macronimous
web · mobile · seo · solutions

SECURITY ASSESSMENT — SAMPLE REPORT

# WordPress Security Audit

example-client.com

| | |
|---|---|
| AUDIT DATE | August 2026 |
| MODE | Full — SSH + WP-CLI, plus external surface testing |
| PREPARED BY | Macronimous Web Solutions |
| PLATFORM | WordPress 7.0.3 · shared LiteSpeed hosting |
| CLASSIFICATION | Sample report — identifiers fictionalized |

*This is a genuine audit of an eleven-year-old production site, published with permission. The domain, usernames, file paths, IP addresses, and theme name have been fictionalized; every finding, count, and command is unchanged. All Critical findings were remediated the same day.*

| 4 | 8 | 8 | 12 |
|---|---|---|---|
| CRITICAL | IMPORTANT | POLISH | PASSED |

# Executive summary

The site is running current, unmodified WordPress core and carries no active malware. The significant finding is historical: **an administrator account created by an attacker in February 2022 is still present in the database**, and it was used to log in as recently as August 2023. Alongside it, the site's 2010-era theme ships two files that would give an attacker code execution and a free spam relay — both currently blocked by a firewall rule that is one configuration rewrite away from disappearing.

Most remaining findings are accumulated debris from roughly eleven years of operation and at least two hosting migrations: forgotten diagnostic scripts, publicly readable log files, downloadable plugin archives, and a web server still running a version of PHP that stopped receiving security patches in 2022. Individually minor; collectively they hand an attacker a detailed map of the site.

Encouragingly, the security tooling already in place is doing real work: blocking directory listings, REST user enumeration, and direct access to theme PHP files, and enforcing brute-force lockouts. Several of the Critical findings below are dangerous *precisely because* they are neutralized by a single rule rather than actually removed.

## Top three actions

1. **Delete the attacker's administrator account** (user ID 5), then rotate the security salts and reset both genuine admin passwords. This closes out a real, evidenced intrusion.
2. **Delete four files from the active theme and webroot** — timthumb.php, sendmail.php, quotesend.php, preflight.php. Roughly five minutes of work that removes the site's two most severe latent vulnerabilities.
3. **Remove the WP_ALLOW_REPAIR flag and move the web server to PHP 8.2.** The first exposes a database tool to anonymous visitors today; the second ends four years of unpatched PHP.

## Scorecard

| AREA | STATUS | BASIS |
|---|---|---|
| **Core integrity** | PASS | WordPress 7.0.3, latest available. wp core verify-checksums passes — no modified or injected core files. |
| **Malware & compromise** | FAIL | No active malware, no PHP in uploads, no obfuscated code. However, an attacker-created admin account from a 2022 intrusion persists. |
| **Plugins** | PASS | All six active plugins cross-checked against CVE databases. No unpatched vulnerabilities. One non-security update pending. |
| **Themes** | FAIL | Active theme is an abandoned 2010 premium theme bundling TimThumb 1.08 (CVE-2011-4106) and an unauthenticated mail script. |
| **Configuration** | PARTIAL | Salts unique, file editor disabled, permissions correct. But WP_ALLOW_REPAIR is enabled and stale cross-host includes remain. |
| **Users** | FAIL | Two legitimate administrators. One rogue administrator account with no email and a null registration date. |
| **Server** | FAIL | Web tier runs PHP 7.4.33, end-of-life since November 2022. CLI runs 8.2 — the mismatch masked this. |
| **External surface** | PARTIAL | HTTPS enforced, directory listing and REST enumeration blocked. But xmlrpc is fully open, logs and plugin archives are publicly downloadable. |

# Findings — Critical

**Critical** Actively exploitable or evidence of compromise · fix within 24–48 hours

| # | FINDING | RISK IN PLAIN LANGUAGE | FIX | EFFORT |
|---|---|---|---|---|
| 1 | **Backdoor administrator account**<br>`User ID 5, wp.service.controller.YktzH` | An attacker created this account by writing directly into the database in February 2022 — it has no email address and a null registration date, which normal signup cannot produce. Records show it successfully logged in and was last active in August 2023, from a hosting-provider IP (192.0.2.66).<br>*Mitigating detail:* the attacker's script assumed the default database table prefix, so WordPress does not currently grant it admin rights. It can still log in, and would regain full control if the prefix were ever normalized. | `wp user delete 5 \ --reassign=1 --yes`<br>Then invalidate anything the attacker may still hold:<br>`wp config shuffle-salts`<br>`wp user reset-password 1`<br>`wp user reset-password 4` | QUICK |
| 2 | **TimThumb 1.08 in the active theme**<br>`CVE-2011-4106 · themes/veltra/timthumb.php` | This image-resizing script can be tricked into downloading an attacker's file and saving it as runnable code on your server, giving them control of the site (remote code execution). It is one of the most heavily exploited WordPress vulnerabilities in history, and this is a pre-patch version.<br>*Currently blocked* — verified returning 403 via a firewall rule. That rule lives in .htaccess, which WordPress rewrites on permalink changes, plugin deactivation, and host migrations. The day it is rewritten, this becomes live. | `cd wp-content/themes`<br>`rm veltra/timthumb.php`<br>Thumbnails currently return 403 anyway, so they are already broken. Replace with WordPress's native the_post_thumbnail(). | MODERATE |
| 3 | **Anonymous database repair endpoint**<br>`WP_ALLOW_REPAIR in wp-config.php` | Verified live: /wp-admin/maint/repair.php returns a working “Repair and Optimize Database” page to a visitor who is not logged in. WordPress deliberately bypasses login for this flag so you can rescue a site you are locked out of — leaving it | `wp config delete WP_ALLOW_REPAIR` | QUICK |

