in the field, and how to resolve each one. Organised by **symptom** first, followed by reference tables for event actions and finding categories, and a checklist for contacting GajShield Support.

8 chapters

Organised by symptom

2 references

Event actions, finding types

1 checklist

Before you contact Support

About this document

Document	GAJWAF Troubleshooting & Support Manual
Version	1.0 · October 2026
Applies to	GAJWAF — AWS Marketplace, Azure Marketplace and BYOL editions
Published by	GajShield Infotech Pvt. Ltd., Mumbai, India

COPYRIGHT

© 2026 GajShield Infotech Pvt. Ltd. All rights reserved. This document is provided for the use of GAJWAF customers and partners. No part of it may be reproduced, distributed or transmitted in any form or by any means without the prior written permission of GajShield Infotech Pvt. Ltd.

DISCLAIMER

The information in this document is provided "as is", without warranty of any kind, express or implied, and is subject to change without notice. Features, menu labels and default values may differ between GAJWAF releases. While every effort has been made to ensure its accuracy, GajShield Infotech Pvt. Ltd. accepts no liability for any loss, damage or service disruption arising from the use of, or reliance on, this document.

USE OF THIS INFORMATION

Apply configuration changes in accordance with your organisation's change-control procedures, and validate them in your own environment before relying on them in production.

TRADEMARKS

GAJWAF and GajShield are trademarks of GajShield Infotech Pvt. Ltd. Amazon Web Services, AWS and AWS Shield are trademarks of Amazon.com, Inc. or its affiliates. Microsoft and Azure are trademarks of the Microsoft group of companies. All other product and company names mentioned in this document are trademarks of their respective owners; they are used for identification purposes only, and their mention does not imply endorsement.

IN THIS MANUAL

Contents

1	Before you start	5
1.1	The first five questions	5
1.2	How to read the evidence	5
A	Customers can't reach the site	6
A.1	"Misdirected Request" / HTTP 421	6
A.2	Browser warns about the certificate	6
A.3	502 / 503 / 504 — upstream-error	6
A.4	Real client IP not reaching the backend	6
B	Console unreachable or can't sign in	8
B.1	Console won't load	8
B.2	Can't sign in for the first time (cloud)	8
B.3	Locked out, lost admin, or 2FA device gone	8
C	Legitimate traffic is being blocked	9
C.1	General method	9
C.2	Tuning by finding family	9
C.3	Was it blocked or only observed?	10
D	Bot check & browser challenge problems	11
D.1	A user is challenged over and over	11
D.2	bot-biometrics Medium events everywhere	11
E	Authentication gateway & zero-trust	12
E.1	Symptoms and fixes	12
F	Attacks are getting through	13
G	Data-leak prevention (DLP)	13
H	Operations	14
H.1	Multi-node / HA — the one setting that matters	14
H.2	Config editor problems	14
H.3	Installed certificate disappeared after "Save & apply"	14
H.4	Licensing	14
H.5	Updates	14
H.6	Performance / high CPU	15
R	Reference	16
R.1	Event action values	16
R.2	Finding categories	16
	Contacting GajShield Support	18
	Related documents	18

AT A GLANCE

How problems break down

GOLDEN RULE

GAJWAF is a reverse proxy. Traffic hits the WAF, it inspects, then forwards clean requests to the backend. Almost every problem falls into one of four classes:

1 Routing

The WAF can't match the request to a site.

→ [Chapter A](#)

2 The backend

The app behind the WAF is unreachable or wrong.

→ [Chapter A](#)

3 A false positive

The WAF flagged legitimate traffic.

→ [Chapters C – E](#)

4 Multi-node

Nodes that don't share state.

