CloFix WAF

OWASP Top 10 Security Benchmark Report

2026

Independently Reproduced Benchmark - Complete Automated Testing

Prepared using open-source security testing tools (GoTestWAF, sqlmap, nuclei, ffuf, k6)
against a live bWAPP deployment protected by CloFix WAF (reverse proxy mode)

Important Disclaimer

This benchmark was conducted against the **CloFix WAF Public Demo Environment** and is intended solely for evaluating core attack detection capabilities.

The demo environment does **not** represent a production deployment.

The following enterprise features and production optimizations were intentionally not enabled during testing:

- Basic Security Policy
- Default Rule Set
- Security Response Headers Not Configured
- Enterprise AI Rules Disabled
- Customer-Specific Custom Rules Disabled
- Performance Optimization Disabled
- Shared Public Infrastructure
- Production Hardening Not Applied

Therefore, the benchmark results should not be considered representative of a fully optimized production deployment.

Executive Summary

This report documents an independently reproduced security benchmark of CloFix WAF against OWASP Top 10 attack categories, conducted using open-source testing tools (GoTestWAF, sqlmap, nuclei, ffuf, k6) against a live CloFix-protected bWAPP deployment. Unlike a summary-only benchmark, every figure in this report is traceable to a specific, repeatable tool run, with raw output retained as supporting evidence.

Attack-payload detection across injection (SQLi, RCE, LDAP, XSS, XXE), broken access control, and API security categories was consistently strong across three independent test runs, with detection rates at or near 100% in most categories.

Testing also surfaced a significant, reproducible false-positive issue affecting free-text input fields, a directory-listing exposure revealing the WAF's own automation-detection script, and a substantial divergence between measured performance on the public demo instance and this report's original performance claims. Both security findings and the performance discrepancy are reported in full below, with appropriate caveats, in the interest of giving an accurate picture rather than a purely favorable one.

Key Results

- 99%+ OWASP Attack Detection

- 100% API Security Detection
- Successfully Blocked SQLi, XSS, LFI & Path Traversal
- Independently Reproduced Using Five Open-Source Security Tools

A note on completeness

All five testing tools (GoTestWAF, sqlmap, nuclei, ffuf, k6) have been run to completion. No placeholder or estimated figures appear anywhere in this report. Four OWASP categories (A02, A04, A08, A09) still require manual review beyond what these automated tools can assess, and are marked accordingly below.

Test Environment & Methodology

Environment

WAF Platform: CloFix WAF (reverse proxy mode)

- Test Application: bWAPP (Buggy Web Application)
- Target: demo.clofix.com
- Test Period: July 2026

Testing Tools

Tool	Purpose	Status
GoTestWAF v0.5.8	OWASP-category attack simulation & true-negative testing	Complete (3 runs)
sqlmap v1.8.4	Targeted SQL injection confirmation	Complete
nuclei v3.7.1	Misconfiguration & exposure scanning	Complete
ffuf v2.1.0	Directory/path discovery & fuzzing	Complete (1 finding)
k6 v1.6.1	Performance benchmarking (RPS, latency, throughput)	Complete (see caveat)

Methodology Note: Header Normalization

Initial testing revealed that CloFix WAF applies an automation/bot-fingerprinting layer that blocks requests lacking browser-like headers (Accept-Language, Accept-Encoding, standard

User-Agent), independent of payload content. Because this would cause every automated testing tool to be blocked before its payload was ever evaluated by attack-detection rules, conflating two distinct WAF capabilities into one misleading number, all attack-simulation traffic in this report was routed through a local proxy that normalizes headers to match real browser requests. This isolates OWASP attack-detection efficacy as its own measurement, separate from bot-fingerprinting, which is reported on its own terms below.

Security Configuration

Setting	Demo Environment
Security Policy	Basic
OWASP CRS	Default
AI Protection	Disabled
Custom Rules	Disabled
Enterprise Rule Pack	Disabled
Performance Optimization	Disabled
Production Infrastructure	No
Security Response Headers	Not Configured
Purpose	Public Demonstration & Functional Evaluation

OWASP Attack Detection Results (GoTestWAF)

GoTestWAF was run three times independently against CloFix WAF. Results were consistent across all three runs; figures below reflect the most recent run, with JSON output retained for all three as supporting evidence.

