

Cyber Securitise Lab Report, October 2026

23 October 2026

Cyber Securitise detects 99.8% of the OWASP Core Rule Set's attack tests and blocked none of the real-visitor requests in our lab, on WordPress and, through the Gateway, on any web stack. Every figure below was measured in an isolated lab on the released product (plugin 1.25.0) and confirmed on live production sites.

At a glance

Measure	Result
OWASP CRS 4.30 attack tests detected (WordPress, default level)	1,661 of 1,665 (99.8%)
Same tests blocked end to end on a live WordPress site	1,551 of 1,646 (94.2%) ; only 17 (1.0%) served normally
Command, PHP and code injection, file inclusion, session fixation, protocol attacks	100% blocked
Real-visitor requests blocked within the tested corpus, all four paranoia levels	0 of 101 observed (23 live requests, 78 in the corpus)
Vulnerable plugin attacked by API or by direct file request	Blocked (403) until updated
Gateway, same attack tests, on HTML, Java, PHP, Rust and Node	94.9% blocked; 0 of 17 tested normal requests blocked , on every stack
Host Scanner, planted vulnerabilities across 5 languages and 6 Linux distributions	All found , with fixed version and command; 0 false alarms on up-to-date systems
Live fleet	32 production sites on 1.25.0, all serving normally

Scope of these figures. Detection rates refer to the OWASP Core Rule Set 4.30 regression suite under the configuration stated below, not to every possible real-world attack. False-positive results are what we observed in the tested traffic (101 WordPress requests, 17 per Gateway stack); they show compatibility with common traffic, not a guaranteed zero rate across all sites. The suite is published by OWASP, so anyone can re-run it. This is an engineering validation by Cyber Securitise, not an independent audit.

How it was tested

All tests ran in a sealed lab with no internet access, using the industry's own test suite, real-visitor traffic and a real vulnerable plugin.

- **Isolation:** Linux with Docker. The sites sit on an internal network with no internet (verified: no DNS, outbound connections fail) behind a single front door that only the test machine can reach. No client data or real keys were used.
- **WordPress site:** WordPress on PHP 8.3 and Apache, MariaDB 11.4, posts in English and Japanese.
- **Other stacks:** static HTML, Java (Spring Boot), PHP, Rust (Axum) and Node (Express), each behind the Gateway.
- **Attack suite:** the OWASP Core Rule Set 4.30 regression tests, about 5,100 attack and benign cases written by the CRS project, checksum-verified. Scored two ways: in the rule engine (did the right rule detect it?) and live through the web server (was the request blocked?).
- **Real-visitor traffic:** 23 requests with genuine browser headers: English and Japanese pages and searches, quotes and symbols, consent and shop cookies holding JSON, tracking links, feeds, sitemap, REST API, log-in, comments with apostrophes and Japanese text, AJAX calls, security reports, Googlebot and author pages. Plus a 78-request corpus taken from real WordPress traffic. Pass = not blocked.
- **Vulnerable plugin:** Really Simple Security 9.1.1, which has a published critical flaw (CVE-2024-10924).

- **Products tested:** WordPress plugin 1.25.0 (released 23 October 2026), Gateway (Coraza 3.8.1 with CRS 4.30.0), Host Scanner (OSV.dev and CISA KEV data).

WordPress firewall

The plugin runs the full OWASP Core Rule Set inside WordPress and catches every application-attack category at 98.9% or better, with no real visitor blocked.

Attack category (CRS)	Detected at the default level
Remote command execution (932)	100%
PHP injection (933)	100%
Generic code injection, SSRF (934)	100%
Local file inclusion (930)	100%
Remote file inclusion (931)	100%
Session fixation (943)	100%
Protocol attacks, request smuggling (921)	100%
Scanner detection (913)	100%
SQL injection (942)	320 of 322 (99.4%)
Cross-site scripting (941)	186 of 188 (98.9%)
All rules	1,661 of 1,665 (99.8%)

The four tests the engine does not detect disguise quotes as curly apostrophes; treating those as quotes would block ordinary WordPress text, so they are left out by design.

Detection versus blocking. The 99.8% is measured in the rule engine. Sent live through the web server, the same suite resolves as follows; only 1.0% of the tests were served normally.

Live outcome, 1,646 attack tests at level 1	Tests	Share
Blocked by the firewall (403)	1,551	94.2%
Refused by rate limiting (429): the suite fires about 5,000 attacks from one address, which trips the per-address limit	55	3.3%
Not found, nothing served (404)	19	1.2%
Rejected by the web server before WordPress (400 or connection closed)	4	0.2%
Served normally (200): 16 HTTP-protocol checks, 1 initialisation test	17	1.0%

The 17 served tests check raw request details (for example unusual header formats) that the web server normalises before WordPress sees them; none is an application attack. Per-test results are in the machine-readable results file (see Reproducing the results).