→ [Chapters D, H](#)

SYMPTOM INDEX

Start from what the customer sees

Symptom	Go to	Page
Site returns 421 Misdirected Request	A.1	6
Browser shows a certificate warning	A.2	6
502 / 503 / 504 — upstream-error	A.3	6
App logs show the WAF's IP instead of the client's	A.4	6
Console on 9443 won't load	B.1	8
Can't sign in / locked out / lost 2FA	B.2	8
Legitimate traffic is blocked	C.1	9
User is challenged repeatedly	D.1	11
Sign-in loops or "session expired"	E.1	12
Attacks are not being blocked	F	13
DLP masking isn't happening	G	13
Re-challenges / session loss across nodes	H.1	14

CHAPTER 1

1 Before you start

1.1 The first five questions

Establish these before diving in — they narrow 90% of tickets:

1. **Is the site in `block`, `detect`, or `off` mode?** Check the Sites page or the mode badge. In **`detect`** nothing is ever blocked or challenged — it only logs.
2. **One node or several?** Is it behind a load balancer or an HA pair? Multi-node issues (re-challenges, session loss) all trace to nodes not sharing `cluster.secret`.
3. **What does the event say?** Get the **Action**, the **Finding rule ID**, and the **Incident ID** from the event detail. Those three identify everything.
4. **One client or everyone?** One user (often behind a corporate proxy or privacy relay), or all traffic? One-client problems are usually identity/cookie churn.
5. **When did it start, and what changed?** A config save, a certificate renewal, a new release, a DNS cut-over, a node added.

1.2 How to read the evidence

Source	What it tells you
Console → Overview	Totals, "Attacks blocked", "Attacks observed but not blocked" (what <i>would</i> have been blocked), and backend/certificate health per site.
Console → Events	Every decision. Filter by site, action, user, IP, category, or triage status (Open / Solved). Each event's detail shows the action, score vs. threshold, client IP + network (Tor/hosting), device, user, TLS fingerprint, the request line, the stack detected, and the findings.
Console banners	A red banner at the top of the console flags sites that are "not being served" (disabled) and licence problems.
Site editor → Validate	Checks a site's settings without applying them. Use it before Save & apply whenever you change a site.

CHAPTER A

A Customers can't reach the site

A.1 "Misdirected Request" / HTTP 421

Meaning: no configured site matched the request's **Host** (and, if the site uses IP-based binding, the local IP it arrived on). The WAF refuses traffic it doesn't recognise.

CAUSES & FIXES, IN ORDER OF LIKELIHOOD

1. **Host name not configured** — the site's **Host names** list doesn't include the domain the client used (e.g. `www.` vs. apex, or a new hostname). Add it to the site and save.
2. **The site is disabled** — a site that failed to start (bad upstream URL, missing/invalid TLS certificate, a listener that won't bind, an auth-gateway misconfiguration) is **disabled**, and its hosts fall through to the same 421. Check the console's red banner ("not being served"), fix the flagged field in the site editor, and save.
3. **Transient / "it fixed itself"** — a certificate file briefly unreadable during a renewal that coincided with a reload can disable a site until the next automatic reload re-enables it. If a site returned 421 and recovered with no change, this was it — no action is needed.
4. **IP-based binding mismatch** — if the site has a **bind** list, the request must arrive on one of those local IPs and match the host. Check the **bind** value against the interface the traffic lands on.

A.2 Browser warns about the certificate

The site has no real certificate installed, so GAJWAF serves a self-signed one. Edit the site → **HTTPS certificate** → upload/paste the full-chain certificate + key → **Validate & install**. Alternatively, enable automatic Let's Encrypt (`tls.acme`) for an Internet-facing site.

IMPORTANT

Install the certificate with **Validate & install** — *not* by typing a path into **Save & apply**.

A.3 502 / 503 / 504 — upstream-error

The client reached the WAF fine, but the **backend** is the problem.

- Edit the site → **Test connection** to the upstream.
- Check the upstream URL/port in the site's **Backend servers**.
- Check the backend health for the site on **Overview** — a backend marked down is shown there.
- Confirm the backend allows inbound traffic from the WAF's IP (and isn't itself down).

A.4 Real client IP not reaching the backend

The application logs show the WAF's IP as the client. Configure the backend to read the forwarded IP: GAJWAF sets `X-Forwarded-For` (and `X-Forwarded-Proto`, `X-Forwarded-Host`) from the verified client.