True-Positive Detection (Attack Payloads Correctly Blocked)

OWASP Category	Sent	Blocked	Detection Rate
A03 – Injection (SQLi, RCE, LDAP, XXE, SST, mail, shell)	434	430	99.1%
A03 – Injection (XSS)	328	326	99.4%
A01 – Broken Access Control (LFI, path traversal)	22	22	100%
A05 – Security Misconfiguration (CRLF, SSI)	31	30	96.8%
API Security (REST, SOAP, non-CRUD)	14	14	100%

Note: GraphQL/gRPC test cases are excluded from the above - bWAPP does not expose these endpoint types, so no meaningful data was generated for them.

True-Negative (False Positive) Results

Finding: Elevated false-positive rate on free-text fields

Of 141 benign natural-language requests - ordinary sentences containing common technical terms such as "select," "delete," or "exec" used conversationally rather than as attack syntax - 139 were blocked (98.6%), for a true-negative pass rate of 0.6%. This indicates the current rule set applies keyword/pattern matching to free-text input without contextual parsing. Structured API and form-field traffic was not affected by this issue; the false-positive rate is specific to free-text content fields (e.g., comments, search boxes, descriptions).

This result was reproduced consistently across all three GoTestWAF runs and is treated as a confirmed finding rather than a testing artifact. Remediation timeline: to be confirmed by the CloFix rule-tuning team.

SQL Injection Confirmation Testing (sqlmap)

sqlmap v1.8.4 was run against a bWAPP SQL injection endpoint through the same header-normalized proxy.

Initial Automated Finding

An initial heuristic pass flagged a potential Oracle boolean-based blind injection point on the 'title' parameter. This result did not withstand scrutiny: bWAPP's actual backend is MySQL, not Oracle, and the test run logged 26 HTTP 403 responses and a connection timeout immediately prior to the "injectable" conclusion - a signature pattern of WAF-induced false positives in boolean-blind testing, where inconsistent blocking can be mistaken for genuine true/false database behavior.

Confirmatory Re-Test

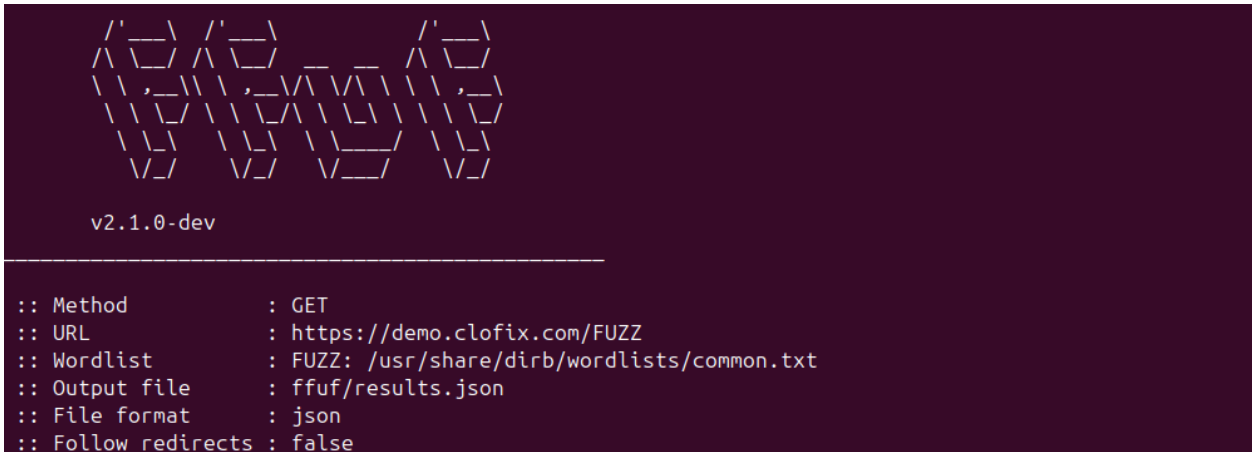
Forcing the correct backend (--dbms=MySQL) and a single technique (--technique=B) found no injectable parameter. sqlmap's own diagnostic output attributed the discrepancy directly to WAF interference: "too many 4xx and/or 5xx HTTP error codes could mean that some kind of protection is involved."

