

GAJWAF · PRE-SALES PLAYBOOK

Running a POC / evaluation

Prove GAJWAF's value on the prospect's own application

Detect first. Enforce with consent.

Prove GAJWAF's value against a prospect's own application — **safely**, and without letting a misconfigured test turn into a false *"it can be bypassed"* claim.

0 risk

To their live app in Detect

6 classes

Of attack to demonstrate

1 sheet

Scope, results & sign-off

GajShield Infotech Private Limited
Mumbai, India

Version 1.0 · October 2026
For GajShield solutions engineers & partners

LEGAL NOTICE

© 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

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. This playbook is general operational guidance. It does not replace your organisation's own incident-response or change-control procedures, and GAJWAF is not a substitute for network-layer (L3/L4) DDoS protection.

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.

AT A GLANCE

The evaluation

GOLDEN RULE

Demonstrate value in **Detect** first — show what GAJWAF **would** block against their real application, with no risk to it.

Then enforce **only with the prospect's agreement**.



✗ DON'T

- **Don't** switch a prospect's live site to Block without agreement and a false-positive review.
- **Don't** let an unscoped "bypass" claim stand without checking for misconfiguration.
- **Don't** invent metrics or customer names to make the case — the live demo is the evidence.

WHO **Solutions engineer** (or partner), with the **prospect's admin**.

CONTENTS

1	Agree scope and success criteria up front.....	3
2	Stand GAJWAF up in Detect	3
3	Show it working with real requests	4
4	Show the operator experience	4
5	Enforce — only with their OK.....	5
!	Bake-off & bypass claims	5
✓	POC scope & sign-off sheet	6

STEP 1 OF 5

1 Agree scope and success criteria up front

PHASE **PLAN** WHO Solutions engineer + prospect's admin

Before touching anything, write down with the prospect — use the **POC scope & sign-off sheet** at the back of this playbook:

Agree	Examples
Which site / application is in scope, and its upstream (backend)	Their customer portal, https://portal.example.com → 10.0.1.20
Which attack classes they want to see stopped	OWASP Top 10, bots, DoS, DLP, API abuse...
Duration	A fixed number of days or a set of sessions
What "success" means to them	Written, measurable criteria — so the outcome isn't argued afterwards

✓ MOVE ON WHEN

Scope, duration and success criteria are written down and agreed by both sides.

STEP 2 OF 5

2 Stand GAJWAF up in Detect

PHASE **DEMONSTRATE** SITE MODE **Detect** WHO Solutions engineer + prospect's admin

Put GAJWAF in front of their site — or a **staging copy** — in **Detect** mode. It inspects and logs everything and **blocks nothing**: zero risk to their live application.

1. Add the site with its host names and upstream (*Quick Setup Guide*, steps 3–4).
2. Leave it in **Detect**.
3. Confirm the **health chips** on the Sites page are **green**.

✓ MOVE ON WHEN

Their traffic flows through GAJWAF, health chips are green, and nothing is being blocked.

STEP 3 OF 5

3 Show it working with real requests

PHASE **DEMONSTRATE** SITE MODE **Detect** WHO Solutions engineer + prospect's admin

Walk through the **GAJWAF demo playbook** against the site — it has the exact requests and a **suggested 15-minute flow** — and show the **Events** that light up:

☐ **Injection**
SQLi, XSS, path traversal, command injection

☐ **API abuse**
Schema violations — shadow endpoints, bad methods / types

☐ **Bots & automation**
Bot detection and the browser check

☐ **Floods**
Rate limiting and under-attack behaviour

☐ **Data leaks**
Response data-leak prevention

☐ **Reconnaissance**
Scanner and recon fingerprinting

TIP
The console's "**Attacks observed but not blocked**" view is the money shot in Detect: everything there is what GAJWAF **would** block in production.

✓ **MOVE ON WHEN**
Every attack class in scope has been shown on **their own application**.

STEP 4 OF 5

4 Show the operator experience

PHASE **DEMONSTRATE** SITE MODE **Detect** WHO Solutions engineer + prospect's admin

Show	Why it matters
The Overview dashboard and per-site Insights	What their team will look at every day.
A compliance report (<i>Compliance reporting</i> playbook)	The evidence they'll hand an auditor.
How a false positive is handled cleanly (<i>Taming false positives</i> playbook)	Honesty about tuning builds trust.

✓ **MOVE ON WHEN**
The prospect has seen what running GAJWAF day-to-day looks like.

STEP 5 OF 5

5 Enforce — only with their OK

PHASE **ENFORCE** SITE MODE **Block (with consent)** WHO Solutions engineer + prospect's admin

If the prospect wants to see **live blocking**:

1. Get their **explicit agreement** — record it on the sign-off sheet.
2. **Clear any false positives first.**
3. Switch the in-scope site to **Block** (*Onboard a site safely* playbook).

IMPORTANT

Never switch a prospect's live site to Block without their agreement and a false-positive review.

✓ MOVE ON WHEN

Any enforcement happened with consent, after false-positive tuning.

IF A COMPETITOR OR PROSPECT TESTS GAJWAF

! Bake-off & bypass claims

If a prospect — or a competing vendor — claims they "**bypassed**" the WAF, check it isn't a **misconfiguration** before accepting it:

Check	If yes, it's...	Fix
Was the site in Detect , not Block?	Expected — Detect never blocks.	Re-test in Block .
Was the tested path exempt (bot-check exempt path or an exclusion)?	Configured to pass — not a bypass.	Re-test on a path that isn't exempt.
Did the request get a 421 — the host not matched ?	The request never reached a protected site.	Re-test against the configured host name.
Did the test hit the origin directly , around the WAF?	A hardening gap , not a WAF failure.	Lock the origin to the WAF (<i>Hardening the deployment</i>).

A FAIR BAKE-OFF TESTS GAJWAF...**In Block**

Not Detect — so findings are actually enforced.

Inline

Every request goes **through** GAJWAF; the origin is not reachable directly.

For the configured host

Tests target the host names the site is set up for.

IMPORTANT

Make this the **agreed setup** before testing starts — so a misconfigured test can't masquerade as a bypass.

FILL IN WITH THE PROSPECT

✓ POC scope & sign-off sheet

SCOPE — AGREED BEFORE STARTING

Prospect / contact	
Site / application	
Upstream (backend)	
Environment	☐ Production (Detect only) ☐ Staging copy
Duration	
Success criteria	

RESULTS — PER ATTACK CLASS IN SCOPE

Attack class	Observed	Blocked	Notes / event IDs
Injection (OWASP)	☐	☐	
API abuse	☐	☐	
Bots & automation	☐	☐	
Floods / DoS	☐	☐	
Data leaks (DLP)	☐	☐	
Reconnaissance	☐	☐	

ENFORCEMENT CONSENT & OUTCOME

Block mode agreed by	☐ Not requested ☐ Agreed — name: _____ date: _____
False positives cleared before Block	☐ Yes
Success criteria met	☐ Yes ☐ Partly ☐ No
Signed — GajShield / partner	
Signed — prospect	