Web server	Configuration
Apache	Enable <code>mod_remoteip: RemoteIPHeader X-Forwarded-For, RemoteIPTrustedProxy <WAF IP></code> , and <code>%a</code> in the log format.
nginx	Use the <code>real_ip</code> module with <code>set_real_ip_from <WAF IP></code> .

NOTE

If the WAF itself is behind another proxy or load balancer, add that proxy's address to `trusted_nets` so the WAF honours the inbound `X-Forwarded-For`. Otherwise it uses the direct peer, and geo rules, rate limits and bans key on the wrong IP.

CHAPTER B

B Console unreachable or can't sign in

B.1 Console won't load

You cannot open `https://<ip>:9443/`.

- **Firewall:** port **9443** must be open from your management IP. Templates restrict 9443 and 22 to the management CIDR by design.
- **Instance running?** Check the instance's status in the AWS or Azure console. A configuration error disables only the affected *site*, never the console.
- If the instance is running and port 9443 is open but the console still won't load, contact **GajShield Support** (see the checklist at the end of this manual).

B.2 Can't sign in for the first time (cloud)

- **User** `admin`; **password** = the instance's cloud ID (AWS EC2 Instance ID `i-...` / Azure vmId GUID). The password must be changed at first sign-in.
- If rejected, confirm you copied the ID of *this* instance (not another VM in the same account) with no extra spaces. If it is still rejected, contact **GajShield Support**.

B.3 Locked out, lost admin, or 2FA device gone

- If another console administrator can still sign in, they can manage your account under **Settings → Console users**.
- If no administrator can sign in, contact **GajShield Support** to have the admin account reset.

TIP

Create at least two named console administrators (Settings → Console users) so a lost password or 2FA device never locks everyone out.

CHAPTER C • FALSE POSITIVES

C Legitimate traffic is being blocked

This is the most common support category.

C.1 General method

1. Open the event. Note the **Finding rule ID** and **Target** (e.g. `ARGS:action`, `recon-sensitive-file`).
2. Confirm it is genuinely legitimate — the client, the path, the payload.
3. Resolve with the **least-broad** tool that fits (in order of preference below).

Tool	When to use it
"Not an attack?" on the event	Adds a path exclusion for that rule. The default precise fix.
Feature-specific exempt list	Preferred wherever one exists — see the table in C.2.
Lower the security level / raise <code>anomaly_threshold</code>	Only if many unrelated rules over-fire.
Put the site in detect	A safety valve while you tune — then switch back to block.

C.2 Tuning by finding family

Finding (rule ID family)	Why it false-positives	Precise fix
<code>hidden-field-tampered</code>	App reuses a form-field name as a dispatch/AJAX parameter (classic: <code>Matomo module/action</code>).	Add the field names to <code>hidden_fields.exempt</code> (Detection → "Fields your JavaScript changes"). Default mode is <code>report</code> (log only).
<code>api-unknown-path</code> , <code>api-schema</code> , <code>api-scope-denied</code>	OpenAPI spec is missing a path, is stale, or scope rules are too tight.	Update/re-upload the OpenAPI spec; set API mode to <code>report</code> while fixing; add the scope.
<code>geo-blocked-country</code> , <code>geo-not-allowed-country</code> , <code>geo-tor</code> , <code>geo-datacenter</code>	Legitimate user on a VPN/relay/hosting IP, or GeoIP mislocates the IP.	Adjust the site's Geo & reputation lists; allow the country; reduce <code>score_datacenter</code> .
<code>bot-biometrics</code> (Medium)	A human solved a check without moving the mouse — normal on a navigation; common on privacy relays.	Informational — never blocks. Leave it; see Chapter D for the re-challenge issue behind it.
<code>behavior-ratelimit</code> , <code>api-rate-limit</code>	Limits too low for a busy client or integration.	Raise <code>rate_limit.rps/burst</code> (or <code>auth_rps/auth_burst</code>); exempt the path.
<code>dlp-credit-card</code> , <code>dlp-custom-*</code> , <code>data-leak masks</code>	A response field matches a pattern but isn't sensitive.	Turn off the specific DLP rule under Response & limits , or narrow the custom pattern.