| # | FINDING | RISK IN PLAIN LANGUAGE | FIX | EFFORT |
|---|---|---|---|---|
| | | enabled lets anyone lock your database on demand, repeatedly, taking the site offline. | | |
| 4 | **Open mail relay in the theme**<br>`themes/veltra/sendmail.php · quotesend.php` | The script takes its recipient straight from the submitted form with no validation, so anyone can send email to any address through your server. Sender values also flow unfiltered into the message, allowing header injection. In practice this means your domain and server IP get used for spam and end up blacklisted, breaking your genuine business email.<br>Currently returning 403 via the same fragile rule as finding 2. | `cd wp-content/themes`<br>`rm veltra/sendmail.php`<br>`rm veltra/quotesend.php`<br>Route contact forms through Contact Form 7, which locks the recipient server-side and enforces nonces. | QUICK |

# Findings — Important

**Important** Meaningful risk reduction · fix within 2 weeks

| # | FINDING | RISK IN PLAIN LANGUAGE | FIX | EFFORT |
|---|---|---|---|---|
| 5 | **Web server runs PHP 7.4.33**<br>`End-of-life Nov 2022` | The PHP version actually running your site stopped receiving security patches almost four years ago. Any PHP-level vulnerability found since then is permanently unfixed here. This was easy to miss: the command line runs PHP 8.2, so a casual check looks fine. | Hosting panel → PHP Configuration → set 8.2 for this domain. Test afterwards — the 2010 theme is the most likely to complain. | MODERATE |
| 6 | **Diagnostic script publicly executable**<br>`wp-content/preflight.php` | Returns 200 to anyone. It publicly reports your PHP version, installed extensions, memory and execution limits, mail availability and outbound API reachability, and presents a form that accepts database credentials. Its own header instructs deletion after use. It sits in wp-content/, which the theme and plugin firewall rules do not cover. | `rm wp-content/preflight.php` | QUICK |
| 7 | **Mail and error logs publicly readable**<br>`/php_mail.log · /php_errorlog` | Both download freely. The mail log exposes genuine recipient email addresses (including staff and client addresses) along with historical server paths; the error log exposes internal filesystem structure. This is both a privacy exposure and a reconnaissance aid. | `rm php_mail.log`<br>`rm php_errorlog`<br>`find . -name "php_errorlog" \`<br>`  -delete`<br>Copies also exist in wp-admin/ and wp-includes/. | QUICK |
| 8 | **Plugin archives and php.ini downloadable**<br>`Six .zip files in wp-content/plugins/` | All six archives download successfully — ZIP files are not covered by the PHP-blocking rules. They hand an attacker a precise history of what this site has run, and several are old enough to have public exploits. /php.ini is also readable, disclosing server configuration. | `cd wp-content/plugins`<br>`rm -f *.zip`<br>Move php.ini out of the webroot or deny it in .htaccess. | QUICK |
| 9 | **xmlrpc.php fully open**<br>`80 methods exposed` | Includes pingback.ping, which lets attackers bounce traffic off your server to attack third parties and probe your internal network, and system.multicall, which packs hundreds of password guesses into a single request — largely sidestepping | Security plugin → WordPress Tweaks → XML-RPC: Disable. Keep enabled only if you use Jetpack or the WordPress mobile apps. | QUICK |

| # | FINDING | RISK IN PLAIN LANGUAGE | FIX | EFFORT |
|---|---|---|---|---|
| | | the login rate limiting you have in place. | | |
| **10** | **Custom management-plugin endpoint live in production**<br>`(plugin name withheld)` | The plugin's code was reviewed and is sound — long random token, constant-time comparison, disabled by default, rate limiting correctly placed before authentication. **No vulnerability was found.**<br>The concern is deployment: the token travels in the URL path, so it is written into web server access logs, proxy logs and Referer headers — on a site that, per findings 6–8, has demonstrably exposed log files. Whoever holds that token can enumerate your full plugin estate, database, and cron. | Disable the endpoint on production if not actively in use, or rotate the token and confirm access logs are not web-reachable. | QUICK |
| **11** | **Active theme abandoned since 2010**<br>`VELTRA 1.0 (ThemeForest)` | Internal filenames date the theme precisely — one backup file is stamped July 2010. It receives no security updates and never will. Findings 2 and 4 both originate here, and any future vulnerability in it will also go unpatched. | Plan a migration to a maintained theme. A current, maintained theme is already installed and is a legitimate starting point. | INVOLVED |
| **12** | **All in One SEO behind current**<br>`4.9.10 → 5.0.0.1` | Hygiene rather than exposure. To be accurate: your installed version is **not** vulnerable — the two published 2025 CVEs affect earlier versions only. The plugin does disclose its version publicly in a generator tag. | `wp plugin update \ all-in-one-seo-pack` | QUICK |

