

GAJSHIELD · WEB APPLICATION & API FIREWALL

GAJWAF

Features & Configuration Guide

Every option, in the same order as the console

Read it with the console open

Chapters follow the console tab by tab. Every option gives the **exact label** you see on screen, its **config key** — the setting's name in the configuration, for automation or review — and what it does. Label and key drive the same setting.

58 features

46 per-site · 12 global

177 options

Each with its console label

11 tabs

Plus Insights and Settings

About this document

Document	GAJWAF Features & Configuration Guide
Version	1.0 · October 2026
Applies to	GAJWAF — AWS Marketplace, Azure Marketplace and BYOL editions
Published by	GajShield Infotech Private Limited., Mumbai, India

COPYRIGHT

© 2026 GajShield Infotech Private Limited. 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 Private Limited.

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 Private Limited. 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 Private Limited. 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 GUIDE

Contents

PART 1 — PER-SITE FEATURES

1	General	6
1.1	Site identity & routing	6
1.2	Mode & blocking sensitivity	7
2	Sign-in	7
2.1	Require sign-in	7
2.2	Identity providers	8
2.3	Trusted devices	8
2.4	Continuous verification	8
2.5	Identity-risk security domain	8
2.6	Backend headers & session	9
2.7	Access rules (zero trust)	9
3	Backend & TLS	9
3.1	HTTPS certificate	9
3.2	HTTPS redirect & HSTS	10
3.3	Automatic certificates (Let's Encrypt)	10
3.4	Load balancing & persistence	10
3.5	Timeouts & health checks	10
3.6	Upstream TLS & source IP	11
4	Detection	11
4.1	Signature & injection detection	11
4.2	Application fingerprinting	11
4.3	Wrong-stack probe blocking	12
4.4	Upload malware scanning	12
4.5	Machine-learning detection	12
4.6	Traffic learning	12
4.7	Hidden-field tampering	12
4.8	Defacement monitoring	13
4.9	Account takeover protection	13
4.10	Page-flow rules	13
5	Geo & reputation	14
5.1	IP reputation & network type	14
5.2	Country policy	14
6	API	14
6.1	OpenAPI schema enforcement	14
6.2	API authentication	15
6.3	Identity-risk forwarding	15
6.4	XML / SOAP validation	15
7	Bot check	16
7.1	Browser check	16
7.2	Biometric bot detection	16
7.3	Device tracking	16
7.4	Client-side script protection	17
7.5	Bot deception	17
7.6	Under attack mode	17
8	Rate limits & bans	17
8.1	Rate limits (per client)	17
8.2	Automatic bans	18
9	Response & limits	18
9.1	Data-leak prevention (DLP)	18
9.2	Response headers & compression	19

9.3	Request limits	19
9.4	GraphQL	19
10	Exclusions	20
10.1	Rule exclusions	20
11	Custom rules	20
11.1	Custom match rules	20
12	Virtual patching	20
12.1	Scanner import & virtual patches	20
 PART 2 — GLOBAL SETTINGS		
13	Settings	21
13.1	Listeners	21
13.2	Management console	21
13.3	Network trust & lists	21
13.4	Alerts	22
13.5	Reports & compliance	22
13.6	Threat-intel IP feeds	22
13.7	GeoIP	22
13.8	Antivirus service	23
13.9	ACME	23
13.10	Software update	23
13.11	High-availability clustering	23
13.12	Licensing	24
A	Feature index (A–Z)	25

AT A GLANCE

Finding things in the console

Per-site options

Sites → (your site) → Edit configuration, then switch the toggle from Simple to All settings.

→ Part 1 · Chapters 1–11

Per-site insights

Read-only dashboards, plus scanner import and virtual patches: Sites → (your site) → Insights.

→ Chapter 12

Global options

Listeners, console, alerts, feeds, antivirus, updates, clustering and licensing: the Settings page.

→ Part 2 · Chapter 13

NOTE

Short path form. This guide writes the path as **Edit → All settings → <Tab> → "<label>"**. Each chapter opens with a navy **WHERE** bar showing its tab.

Security levels (the policy picker on the General tab) set many options at once. Pick one as your starting point, then fine-tune any option in this guide.