Finding (rule ID family)	Why it false-positives	Precise fix
<code>client-csp-violation</code> , <code>client-side</code>	A new or changed third-party script reported by the browser.	Use " Approve source " on the event to add the origin to <code>client_side.allowed_sources</code> — this, not "Not an attack?", is what unblocks it.
<code>ml-anomaly</code> , <code>ml-observe</code>	ML scored a request as suspicious.	Set ML to <code>monitor</code> ; investigate before <code>enforce</code> .
<code>recon-sensitive-file</code> , <code>recon-cms-mismatch</code> , <code>recon-tech-mismatch</code> , <code>scanner-*</code>	A real admin tool or health check hits a path that looks like a probe.	Exempt the path. Otherwise these are almost always genuine scanners.
<code>sqli</code> , <code>xss</code> , <code>cmdi</code> , <code>lfi</code> , <code>rce</code> , <code>nosqli</code> , <code>xxe</code>	A field legitimately contains SQL/HTML/shell-like text (search box, code snippet, WYSIWYG body).	"Not an attack?" for that specific path + rule; consider raising the threshold for that path only.
<code>upload-dangerous-ext</code> , <code>upload-double-ext</code> , <code>upload-path</code> , <code>upload-scan</code>	Legitimate upload type blocked.	Allow the extension in the upload policy; confirm it really is safe.
<code>hostname</code> , <code>protocol</code> , <code>proto-body-too-large</code>	Oversized but legitimate body, or unusual but valid protocol use.	Raise the body limit; verify the client.

C.3 Was it blocked or only observed?

Event action	What it means
<code>detect</code>	Not blocked — the site is in detect mode and this is a preview.
<code>block</code> / <code>deny</code> / <code>ban</code>	Enforced — the request was stopped.
<code>allow</code> with score <code>0.0</code>	A rule matched but didn't reach the block threshold (or is report-only).

CHAPTER D

D Bot check & browser challenge problems

D.1 A user is challenged over and over

A passed check grants a clearance cookie valid for `challenge.ttl` (default 12 h). Repeated challenges mean the clearance isn't matching the user on later requests. Match the pattern you see:

What you see	Cause	Fix
User on a mobile network or privacy relay (e.g. iCloud Private Relay) is re-challenged on every page	Clearance is bound to the client IP, which keeps changing; device tracking is off.	Turn on tracking.device_id — binds the clearance to a 1-year device cookie that survives IP changes.
Re-challenges or lost sessions only when several GAJWAF nodes sit behind a load balancer	Multi-node, keys not shared — a cookie issued by node A is not accepted by node B.	Give every node the same cluster.secret (with cluster.node set). The #1 cause of relay re-challenge — see H.1.
No challenge appears at all	The check only runs in block mode; detect never challenges.	Expected — not a fault.
No challenge on certain paths only	The path is in challenge.exempt_paths .	Expected.
User is challenged once , then browses normally	Working correctly — one check per session.	Nothing to do.

NOTE

If the pattern isn't clear, turn on **Bot check** → "**Debug logging**" (**challenge.debug**) for the site, reproduce the problem, note the exact time and the user's IP, and pass these to **GajShield Support**.

D.2 bot-biometrics Medium events everywhere

Informational only — a human solving a navigation check without pointer movement (very common on iCloud Private Relay). GAJWAF deliberately never bans for it on hosting/relay networks. Ignore unless you also see re-challenge churn (D.1).