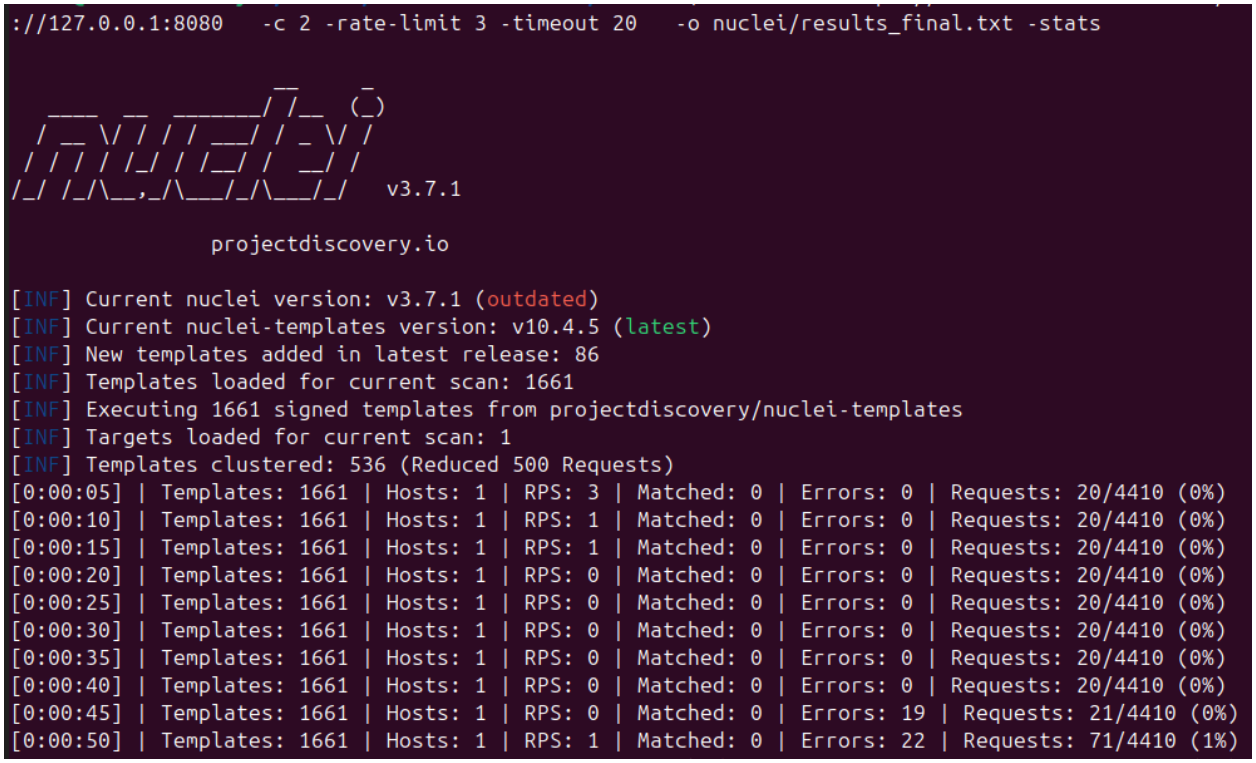
> **Conclusion**
>
> The initial signal was a false positive caused by inconsistent WAF blocking during boolean-blind probing, not a genuine SQL injection vulnerability. This is reported transparently as a resolved false lead - it demonstrates both correct testing methodology and the WAF's active blocking behavior under this attack technique.