SITE MODES

General → "Mode" — every site runs in one of three

Mode	Behaviour	Use it when
block	Enforce — attacks are blocked, challenges and masking apply.	The site is tuned and live.
detect	Log only — never blocks or challenges.	Onboarding a new site or tuning false positives.
off	Pass through — no inspection.	Temporarily bypassing the WAF.

IMPORTANT

The browser check, rate limiting and DLP masking are **enforcement** — they act only in **block** mode. These entries are marked **BLOCK MODE ONLY**.

HOW TO READ AN ENTRY

Every feature follows the same layout

Column	What it tells you
Console label	The exact text next to the option on the tab.
Config key	The setting's name in the configuration, for automation or review. Dotted keys are nested (tls.acme = acme under tls).
Notes	Accepted values, defaults and caveats.

PART 1

Per-site features

Open **Sites** → (your site) → **Edit configuration**, then switch **Simple** to **All settings**. Chapters 1–11 follow the tabs from left to right.

1 General

WHERE Edit → All settings → **General**

Who the site is, where its traffic goes, and how strictly it blocks.

1.1 Site identity & routing

Which host names this site answers for, and where clean traffic goes.

CONSOLE LABEL	CONFIG KEY	NOTES
Name	<code>name</code>	Label used in logs and bans.
Host names	<code>hosts</code>	Exact names, a <code>*.example.com</code> wildcard, or <code>*</code> for everything else.
Serve only on these server IPs	<code>bind</code>	IP-based virtual hosting. Blank = all listen addresses.
Backend servers	<code>upstreams</code>	The web servers behind the WAF, used round-robin. Also on the Simple view as "Your web server".
Preserve Host header	<code>preserve_host</code>	
Allowed methods	<code>allowed_methods</code>	

1.2 Mode & blocking sensitivity

Whether the site enforces, and how much suspicion it takes to block a request.

CONSOLE LABEL	CONFIG KEY	NOTES
Mode	<code>mode</code>	<code>block</code> (enforce), <code>detect</code> (log only), or <code>off</code> (pass through).
Blocking sensitivity (score)	<code>anomaly_threshold</code>	Block when a request's suspicion score reaches this. One serious finding ≈ 5. Lower = stricter.
Block status code	<code>block_status</code>	HTTP code returned to blocked clients (default 403).

- **Security level** (the policy picker) and **HTTPS certificate** also appear on the **Simple** view for quick setup.

2 Sign-in

WHERE Edit → All settings → **Sign-in** — or use the "Set up zero-trust access" wizard there

An authentication gateway that puts a login — and zero-trust access rules — in front of your applications.

2.1 Require sign-in

Put a login in front of the app, or protect only some paths.

CONSOLE LABEL	CONFIG KEY	NOTES
Require sign-in	<code>auth_gateway.enabled</code>	
Check passwords with	<code>auth_gateway.provider</code>	<code>ldap</code> , <code>radius</code> , or <code>oidc</code> — see 2.2.
Sign-in page heading	<code>auth_gateway.title</code>	
Protected paths	<code>auth_gateway.paths</code>	
Public paths	<code>auth_gateway.exempt</code>	
Only these groups may sign in	<code>auth_gateway.allowed_groups</code>	

2.2 Identity providers

Connect the gateway to your directory or identity provider.

PROVIDER	CONFIG KEYS	CONSOLE FIELDS
LDAP / Active Directory	<code>auth_gateway.ldap.*</code>	Server, Service account DN/password or direct bind, Search base, User filter, Group attribute, StartTLS.
RADIUS	<code>auth_gateway.radius.*</code>	Server, Shared secret, NAS identifier, Timeout.
OIDC	<code>auth_gateway.oidc.*</code>	Issuer, Client ID/secret, Scopes, User/Groups claim, Redirect URL, Required auth levels (<code>acr_values</code>) for MFA.

2.3 Trusted devices

Enrol the browser as a trusted device on sign-in. Access rules can then require one.

CONSOLE LABEL	CONFIG KEY	NOTES
Remember trusted devices for	<code>auth_gateway.remember_device</code>	

2.4 Continuous verification

Re-check the session on every request, and force re-authentication on change, time or risk.