CHAPTER E · AUTHGW-* EVENTS

E Authentication gateway & zero-trust

E.1 Symptoms and fixes

Symptom / event	Likely cause	Fix
Sign-in loops; <code>authgw-reverify</code> ; Microsoft re-login loop	Privacy relay inflates risk / changes TLS fingerprint each request; continuous device binding drops the session.	Ensure device binding uses the device cookie (current builds do); confirm <code>cluster.secret</code> is shared; relax <code>continuous.risk_step_up</code> if it forces re-auth.
"OIDC state mismatch (possible CSRF)"; lands on wrong page	Multiple concurrent sign-ins overwriting the flow cookie (fixed in current builds) — or clock skew.	Update to the current build; check server time/NTP.
Sign-in succeeds but every action says "session expired"	Session cookie too large (user in dozens of groups) — fixed by multi-cookie split in current builds; or the WAF session idled out before the app's.	Update; raise the gateway idle/session length to match the app.
<code>authgw-signin-failed</code> (single)	A normal failed login.	Benign unless repeated.
<code>authgw-bruteforce</code> (Critical, deny)	Failure limit tripped.	Expected protection; investigate the source IP.
<code>authgw-revoked</code>	Session ended by an admin "Sign out" or key rotation.	Expected; the user signs in again.
"needs the signing key" when enabling the gateway	Validation path lacked the key (fixed in current builds).	Update to the current build.

TIP

Use **Events** → **filter by user** to see a person's full sign-in and activity trail.

CHAPTER F

F Attacks are getting through

1. **Is the site in detect or off?** Detect logs but never blocks — switch to **block**. Check per site: a mixed estate often has some sites still in detect.
2. **Is the threshold too high?** The WAF blocks when a request's score reaches `anomaly_threshold`. One serious finding scores about 5. If attacks score below the threshold, lower it (or raise the security level).
3. **Check "Attacks observed but not blocked"** on the dashboard — that panel shows exactly what matched but wasn't enforced, with the reason.
4. **Rule out bypass.** The WAF only judges traffic that passes *through* it. If clients can reach the backend directly, it never sees those requests. Lock the backend to the WAF's IP.

G Data-leak prevention (DLP)

- DLP runs on **responses** and only masks data the app actually emits. If the app never returns a card/Aadhaar/PAN, there is nothing to mask — not a fault.
- Configuration lives under `response.dlp` — `credit_cards`, `stack_traces`, `sql_errors`, `private_keys`, `directory_listing`, `aadhaar`, `pan`, `us_ssn`, `custom` — each set to `off`, `mask` or `block`.
- Over-masking a benign field → turn off that specific rule or narrow the custom pattern.

IMPORTANT

Masking is enforcement — it does **not** fire in detect mode. If you expect masking and don't see it, confirm the site is in **block** mode.

CHAPTER H

H Operations

H.1 Multi-node / HA — the one setting that matters

Every node must have `cluster.node` set and the identical `cluster.secret`. All nodes then derive one signing key, so browser-check clearances, device cookies and login sessions stay valid as clients move between nodes.

IMPORTANT

A missing or mismatched secret causes relentless re-challenges and session loss for clients that aren't pinned to one node — worst on privacy relays.

- Keep each node's **site config in sync** via the console's export/import.
- The console is **not** load-balanced — reach each node directly on 9443.

H.2 Config editor problems

- A disabled site used to make the editor unusable (Export 500) — fixed in current builds; the disabled site now opens so you can fix it.
- Use **Validate** before **Save & apply**. The console's save validates strictly and refuses a broken site, telling you which field to fix.

H.3 Installed certificate disappeared after "Save & apply"

Fixed in current builds (the editor used to write a stale empty `tls.cert`). Re-install via the certificate panel's **Validate & install** and make sure the build is current.

H.4 Licensing

Edition	Site allowance
Unlicensed	1 free site
AWS paid Marketplace	Billed per instance by AWS — no site cap
BYOL / Azure	Per-site license

Install a license under **Settings → License**. If sites won't enable, check the license count and that the build trusts the production signing key.

H.5 Updates

Use **Settings → Software update** (signed, encrypted packages). No new image/AMI is needed for a version bump. The running version is shown at the bottom of the console.

H.6 Performance / high CPU

- A very low `challenge.difficulty` under a flood, or a very busy site — check the Overview rate-limit and challenge counters.
- `antivirus` scanning large uploads — check `av-oversize` / `av-scan-error`.
- Check the instance's CPU and memory graphs in your AWS or Azure console. GAJWAF is normally light; sustained high load with normal traffic is worth raising with **GajShield Support**.