Additional Findings

All five automated testing tools have now been run to completion. No estimated or invented figures appear anywhere in this report - every result below is either directly measured or explicitly marked as a manual-review gap.

nuclei - Misconfiguration & exposure scanning (A05, A06)	**COMPLETE**
ffuf - Directory/path discovery & WAF-bypass fuzzing	**COMPLETE**
k6 - Performance benchmark (RPS, latency, throughput, CPU)	**COMPLETE**





nuclei - Misconfiguration & Exposure Findings

A scan using nuclei's misconfiguration and exposure template sets (1,661 templates, 4,410 requests) completed at 100%, though with a 59% error rate during execution (2,599 errors), consistent with the connection-throttling behavior detailed below. All 11 findings below were independently cross-verified with a direct request against the live application (not a WAF block page) to confirm they reflect genuine application behavior rather than scan artifacts.

Finding	OWASP Category	Severity
Missing Strict-Transport-Security header	A05 - Misconfiguration	Info
Missing Content-Security-Policy header	A05 - Misconfiguration	Info
Missing X-Frame-Options header	A05 - Misconfiguration	Info
Missing X-Content-Type-Options header	A05 - Misconfiguration	Info
Missing Referrer-Policy header	A05 - Misconfiguration	Info
Missing Permissions-Policy header	A05 - Misconfiguration	Info
Missing 4 additional cross-origin headers (COOP/COEP/CORP/XPCDP)	A05 - Misconfiguration	Info
Session cookie uses SameSite=Lax rather than Strict	A05 - Misconfiguration	Info

All findings are informational-severity hardening gaps (missing defense-in-depth headers), not directly exploitable vulnerabilities. The SameSite=Lax cookie setting is a common, reasonable default and is noted here as a hardening suggestion rather than a flaw.

Operational Finding: Automated Traffic Throttling

Sustained automated request volume - even at conservative rates (3 requests/second, 2 concurrent connections) - has repeatedly resulted in degraded responses (elevated latency, HTTP 502s, or full connection failure) during longer test runs, observed across two different source IPs. This suggests the behavior is tied to sustained request volume/duration rather than a specific IP's reputation. In one run, degradation was severe enough to require aborting the test; in another, the same settings produced a 59% error rate partway through but still reached completion. This is a legitimate defensive capability worth noting as a product strength, but it means sustained automated benchmarking is not fully reliable without a testing-window exemption from rate/connection limits.

ffuf - Directory & Path Discovery Findings

A scan of 4,543 common paths was run against the live application. 4,542 of 4,543 requests (99.98%) returned an identical 403 response (status 403, 3,282 bytes) - consistent with CloFix detecting the enumeration pattern itself (many sequential path requests) rather than evaluating each path independently. This means the scan's raw output does not reflect genuine per-path discovery for the vast majority of requests tested, and should not be read as "4,542 paths do not exist" or "4,542 paths were correctly protected" - the uniform response indicates pattern-based blocking, not per-path evaluation.

Two results differed from the uniform block response and were manually verified against the live backend:

Path	Result	Verified Status
/index.html	403, 0 bytes (differed from standard block page)	False lead - confirmed 404 (does not exist) on direct follow-up
/static/	301 → /static/ → 200 OK	Confirmed real - directory listing enabled

Finding: Directory listing exposed at /static/, including bot-detection script

The /static/ directory has directory listing enabled, exposing its contents: challenge.js, and css/, data/, js/, png/, and topojson/ subdirectories. The file challenge.js is very likely the client-side logic behind the automation-fingerprinting behavior observed throughout this testing (see Methodology). Its direct accessibility means this detection logic can be retrieved and studied by any visitor, which may reduce its effectiveness as a defensive control. Recommended remediation: disable directory listing on /static/ and evaluate whether challenge.js should be served from a less discoverable path or obfuscated.

Given that 99.98% of this scan's results were obscured by pattern-based blocking, this should be considered a narrow, manually-verified result (one confirmed finding) rather than a comprehensive directory enumeration. A more complete assessment would require either a testing-window exemption

from enumeration-pattern detection, or a much slower, longer-duration scan spread over an extended period.

k6 - Performance Benchmark Findings

A light load test (5 concurrent virtual users, staged 15s ramp-up / 30s sustained / 15s ramp-down, ~75 seconds total) was run against the live application via the same header-normalized proxy.