CONSOLE LABEL	CONFIG KEY	NOTES
Re-verify the session on every request	<code>auth_gateway.continuous.enabled</code>	
Drop the session if the country changes	<code>auth_gateway.continuous.bind_country</code>	
...if the device changes	<code>auth_gateway.continuous.bind_device</code>	
Force re-sign-in after	<code>auth_gateway.continuous.reauth</code>	
Re-authenticate when request risk reaches	<code>auth_gateway.continuous.step_up_score</code>	
...when identity risk reaches	<code>auth_gateway.continuous.risk_step_up</code>	

2.5 Identity-risk security domain

Correlate per-user risk across apps that share one directory.

CONSOLE LABEL	CONFIG KEY	NOTES
Identity-risk security domain	<code>auth_gateway.risk_domain</code>	Use the same value on all apps in one directory. Blank = isolated.

2.6 Backend headers & session

Pass the signed-in identity to the backend, and control session length.

CONSOLE LABEL	CONFIG KEY	NOTES
User header	<code>auth_gateway.user_header</code>	
Groups header	<code>auth_gateway.groups_header</code>	
Also send credentials (HTTP Basic)	<code>auth_gateway.forward_basic</code>	
Maximum session	<code>auth_gateway.session</code>	
Sign out after idle	<code>auth_gateway.idle</code>	
Signed SSO assertion	<code>auth_gateway.sso_secret</code>	Signs an identity assertion for the backend.

2.7 Access rules (zero trust)

Per-request Allow/Deny rules by group/user, path and method, with context — country, Tor, hosting, known device.

CONSOLE LABEL	CONFIG KEY	NOTES
Access rules (zero trust)	<code>auth_gateway.authorize</code>	
Refuse anything no Allow rule covers	<code>auth_gateway.default_deny</code>	Full deny-by-default.

3 Backend & TLS

WHERE Edit → All settings → **Backend & TLS**

Certificates served to clients, and how GAJWAF talks to your backend servers.

3.1 HTTPS certificate

The certificate served to clients.

CONSOLE LABEL	CONFIG KEY	NOTES
HTTPS certificate → Validate & install	<code>tls.cert</code> <code>tls.key</code>	Paste or upload the full-chain certificate and its key. Also on the Simple view.

3.2 HTTPS redirect & HSTS

Force clients onto HTTPS.

CONSOLE LABEL	CONFIG KEY	NOTES
Redirect HTTP to HTTPS	<code>tls.redirect_http</code>	
Send HSTS	<code>tls.hsts</code>	

IMPORTANT Turn on HSTS only once HTTPS works for **every** subdomain — browsers remember it.

3.3 Automatic certificates (Let's Encrypt)

Obtain and renew a certificate automatically for the site's host names.

CONSOLE LABEL	CONFIG KEY	NOTES
Get a free certificate automatically (Let's Encrypt)	<code>tls.acme</code>	Set the contact e-mail under Settings → ACME (13.9).
Use Let's Encrypt or an uploaded certificate — not both.		

3.4 Load balancing & persistence

Keep a client on the same backend when you have several.

CONSOLE LABEL	CONFIG KEY	NOTES
Keep each client on one server	<code>load_balancing.persistence</code>	<code>none</code> , <code>cookie</code> , or <code>source_ip</code> .
Cookie name	<code>load_balancing.cookie_name</code>	Used with cookie persistence.

3.5 Timeouts & health checks

How long to wait for the backend, and how to detect an unhealthy one.

CONSOLE LABEL	CONFIG KEY	NOTES
Backend response timeout	<code>timeouts.upstream</code>	
Health check path / interval	<code>health_check.path</code> <code>health_check.interval</code>	An empty path disables health checks.

3.6 Upstream TLS & source IP

Options for `https://` backends, and for passing the real client IP at the network level.

CONSOLE LABEL	CONFIG KEY	NOTES
Upstream TLS server name	<code>upstream_tls_server_name</code>	The SNI / certificate name to use when the backend is addressed by IP.
Skip backend certificate verification	<code>upstream_insecure_tls</code>	Trusted networks only.
Transparent source IP to backend (TPROXY)	<code>transparent_upstream</code>	The backend sees the real client IP as the TCP source. Needs network-level routing on the instance — contact GajShield Support before enabling.
Without TPROXY, the client IP is passed to the backend in <code>X-Forwarded-For</code>.		