APPENDIX

R Reference

R.1 Event action values

Action	Meaning	Blocked?
allow	Passed (clean, or a rule matched below threshold).	No
detect	Would have blocked, but the site is in detect mode.	No (logged)
block	A rule/score blocked the request.	Yes
deny	Access policy denied it (denylist, blocked country, IP-reputation feed, authz).	Yes
ratelimit	Too many requests from this client.	Yes (throttled)
ban	Client auto-banned (repeated attacks / automation).	Yes
manual-ban	Client banned by an administrator.	Yes
challenge	Browser check (proof-of-work) shown.	Held pending solve
challenge-pass	Browser check solved.	No (allowed)
response-block	A response was blocked/masked (DLP, defacement).	Response altered
upstream-error	The backend failed or was unreachable.	N/A (origin error)

R.2 Finding categories

Category	Example rule IDs	What it is
injection	sqli, xss, cmdi, lfi, rce, nosqli, xxe, ssrf*, open-redirect, prototype-pollution	Classic web-attack payloads.
recon / scanner	recon-sensitive-file, recon-*-mismatch, scanner-ua, scanner-header, scanner-oast	Probes, vulnerability scanners, sensitive-path hunting.
bot	bot-biometrics, bot-deception	Browser-check scoring and hidden trap-link hits.
tampering	hidden-field-tampered	A hidden form value changed after the server sent it.
api	api-unknown-path, api-schema, api-scope-denied, api-auth, api-rate-limit	OpenAPI contract / API auth / API limits.
auth	authgw-signin-ok/failed, authgw-bruteforce, authgw-reverify, authgw-revoked, authgw-risk, authz-denied, account-takeover, ato-*	Sign-in gateway and account-protection lifecycle.
geo / reputation	geo-blocked-country, geo-not-allowed-country, geo-tor, geo-datacenter, reputation-feed, anonymizer, hosting	Country, Tor, hosting/cloud, threat-intel feeds.

Category	Example rule IDs	What it is
data-leak	<code>dlp-credit-card</code> , <code>dlp-custom-*</code>	Sensitive data in a response.
client-side	<code>client-csp-violation</code>	Browser-reported script/CSP violations (card skimming).
upload / xml	<code>upload-*</code> , <code>xml-schema</code> , <code>xml-doctype</code> , <code>xml-limit</code> , <code>xxe</code>	File-upload and XML/SOAP validation.
ml	<code>ml-anomaly</code> , <code>ml-observe</code>	Machine-learning anomaly scoring.
ips / cve	<code>ips-*</code> , <code>cve-exploit</code> , <code>scan-patch-*</code>	Signature packs and virtual patches.
flow	<code>page-flow</code> , <code>flow</code>	Business-logic / page-sequence violations.

BEFORE YOU RAISE A TICKET

Contacting GajShield Support

Collect the following before contacting GajShield Support — it lets us resolve your ticket without a round of follow-up questions:

☐ 1. Version & edition	GAJWAF version (shown at the bottom of the console) and edition (AWS / Azure / BYOL).
☐ 2. Topology	One node or a cluster? If a cluster: is cluster.secret identical on all nodes? Include each node's configuration Export from the console (secrets redacted).
☐ 3. Site config	The affected site's mode and the relevant config block (challenge, auth, hidden_fields, dlp — whichever the ticket touches).
☐ 4. Event detail	Action, Finding rule ID, Incident ID, Target, score — or several events if intermittent.
☐ 5. Time of the problem	The exact date and time it happened (with time zone), so Support can match it in the logs.
☐ 6. Client symptom	Exact status code, screenshot, URL, and which user / IP / user-agent.
☐ 7. What changed	Config save, certificate renewal, release, node added — and when.

Related documents

Document	Covers
GAJWAF Quick Setup Guide — AWS & Azure	First-run setup for AWS/Azure: sign-in, adding a site, HTTPS certificates and switching to blocking.