# Findings — Polish

**Polish** Hardening and hygiene · no fixed deadline

| # | FINDING | RISK IN PLAIN LANGUAGE | FIX | EFFORT |
|---|---|---|---|---|
| 13 | **Username enumeration** | /?author=1 redirects to /author/example8usr/, handing an attacker a confirmed admin username. Combined with finding 9, that is half of a login. | Security plugin → WordPress Tweaks → block user enumeration. | QUICK |
| 14 | **Missing security headers** | No HSTS, X-Frame-Options, X-Content-Type-Options or Referrer-Policy. The x-powered-by header also advertises your exact PHP version to anyone. | Add the four headers via .htaccess; remove X-Powered-By in PHP configuration. | QUICK |
| 15 | **wp-config.php permissions** | Set to 644 (readable by every account on the shared server). It holds your database credentials. | `chmod 640 wp-config.php` | QUICK |
| 16 | **Site URL still http://** | Works via a server-level redirect, but costs every visitor an extra hop and risks mixed-content warnings. | `wp option update siteurl \ 'https://example-client.com'` — and the same for home. | QUICK |
| 17 | **Stale cross-host includes** | wp-config still loads a path from a previous hosting provider. Verified inert — the path does not exist and is not writable — but it would execute code before WordPress boots if it ever did. | Delete the two @include_once lines from wp-config.php. | QUICK |
| 18 | **Eleven unused themes** | Every inactive theme is still code on disk that must be kept patched. Inactive Akismet likewise. | Delete unused themes, retaining one current default as a fallback. | QUICK |
| 19 | **Stale security-plugin paths** | The security plugin still writes logs to a dead pre-migration path, so some logging may silently not work. | Re-save the plugin's settings to regenerate paths. | QUICK |
| 20 | **Webroot clutter** | Default.html (the pre-WordPress site), sitemap.backup.xml, .htaccessbk, and deprecated core stubs. Each was read individually — the deprecated files are genuine WordPress leftovers, not backdoors — but they expand the surface for no benefit. | Remove the obsolete files from the webroot. | QUICK |

## Checks that passed

**A meaningful amount of this site is in good shape.**

Several Critical findings above are dangerous specifically because existing protections are holding them back — the protections work; the underlying files simply need removing.

- WordPress core 7.0.3, current, checksums verify clean
- No modified or injected core files
- No mu-plugins directory (a common malware hiding place)
- No PHP files in the uploads directory
- No obfuscated code — every match traced to legitimate libraries
- .htaccess clean, no injected redirects
- Security salts unique, no placeholder values
- DISALLOW_FILE_EDIT enabled
- File permissions correct, nothing world-writable
- REST API user enumeration blocked
- Brute-force lockout active (3 attempts / 15 minutes)
- Directory listing disabled; readme.html blocked
- HTTP to HTTPS redirect working
- All six active plugins free of unpatched CVEs

## Suggested order of work

| WHEN | FINDINGS | WORK |
|---|---|---|
| **Today** | 1 | Delete user ID 5 (with --reassign), shuffle salts, reset both admin passwords. Closes the evidenced intrusion. |
| **Today** | 2, 4, 6, 7, 8 | Delete timthumb.php, sendmail.php, quotesend.php, preflight.php, the log files and the plugin archives. All deletions — roughly five minutes together. |
| **Today** | 3 | Remove the WP_ALLOW_REPAIR flag. |
| **This week** | 5, 9, 10, 12 | Move to PHP 8.2, update All in One SEO, disable xmlrpc, decide on the custom management endpoint. |
| **This month** | 13–20 | Work through the Polish tier, then plan the theme migration (11) — which retires this entire class of problem permanently. |

## Methodology & limitations

This was a read-only audit conducted in August 2026 with full server access (SSH and WP-CLI) plus external surface testing. It comprised: core checksum verification against official WordPress hashes; a complete plugin and theme inventory cross-checked against the WPScan, Patchstack, Wordfence and NVD vulnerability databases by version; a wp-config and file permission review; a malware and compromise sweep covering uploads, mu-plugins, obfuscation patterns, .htaccess and user accounts; a triaged source review of the site's custom plugin; and passive external checks of headers, xmlrpc, user enumeration and file exposure.

**No intrusive testing was performed.** No exploits were attempted, no credentials were brute-forced, and no payloads were injected. Vulnerability matching is version-based, which means a back-ported vendor patch could cause a version to appear vulnerable when it is not. This is a security audit, not a penetration test.

> **Scope limitation worth stating plainly**
>
> Evidence shows the site was compromised in February 2022, with attacker activity through August 2023. No surviving malware and no modified core files were found, and the firewall rules added since have closed the likely entry routes. However, absence of evidence three years after the fact is not proof that nothing else was left behind — file timestamps that old are not reliable either way. Definitive assurance would require a database-level review of wp_options for injected autoloaded entries, plus a diff against a known-good backup predating February 2022.