4 Detection

WHERE Edit → All settings → **Detection**

The engines that find attacks in requests — signatures, machine learning, learned behaviour, and application-specific checks.

4.1 Signature & injection detection

The core engine — SQLi, XSS, command injection, LFI/RCE, SSRF, NoSQL, XXE, prototype pollution, open redirect, traversal and the IPS packs. **Always on**, tuned by the security level and blocking sensitivity (1.2).

CONSOLE LABEL	CONFIG KEY	NOTES
Disabled rules	<code>disabled_rules</code>	Rule IDs or globs silenced on this site.

4.2 Application fingerprinting

Detect the protected application's technology (to apply the right checks and virtual patches), and hide what it runs.

CONSOLE LABEL	CONFIG KEY	NOTES
Auto-detect the protected application	<code>fingerprint.auto</code>	
Actively check the backend	<code>fingerprint.active_probe</code> <code>fingerprint.probe_interval</code>	
Hide backend technology headers	<code>fingerprint.hide_backend_headers</code>	Strips headers such as <code>Server</code> and <code>X-Powered-By</code> .

4.3 Wrong-stack probe blocking

Block probes for technology this site doesn't run — e.g. `/wp-login.php` on a non-WordPress site.

CONSOLE LABEL	CONFIG KEY	NOTES
Block probes for technologies this site does not run	<code>block_wrong_stack</code>	

4.4 Upload malware scanning

Scan uploaded files and block malware before it reaches the backend.

CONSOLE LABEL	CONFIG KEY	NOTES
Scan uploaded files for malware (GajAV)	<code>antivirus</code>	Needs the antivirus service configured under Settings (13.8).

4.5 Machine-learning detection

A classifier flags request values that look like attacks, even with no signature match.

CONSOLE LABEL	CONFIG KEY	NOTES
Score values with the classifier	<code>ml_detection.enabled</code>	
Mode	<code>ml_detection.mode</code>	<code>monitor</code> or <code>enforce</code> .
Also log near-misses	<code>ml_detection.observe_from</code>	

4.6 Traffic learning

Learn each endpoint's normal parameters, types and lengths, then score deviations.

CONSOLE LABEL	CONFIG KEY	NOTES
Learn normal traffic	<code>learning.enabled</code>	
Score deviations	<code>learning.enforce</code>	Turn on once endpoints show "ready" under Insights.
Samples per endpoint	<code>learning.min_samples</code>	
Minimum learning period	<code>learning.min_period</code>	

4.7 Hidden-field tampering

Flag a submission that changes a hidden form value the server sent — price, ID or role tampering.

CONSOLE LABEL	CONFIG KEY	NOTES
Detect changed hidden fields	<code>hidden_fields.enabled</code>	
When a hidden field was changed	<code>hidden_fields.mode</code>	<code>report</code> or <code>enforce</code> .
Fields your JavaScript changes	<code>hidden_fields.exempt</code>	E.g. Matomo <code>module</code> / <code>action</code> .

4.8 Defacement monitoring

Fetch chosen pages on a schedule, and alert or restore if their visible text changes.

CONSOLE LABEL	CONFIG KEY	NOTES
Watch pages for defacement	<code>defacement.enabled</code>	
Pages to watch	<code>defacement.pages</code>	
When a page is defaced	<code>defacement.action</code>	<code>alert</code> or <code>restore</code> .
Check every	<code>defacement.interval</code>	
Text change that counts (%)	<code>defacement.threshold</code>	
Extra defacement phrases	<code>defacement.keywords</code>	

4.9 Account takeover protection

Detect credential stuffing, password spraying and attacks on a single account. Passwords are never stored.

CONSOLE LABEL	CONFIG KEY	NOTES
Protect sign-ins	<code>account_protection.enabled</code>	
Ban a client after failures on N accounts	<code>max_accounts_per_client</code>	
Protect an account after N failures	<code>max_failures_per_account</code>	
Protect for	<code>protect_for</code>	
Block a password tried on N accounts	<code>spray_accounts</code>	
Extra username field names	<code>username_fields</code>	

4.10 Page-flow rules

Enforce that multi-step flows are followed in order.

