



EXTERNAL SECURITY ASSESSMENT

Security Assessment

Low-invasive external penetration test

OWASP / PTES methodology

This report presents the results of passive reconnaissance (OSINT), TLS/SSL configuration analysis, an HTTP security header audit, WordPress CMS fingerprinting, and a review of the domain's email protection.

Target system: [REDACTED]. The assessment was performed with minimal impact on the client's infrastructure.

Field	Value
Target asset	[REDACTED]
Report reference	[REDACTED]
Prepared by	Synt World
Original report date	[REDACTED]
Version	1.2 (public sample edition, 2026-10)
Classification	[REDACTED]

OWASP WSTG / PTES / CIS

Anonymized sample. English translation. Redacted information has not been restored.

PUBLIC SAMPLE | EDITORIAL REVISION 1.2

Some findings and evidence were removed before publication. Original finding IDs and section order are retained; the page layout of this edition differs from the original report. Revision 1.1 reconciled numeric scores with the supplied CVSS v3.1 vectors. Revision 1.2 re-rates three entries whose original ratings were not supported by the retained evidence (CRIT-02, HIGH-02 and MED-05; see section 4.1). These are editorial corrections, not a new assessment or a retest. Where evidence is redacted, impact cannot be independently validated from this sample. Three low-priority entries have no supplied vectors. Counts and ratings in this edition apply only to retained entries and are not an invoice.

Contents

1. Executive Summary	4
2. Scope, Methodology and Rules of Engagement	5
2.1. Scope	5
2.2. Methodology	5
2.3. Rules of Engagement	6
3. Technology Stack and Asset Map	6
3.1. Network Infrastructure	6
3.2. Web Stack	6
3.3. Discovered User Accounts	7
3.4. External Integrations (Third Parties)	7
4. Summary of Findings (Risk Matrix)	8
4.1. Rating Revisions in Edition 1.2	9
5. Critical Findings	10
5.1. CRIT-01 – CORS Misconfiguration with Credentials	10
5.2. CRIT-02 – PHP 7.4.33 End-of-Life	11
5.3. CRIT-03 – WordPress 6.0.x EOL and Outdated Plugins	12
6. High-Severity Findings	13
6.1. HIGH-01 – User Enumeration via REST API	13
6.2. HIGH-02 – Missing Security Headers (Hardening)	14
6.3. HIGH-03 – Email Security Misconfiguration	15
6.4. HIGH-04 – Directory Listing in wp-content	16
7. Medium-Severity Findings	17
7.1. MED-01 – Missing CAA DNS Records	17
7.2. MED-02 – Accessible readme.html, license.txt, wp-cron.php	17
7.3. MED-03 – Missing OCSP Stapling	18
7.4. MED-04 – Wildcard Certificate *.[REDACTED]	18
7.5. MED-05 – .git/HEAD Returns 403	18
8. Low-Severity Findings and Informational Notes	20
8.1. LOW-01 – Missing /.well-known/security.txt	20
8.2. LOW-02 – Missing BIMl Record	20
8.3. LOW-03 – RSS Feed Generates Fingerprinting Patterns	20
8.4. INFO-01 – Google Site Verification in TXT	21
8.5. Positive Findings (What Is Configured Well)	21
9. Detailed TLS/SSL Analysis	22
9.1. Certificate	22

9.2. Supported Protocols and Ciphers	22
9.3. Comparison with Mozilla Intermediate Recommendations	22
9.4. testssl.sh Commands for Retesting	22
10. DNS and Email Infrastructure Analysis	23
10.1. Complete DNS Record Map	23
10.2. RDAP/WHOIS Analysis	23
10.3. Email Protection Analysis and Phishing Attack Scenario	23
10.4. Recommended Target DNS Configuration	23
11. WordPress Hardening Checklist	24
11.1. Critical Fixes (24 Hours)	24
11.2. Urgent Fixes (1 Week)	24
11.3. Scheduled Fixes (1 Month)	24
11.4. Recommended Improvements (3 Months)	25
12. Subdomain Takeover Assessment	25
12.1. Subdomain Check Results	25
13. Remediation Roadmap	26
Appendix A. Audit Reproduction Commands	27
A.1. DNS Reconnaissance	27
A.2. TLS/SSL Analysis	27
A.3. HTTP Header Audit	27
A.4. WordPress Fingerprinting	27
A.5. CORS Verification	27
A.6. Sensitive Path Probe	27
A.7. Subdomain Enumeration	27
Appendix B. Recommended Tools for Future Assessments	27
B.1. Passive OSINT	27
B.2. Active Scanning (Low-Invasive)	28
B.3. WordPress-Specific Tools	28
B.4. Directory Fuzzing (Requires an Expanded Scope)	28
Appendix C. References and Standards	29
C.1. Standards and Methodologies	29
C.2. CWE References	29
C.3. RFCs	29
C.4. Legal Acts ([REDACTED])	30
C.5. Tools and Resources	30

Page numbers refer to the pages of this PDF, counting the title page as page 1.