SQL injection by structure. The plugin includes a pure-PHP port of libinjection (BSD licence), the detector used by ModSecurity and Coraza. It recognises injection by its grammar, so tautology tricks such as ' OR 1=1 are blocked at the default level, while names like O'Brien, contractions and Japanese text pass. Checked against the reference implementation on 101,467 inputs. Cost: about 0.2 ms per request.

Paranoia levels. Site owners can raise the rule strictness from level 1 to 4. Higher levels start in watch-only mode, so nothing is blocked until the owner has seen what would be.

Level	Attack tests detected	Real visitor requests blocked
1 (default)	99.8%	0
2	92.2% of a larger, stricter set	0
3	91.1%	0
4	90.8%	0

Normal traffic covers consent and shop cookies holding JSON, XML in parameters, CSS values, inch and foot marks in searches, JSON in AJAX fields and Japanese text, all verified live at every level in block mode.

Virtual patching

When an installed plugin has a published vulnerability, Cyber Securitise closes every route to it until the plugin is updated, with no action from the site owner.

- **API routes:** the vulnerable plugin's AJAX actions and REST routes are refused for visitors who are not signed in.
- **Direct file requests:** the web server itself refuses PHP files in that plugin's folder ([.htaccess](#) on Apache and LiteSpeed, a generated rule for nginx). This stops the attack pattern behind the WP File Manager mass exploitation, where the attacker calls a file that never loads WordPress. Payment callbacks, captcha and cache endpoints stay open.
- **Automatic removal:** the moment the plugin is updated to a fixed version, every rule is lifted.

Test, Really Simple Security 9.1.1 (CVE-2024-10924)	Result
Vulnerable REST route, before patch	Reachable
Same route, patched	Blocked
Plugin PHP file by URL, patched	403
Home page and 23 normal requests while patched	200, 0 blocked
After update to a fixed version	Rule removed, file reachable again

Any stack: the Gateway

The Gateway brings the same OWASP-grade firewall to sites that are not WordPress, and blocked 94.9% of the attack suite on all five stacks tested with zero false blocks.

Stack	Attack tests blocked	Normal requests blocked (of 17)	Added delay
Static HTML	94.9%	0	3-7 ms
Java (Spring Boot)	94.9%	0	about 8 ms
PHP	94.9%	0	about 4 ms
Rust (Axum)	94.9%	0	about 5 ms
Node (Express)	94.9%	0	about 7 ms
Gateway alone, no proxy in front	95.6%	-	-

- **What it is:** a single 30 MB container, built on OWASP Coraza 3.8.1 and CRS 4.30.0 (pinned by checksum), with libinjection, the same false-positive tuning as the plugin, AI-crawler and scraper control, rate limits and security headers. All dependencies are Apache, MIT or BSD licensed.
- **HTTPS:** own certificates or free automatic certificates (ACME), HTTP/2, HTTPS redirect, HSTS; old TLS refused. Certificate issuance tested end to end.

- **Scales out:** rate limits shared across servers through Redis (2 gateways at 20 per minute allowed exactly 20); if Redis is unavailable, visitors are let through, never locked out.
- **Response checks:** blocks pages that leak database errors or contain a web shell; a 50 MB download and live event streams pass untouched.
- **Reporting:** sends blocks to the Cyber Securitise dashboard over the same signed protocol as the plugin, with retry and IP shortening by default.

Below the application: server software

Cyber Securitise also finds the holes a firewall cannot see, in PHP, the web server, the Linux kernel, OpenSSL, Node and the database, and gives the exact fix.

Server software check (in the plugin, 1.25.0). The plugin reads installed versions from files, never running a command, and the dashboard checks them against OSV.dev, CISA's list of exploited vulnerabilities and end-of-life dates. Distribution packages are matched against the distribution's own advisories, so Ubuntu and Debian backports are never reported as vulnerable from the version number alone, and an advisory only counts if the distribution lists that package as affected. Each finding names the system, the advisories not yet installed, whether any is on CISA's exploited list, and the exact upgrade command for that system (for example `apt-get install --only-upgrade` for the affected package). Where the build cannot be identified, as on managed hosting, the finding is a "check with your host" warning rather than an alert.

Host Scanner (any server). One read-only file, about 45 MB, that needs nothing installed; runs fully offline if required. It checks OS packages, Docker images, hand-installed runtimes and application lock files.