CONSOLE LABEL	CONFIG KEY	NOTES
Page-flow rules	<code>page_flows</code>	

5 Geo & reputation

WHERE Edit → All settings → **Geo & reputation**

Decide who gets in based on where a request comes from.

5.1 IP reputation & network type

Block — or add suspicion to — known-bad IPs, Tor exit nodes and hosting/cloud networks.

CONSOLE LABEL	CONFIG KEY	NOTES
Use known-bad IP lists	<code>reputation</code>	Feeds are configured under Settings (13.6).
Block Tor exit nodes	<code>geo.block_tor</code>	
Add suspicion to Tor	<code>geo.score_tor</code>	
Block hosting/cloud networks	<code>geo.block_datacenter</code>	
Add suspicion (hosting/cloud)	<code>geo.score_datacenter</code>	

5.2 Country policy

Block countries, or allow only a list of countries.

CONSOLE LABEL	CONFIG KEY	NOTES
Country block list	<code>geo.block_countries</code>	
Country allow list	<code>geo.allow_countries</code>	Only these countries may connect.

6 API

WHERE Edit → All settings → **API**

Contract enforcement and authentication for APIs and web services.

6.1 OpenAPI schema enforcement

Validate requests against your OpenAPI spec and block off-contract calls.

CONSOLE LABEL	CONFIG KEY	NOTES
Mode	<code>api_schema.mode</code>	<code>off</code> , <code>report</code> , or <code>enforce</code> .
Check requests under	<code>api_schema.base_path</code>	
Specification file	<code>api_schema.file</code>	
Upload the spec here, or from the site page: API protection → Upload OpenAPI spec.		

6.2 API authentication

Verify a token or API key at the WAF and identify the caller — for events, per-identity limits and risk.

CONSOLE LABEL	CONFIG KEY	NOTES
Check credentials on	<code>api_auth.paths</code>	
Require credentials	<code>api_auth.require</code>	
JWT settings	<code>api_auth.jwt.*</code>	JWKS URL or shared secret, algorithms, issuers, audiences, user claim.
Per-identity limits	<code>api_auth.rate_limit.rps</code> <code>api_auth.rate_limit.burst</code>	
Scope rules	<code>api_auth.scope_rules</code>	

6.3 Identity-risk forwarding

Send the WAF's per-user risk score to the backend, so an app with its own login can step up.

CONSOLE LABEL	CONFIG KEY	NOTES
Send the identity-risk score to the backend	<code>identity_risk.enabled</code>	
Score header	<code>identity_risk.header</code>	
Reasons header	<code>identity_risk.reasons_header</code>	
Minimum score	<code>identity_risk.min</code>	

6.4 XML / SOAP validation

Validate XML bodies under a path against an XSD or WSDL.

CONSOLE LABEL	CONFIG KEY	NOTES
XML schemas editor	<code>xml_schemas</code>	Each entry: path, XSD/WSDL file, mode.

7 Bot check

WHERE Edit → All settings → Bot check

Separate real visitors from scripts and bots — and defend pages against card-skimming scripts.

7.1 Browser check

A JavaScript proof-of-work puzzle that stops scripted floods and headless bots. Real visitors wait about one second, once.

CONSOLE LABEL	CONFIG KEY	NOTES
Check browsers	<code>challenge.mode</code>	<code>off</code> , <code>suspicious</code> , or <code>always</code> .
Difficulty	<code>challenge.difficulty</code>	
Pass valid for	<code>challenge.ttl</code>	Default 12 h .
Clients that cannot run the check	<code>challenge.non_browser</code>	
Never check these paths	<code>challenge.exempt_paths</code>	
Debug logging	<code>challenge.debug</code>	Verbose diagnostics — turn on only when GajShield Support asks, then turn off.
BLOCK MODE ONLY Enforcement — this does not act while the site is in detect mode.		

7.2 Biometric bot detection

Score pointer, keyboard, touch and scroll activity during the check.