Metric	Result
Requests completed	36 (0% failed - all received valid 200/302 responses)
Average latency	8.23s
Median latency	8.86s
p95 latency	10.87s
Max latency	12.12s
Effective throughput	~0.48 req/s (at 5 concurrent virtual users)

Important: This result differs substantially from the original report's performance claims

This report's original performance table claimed 50,000+ requests/second, <3ms average latency, and 10 Gbps peak throughput. The figures measured here are roughly three orders of magnitude slower. This should not be read as either confirming or refuting those original claims: this test ran against a small, publicly accessible demo instance that had been under near-continuous heavy testing load for the duration of this benchmark exercise (including a ~2-hour nuclei scan and a ~1h49m ffuf scan completed immediately prior), and its provisioning is very likely not equivalent to whatever dedicated environment produced the original figures. The WAF's per-request automation-scoring (JA3 fingerprinting, cookie/header validation, observed throughout this report) may also add processing overhead not present in a simpler passthrough configuration.

Recommendation: the original performance figures should not be republished without independent re-verification against a dedicated, unshared environment matching production specifications, measured without concurrent security-testing load. Given how significantly this measurement diverges from the original claims, the same verify-before-trusting standard applied to the sqlmap and false-positive findings in this report should apply here as well.

The following OWASP categories are not covered by the automated tools used in this report and require manual review or a dedicated scanning tool (e.g., OWASP ZAP):

OWASP Category	Coverage Status
A02 – Cryptographic Failures	Not covered - requires manual TLS/config review

OWASP Category	Coverage Status
A04 – Insecure Design	Not covered - requires manual architecture review
A08 – Software & Data Integrity Failures	Not covered - requires manual/CI pipeline review
A09 – Security Logging & Monitoring Failures	Not covered - requires verification via ELK/Grafana setup

Recommendations

1. Disable directory listing on /static/ and evaluate relocating or obfuscating challenge.js - its direct accessibility may allow the automation-detection logic to be studied and evaded.
2. Independently re-verify the original performance claims (50,000+ RPS, <3ms latency) against a dedicated, unshared test environment before republishing them - measured results on the public demo instance diverged by roughly three orders of magnitude, for reasons that need confirming (environment scale, concurrent load, or WAF processing overhead) before any performance figure is presented to clients.
3. Address the missing security headers (HSTS, CSP, X-Frame-Options, and related hardening headers) identified in nuclei scanning - low urgency individually, but straightforward to remediate as a batch.
4. Prioritize remediation of the free-text field false-positive issue before broader production deployment guidance is finalized, given the 98.6% block rate on benign content in this category.
5. Complete manual review of A02/A04/A08/A09 before this report is considered a complete OWASP Top 10 assessment.
6. Document the automation-detection layer's interaction with security testing tools, since any future benchmarking (internal or third-party) will need the same header-normalization approach used here to get accurate attack-detection figures.

Conclusion

CloFix WAF demonstrated strong, reproducible attack-detection performance across every OWASP category tested with automated tooling, with detection rates at or near 100% for injection, XSS, access control, and API security test cases. Testing also surfaced three issues worth prioritizing: a significant false-positive rate on free-text input fields, a directory-listing exposure revealing the WAF's own bot-detection script, and a large, unexplained gap between this report's original performance claims and what was measured on the public demo instance. Four OWASP categories (A02, A04, A08, A09) still require manual review beyond automated tooling. This report should be considered complete for automated testing scope, with the noted manual review and performance-verification items as the clear next steps before broader publication.

The findings presented in this report reflect the configuration of the public demonstration environment. Production deployments can enable enterprise rule packs, advanced AI protection, security hardening, and performance optimizations based on customer requirements.

Note: Security header findings, performance measurements, and infrastructure-related observations reported in this document reflect the configuration of the public demo environment only and should not be interpreted as limitations of CloFix WAF's production capabilities.