Planted in the lab	Found	Advisories on that version	Upgrade target reported (clears every advisory)
log4j-core 2.14.1 (Java), incl. Log4Shell	Yes, flagged "being exploited now"	7	2.25.4
lodash 4.17.20 (Node)	Yes	5	4.18.0
guzzlehttp/psr7 1.8.3 (PHP)	Yes	6	2.12.3
urllib3 1.26.4 (Python)	Yes	11	2.8.0
smallvec 1.6.0 (Rust)	Yes	RUSTSEC-2021-0003	1.6.1
Node 22.0.0 installed by hand	Yes, 9 security releases missing	-	22.23.4
Go 1.21 toolchain	Yes, end of life plus advisories	-	1.26.10

The scanner reports the version that clears every known advisory on the package, not just the first fix for one CVE: for Log4Shell, 2.15.0 fixed the original flaw but later advisories need 2.25.4. The first fixed version for each individual advisory is in the machine-readable output. Where that is a major upgrade (urllib3 1.x to 2.x, psr7 1.x to 2.x), the per-advisory versions show the smallest fix for each flaw.

System checked	Result
Old images: Debian 12.0, Rocky 9.0, AlmaLinux 9.0, UBI 9.0, Alpine 3.18	Real fixable findings on each (Rocky 9.0: 36)
Current Rocky and AlmaLinux 9	Exactly the vendor's own <code>dnf</code> security list
Up-to-date Ubuntu 24.04, Debian 12, UBI 9, Alpine	0 findings
Bare Ubuntu 22.04/24.04, Debian 12, Rocky 9 with no Python	Runs from the single file; leaves nothing behind

Online mode sends only package names, versions and ecosystems; no hostnames, IP addresses, paths or image names. Fixable findings are identical in offline mode.

Tamper watch

If anyone switches protection off or alters the plugin, the site owner is told, so an intruder cannot quietly disable the firewall.

Event	Dashboard response
Site stops reporting for 30 minutes (adjustable) while still online	Alert
Plugin files differ from the signed release	Alert
Firewall, rule set, paranoia level or virtual patching lowered by someone other than the owner	Critical alert
An unknown PHP auto-prepend file appears	Critical alert
Plugin deactivated	Alert, from a signed last report
Owner lowers protection from wp-admin	Logged, no alarm

Every release is signed; the dashboard computes the expected file fingerprint from the signed package when it is published. Legitimate updates and site downtime do not trigger false alarms (covered by 8 automated dashboard tests).

Found and fixed during testing

The lab is part of the release process: every issue it found was fixed, re-tested and shipped within days, between 1.23.11 and 1.25.0.

Found	Fix	Shipped
Virtual patching missed plugins whose API name drops the dashes of the plugin name	All name forms matched	1.23.11
Full test of every plugin function: 8 defects (scan checks that could not reach the site, <code>.htaccess</code> fixes from the command line, restored files re-quarantined, plan check for command-line backups, own loader and own probes reported as threats, JSON output, password reset when email fails)	All 8 fixed, with tests	1.24.0
Paranoia levels 2-4 blocked pages with consent and shop cookies	Stricter SQL rules no longer read cookies	1.24.0, before release
libinjection port flagged XML declarations, CSS values and one plain sentence	Three narrow, documented exceptions	1.24.1, before release
Levels 2-4 blocked XML in parameters, CSS values, inch and foot marks, JSON in AJAX fields	Per-field exceptions; detection unchanged	1.24.3
Live logs showed crawlers sending a Content-Type header on page requests being blocked	Content type ignored on requests without a body, on every server type	1.24.2, 1.24.4

Each fix was verified on the same lab tests and on live sites before the next release.

In production

The tested release runs on real business sites today, and the network defences that a sealed lab cannot exercise were measured on live traffic instead.

-
- **Fleet:** 32 production WordPress sites in Japan, Singapore and the US, all on 1.25.0 and serving normally (home pages and page requests with unusual headers checked after each release).
 - **Dashboard:** sites report status, findings and blocks continuously; 31 sites were sending tamper-watch heartbeats within an hour of the 1.25.0 rollout.
 - **Network blocks, 19-23 October 2026:** about 2,250 requests from Spamhaus DROP networks (address ranges run by criminal operations) and about 2,100 from data-centre networks on log-in and admin pages, where real visitors do not come from.
 - **Rapid response:** when PHP disclosed new code-execution flaws on 21 and 23 October 2026, the fleet was patched or mitigated within days; the published mitigation is in place on all 32 sites.

Reproducing the results

The lab is scripted, and every number in this report can be re-run from the OWASP Core Rule Set 4.30.0 regression tests, the real-visitor traffic scripts and the product's own test suites (1,276 plugin tests, Gateway tests, Host Scanner self-test and dashboard tests). The configuration was: firewall in block mode, OWASP rules on in every category, anomaly threshold 5, paranoia level 1 (levels 2-4 tested separately), virtual patching on. The signed release hash, the exact configuration, the command sequence and the per-test machine-readable results are available to customers and reviewers on request.

Versions: WordPress plugin 1.25.0 (signed release), OWASP CRS 4.30.0, Coraza 3.8.1, osv-scanner 2.6.0.