CONSOLE LABEL	CONFIG KEY	NOTES
Score human interaction during the check	<code>challenge.biometrics</code>	

7.3 Device tracking

A signed `waf_did` cookie that follows a browser across IP changes, so events, bans and passes survive privacy relays.

CONSOLE LABEL	CONFIG KEY	NOTES
Identify devices with a WAF cookie	<code>tracking.device_id</code>	

7.4 Client-side script protection

Card-skimming (Magecart) defence: inventory the scripts your pages load, alert on new or changed ones, and enforce an approved-origin policy.

CONSOLE LABEL	CONFIG KEY	NOTES
Client-side script protection	<code>client_side.enabled</code>	
Script policy	<code>client_side.mode</code>	<code>monitor</code> or <code>enforce</code> .
Approved script origins	<code>client_side.allowed_sources</code>	Use " Approve source " on an event to add one.
Learn silently for	<code>client_side.learn_for</code>	

7.5 Bot deception

A hidden trap link real users never see — clients that follow it are bots.

CONSOLE LABEL	CONFIG KEY	NOTES
Insert a hidden trap link	<code>bot_deception.enabled</code>	
Ban clients that follow it	<code>bot_deception.ban</code>	

7.6 Under attack mode

Challenge every visitor with a harder puzzle — switched on automatically under a flood.

CONSOLE LABEL	CONFIG KEY	NOTES
Switch on automatically at	<code>challenge.auto_threshold</code>	
Keep it on for	<code>challenge.auto_duration</code>	

8 Rate limits & bans

WHERE Edit → All settings → **Rate limits & bans**

Cap how fast clients can go, and remove repeat offenders.

8.1 Rate limits (per client)

Cap requests per client, with a tighter cap on login attempts.

CONSOLE LABEL	CONFIG KEY	NOTES
Requests per second	<code>rate_limit.rps</code>	
Burst	<code>rate_limit.burst</code>	
Login attempts per second	<code>rate_limit.auth_rps</code>	
Login burst	<code>rate_limit.auth_burst</code>	
BLOCK MODE ONLY Enforcement — this does not act while the site is in detect mode.		

8.2 Automatic bans

Ban a client that trips too many blocks, 404s, failed logins or rate-limit violations.

CONSOLE LABEL	CONFIG KEY	NOTES
Ban duration	<code>behavior.ban_duration</code>	
Blocked requests/minute before ban	<code>behavior.max_blocked_per_minute</code>	
404s/minute	<code>behavior.max_404_per_minute</code>	
Failed logins/10 min	<code>behavior.max_auth_failures</code>	
Rate-limit violations/minute	<code>behavior.max_rate_violations</code>	
Manual bans are managed on the Bans page.		

9 Response & limits

WHERE Edit → All settings → Response & limits

Protect what your application sends back, and cap what clients can send in.

9.1 Data-leak prevention (DLP)

Mask or block sensitive data in responses. Each type is set to **off**, **mask** or **block**.

CONSOLE LABEL	CONFIG KEY	NOTES
Payment card numbers	<code>response.dlp.credit_cards</code>	
Stack traces / debug pages	<code>response.dlp.stack_traces</code>	
Database errors	<code>response.dlp.sql_errors</code>	
Private keys & cloud credentials	<code>response.dlp.private_keys</code>	
Directory listings	<code>response.dlp.directory_listing</code>	
Aadhaar	<code>response.dlp.aadhaar</code>	
PAN	<code>response.dlp.pan</code>	
US SSNs	<code>response.dlp.us_ssn</code>	
Custom patterns	<code>response.dlp.custom</code>	
BLOCK MODE ONLY Enforcement — this does not act while the site is in detect mode.		

9.2 Response headers & compression

Add protective headers and a Content-Security-Policy to responses.

CONSOLE LABEL	CONFIG KEY	NOTES
Add security headers	<code>response.security_headers</code>	
Compress responses (gzip)	<code>response.compress</code>	
Content-Security-Policy	<code>response.content_security_policy</code>	
Max response bytes inspected	<code>response.max_inspect_bytes</code>	

9.3 Request limits

Hard caps that reject abusive or oversized requests.

CONSOLE LABEL	CONFIG KEY	NOTES
Max body bytes	<code>limits.max_body_bytes</code>	
Body bytes inspected	<code>limits.inspect_body_bytes</code>	
Max arguments	<code>limits.max_args</code>	
Max argument length	<code>limits.max_arg_length</code>	
Max JSON depth	<code>limits.max_json_depth</code>	
Max headers	<code>limits.max_headers</code>	
Max URI length	<code>limits.max_uri_length</code>	
XML caps	<code>limits.xml_max_depth</code> <code>limits.xml_max_elements</code> <code>limits.xml_max_attributes</code>	

9.4 GraphQL

Limit GraphQL abuse.

CONSOLE LABEL	CONFIG KEY	NOTES
Block introspection	<code>graphql.block_introspection</code>	
Max depth	<code>graphql.max_depth</code>	
Max aliases	<code>graphql.max_aliases</code>	
Max batch size	<code>graphql.max_batch</code>	

10 Exclusions

WHERE Edit → All settings → **Exclusions**

Fine-tune detection for your application.

10.1 Rule exclusions

Stop a specific rule firing on a specific path.

CONSOLE LABEL	CONFIG KEY	NOTES
Exclusions	<code>exclusions</code>	The "Not an attack?" button on an event creates one for you.

11 Custom rules

WHERE Edit → All settings → **Custom rules**

Your own match rules, alongside the built-in engine.

11.1 Custom match rules

Add your own match rules.

CONSOLE LABEL	CONFIG KEY	NOTES
Custom rules	<code>rules</code>	

12 Virtual patching

WHERE Sites → (your site) → **Insights** → "Scanner findings and virtual patches"

Found on the site's Insights page — not in the configuration editor.

12.1 Scanner import & virtual patches

Protect vulnerable pages without changing the application.

1. Open **Sites** → (your site) → **Insights**.
2. Go to "**Scanner findings and virtual patches**" → **Import scanner report**.
3. Upload a report from **OWASP ZAP**, **Burp**, **Nessus**, **OpenVAS** or a **CSV**.

CONSOLE LABEL	CONFIG KEY	NOTES
Scanner findings and virtual patches	<code>scan_patches</code>	Each injectable page/parameter in the report gets its own virtual patch.
• CVE virtual patches for the detected stack (see 4.2) are applied automatically.		

PART 2

Global settings

Options that apply to the whole GAJWAF instance — the **Settings** page (admin role).

13 Settings

WHERE **Settings** (admin role)

Instance-wide options. They apply to every site.

13.1 Listeners

The ports GAJWAF serves web traffic on.

CONSOLE LABEL	CONFIG KEY	NOTES
HTTP	<code>listen.http</code>	
HTTPS	<code>listen.https</code>	

13.2 Management console

The admin console and how it is protected.

CONSOLE LABEL	CONFIG KEY	NOTES
Console users	<code>admin.*</code>	One account per operator.
Require login · Two-factor authentication	<code>admin.require_login</code>	
Console listen address	<code>admin.listen</code>	Default <code>0.0.0.0:9443</code> .
Console token	<code>admin.token</code>	

13.3 Network trust & lists

Which upstream proxies to trust, and global IP allow/deny lists.

CONSOLE LABEL	CONFIG KEY	NOTES
Trusted proxies	<code>trusted_proxies</code>	Trust <code>X-Forwarded-For</code> only from these.
Allow list	<code>allowlist</code>	
Deny list	<code>denylist</code>	

13.4 Alerts

Notify on attacks, spikes, outages and certificate expiry.

CONSOLE LABEL	CONFIG KEY	NOTES
Email	<code>alerts.email.*</code>	SMTP server, encryption, user, password, from, to, alert types.
SNMP	<code>alerts.snmp.*</code>	
Syslog	<code>alerts.syslog.*</code>	With <code>min_score</code> .
Attack spike	<code>alerts.attack_spike_per_minute</code>	
Cooldown	<code>alerts.cooldown</code>	
Warn before certificate expiry	<code>alerts.cert_expiry_days</code>	
Webhooks (e.g. Slack) are also supported.		

13.5 Reports & compliance

Scheduled PDF, HTML and CSV reports, including compliance reports.

CONSOLE LABEL	CONFIG KEY	NOTES
Reports	<code>reports</code>	

13.6 Threat-intel IP feeds

The known-bad IP lists that sites enable with Geo & reputation → "Use known-bad IP lists" (5.1).

CONSOLE LABEL	CONFIG KEY	NOTES
Threat-intel feeds	<code>reputation</code> <code>reputation.refresh</code>	

13.7 GeoIP

Country/ASN data for geo policy and the attack map.

CONSOLE LABEL	CONFIG KEY	NOTES
Download free DB-IP Lite databases automatically	<code>geoip.auto_download</code>	Or upload your own <code>.mmdb</code> file.

13.8 Antivirus service

The malware scanner used by upload scanning (4.4).

CONSOLE LABEL	CONFIG KEY	NOTES
Daemon address	<code>antivirus.address</code>	
Scan timeout	<code>antivirus.timeout</code>	
If the antivirus is unavailable	<code>antivirus.on_error</code>	
Files over the size limit	<code>antivirus.on_oversize</code>	
Also scan raw file uploads	<code>antivirus.scan_raw_uploads</code>	
Log every scanned file	<code>antivirus.log_clean_scans</code>	

13.9 ACME

Contact details for Let's Encrypt certificates (3.3).

CONSOLE LABEL	CONFIG KEY	NOTES
Contact e-mail	<code>acme.email</code>	
Directory URL	<code>acme.directory_url</code>	

13.10 Software update

Signed, encrypted updates applied from the console.

CONSOLE LABEL	CONFIG KEY	NOTES
Software update	<code>updates.url</code> <code>updates.interval</code>	

13.11 High-availability clustering

Share the signing key across nodes, so browser-check passes, device cookies and login sessions stay valid as clients move between nodes.

CONSOLE LABEL	CONFIG KEY	NOTES
Node name	<code>cluster.node</code>	Set on every node.
Cluster secret	<code>cluster.secret</code>	Identical on every node.
IMPORTANT A missing or mismatched cluster secret causes repeated re-challenges and session loss.		

13.12 Licensing

Unlicensed = 1 free site. A signed, node-locked license raises the limit.

CONSOLE LABEL	CONFIG KEY	NOTES
Settings → License	—	Install the license file here.
• AWS paid Marketplace: billed per instance — no site cap. • BYOL / Azure: per-site license.		

APPENDIX

A Feature index (A–Z)

A		I	
Access rules (zero trust).....	9	IP reputation & network type.....	14
Account takeover protection	13	L	
ACME.....	23	Licensing	24
Alerts	22	Listeners.....	21
Antivirus service.....	23	Load balancing & persistence	10
API authentication.....	15	M	
Application fingerprinting	11	Machine-learning detection	12
Automatic bans	18	Management console	21
Automatic certificates (Let's Encrypt).....	10	Mode & blocking sensitivity	7
B		N	
Backend headers & session.....	9	Network trust & lists.....	21
Biometric bot detection.....	16	O	
Bot deception	17	OpenAPI schema enforcement.....	14
Browser check.....	16	P	
C		Page-flow rules.....	13
Client-side script protection.....	17	R	
Continuous verification	8	Rate limits (per client).....	17
Country policy	14	Reports & compliance.....	22
Custom match rules	20	Request limits.....	19
D		Require sign-in.....	7
Data-leak prevention (DLP)	18	Response headers & compression	19
Defacement monitoring	13	Rule exclusions	20
Device tracking.....	16	S	
G		Scanner import & virtual patches	20
GeoIP.....	22	Signature & injection detection	11
GraphQL	19	Site identity & routing.....	6
H		Software update	23
Hidden-field tampering	12	T	
High-availability clustering.....	23	Threat-intel IP feeds	22
HTTPS certificate	9	Timeouts & health checks	10
HTTPS redirect & HSTS.....	10	Traffic learning	12
I		Trusted devices.....	8
Identity providers.....	8	U	
Identity-risk forwarding	15	Under attack mode	17
Identity-risk security domain	8	Upload malware scanning	12
		Upstream TLS & source IP	11
		W	
		Wrong-stack probe blocking.....	12
		X	
		XML / SOAP validation	15

Related documents

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

GAJWAF Troubleshooting & Support Manual — Symptom-first fixes, event and finding references, and contacting Support.