1. Executive Summary

This report presents the results of a low-invasive external security assessment of the web resource [REDACTED], conducted at the client's request on [REDACTED]. The assessment was performed under a penetration testing agreement [REDACTED], using the OWASP Web Security Testing Guide (WSTG v4.2), PTES and CIS Benchmarks.

The objective was to assess the external attack surface without destructive effects on the client's business processes and to identify vulnerabilities available to an unauthenticated external attacker.

The work included passive reconnaissance (OSINT) of 16 common subdomains and DNS/WHOIS/TLS infrastructure, analysis of HTTP security headers and server responses to standard requests, fingerprinting of the root WordPress CMS without authentication, and assessment of domain email configuration (SPF, DKIM, DMARC, MTA-STS) and TLS/SSL. Active checks were limited to individual HTTP requests to common WordPress endpoints and checks of responses to standard sensitive-file paths. High-intensity scanning (fuzzing, password brute force and exploitation) was not performed, in line with the client's requirement for a low-invasive assessment.

This anonymized edition retains 16 entries: vulnerabilities, configuration weaknesses and informational observations. Some findings and supporting materials were removed from the original assessment. The retained items cover CORS configuration, PHP and WordPress components, public API metadata, security headers, DNS/email controls and directory listings. Counts in this edition describe only the retained material; they are not a count of billable confirmed vulnerabilities.

Public API responses exposed account metadata; the sample does not establish account roles. Other observations include missing security headers, absent DMARC and MTA-STS records, no DKIM record at the tested selector, and missing CAA records. Directory listings were reported under `/wp-content/plugins/` and `/wp-content/themes/`. Evidence and identifiers remain redacted in this public sample.

Overall recommendation. Prioritize validation and correction of the reported CORS configuration, then review PHP, WordPress and plugin support and update compatibility. Follow the remediation roadmap for the remaining controls. Exploitation and administrative compromise were not demonstrated within the stated low-invasive scope; potential attack chains require separate validation.

Edition 1.2. Three ratings were revised because the retained evidence did not support them: CRIT-02 (unsupported PHP version, observed through a response header) is now Medium; HIGH-02 (missing security headers) is now Low, as a hardening item; and MED-05 (`/git/HEAD` returning HTTP 403) is now Informational. Original IDs are kept. See section 4.1.

2. Scope, Methodology and Rules of Engagement

2.1. Scope

The assessment target is the external web resource [REDACTED]. The assessment is limited to the external attack surface: DNS, TLS, HTTP headers, publicly accessible WordPress endpoints, domain email records (SPF/DKIM/DMARC/MTA-STS), and subdomains found through CT logs and passive sources. The scope excludes the client's internal network, servers outside IONOS hosting, employee applications, mobile applications and third-party resources (Google Workspace and social media accounts).

Parameter	Value
Target domain	[REDACTED]
IPv4 address	[REDACTED]
IPv6 address	[REDACTED]
Hosting provider	[REDACTED]
Reverse DNS	[REDACTED]
DNS servers	[REDACTED]
Email provider	Google Workspace (aspmx.l.google.com)
Testing period	[REDACTED]
Testing depth	Passive + low-invasive active checks (no fuzzing or brute force)

2.2. Methodology

The assessment followed these standards and methodologies: OWASP Web Security Testing Guide (WSTG) v4.2 as the primary web testing methodology; PTES (Penetration Testing Execution Standard) for the overall testing methodology; CIS Benchmarks for Apache/PHP configuration assessment; OWASP Top 10 2021 for vulnerability classification; and Mozilla SSL Configuration Generator (Intermediate level) for TLS assessment. All checks were isolated from production users, with minimal server load (no more than one request per second for active checks, and fully passive OSINT).

Work stages:

- Passive OSINT: DNS reconnaissance, RDAP/WHOIS, certificate transparency (crt.sh), subdomain enumeration through passive sources, and technology discovery from headers.
- TLS/SSL analysis: protocols, ciphers, certificate chain, OCSP stapling and HSTS checks using openssl s_client.
- HTTP header audit: security headers on the root page and key WordPress endpoints.
- CMS fingerprinting: WordPress version, themes and plugins identified through the generator meta tag, static resource links, REST API and default files.
- Email security audit: SPF, DKIM, DMARC, MTA-STS, CAA and BIML checks.
- Subdomain takeover check: 16 common subdomains checked for existence and dangling CNAME records.
- Sensitive path probe: individual requests to common sensitive paths (.git/, .env, wp-config.php and backup files).
- CORS verification: server response to a request with an arbitrary Origin and an Authorization header.

2.3. Rules of Engagement

[REDACTED]. Incident escalation contacts were agreed in advance. In the event of an unexpected impact on service availability, such as an accidental WAF trigger, testing must stop immediately and the client's designated contact must be notified through the agreed channels.

The target domain is subject to the jurisdiction of [REDACTED]. No server-side modifications were made during the assessment; all requests were read-only.

Disclosure of findings: this report is confidential and is provided exclusively to the client's authorized representatives. Publication of findings, including CVEs in WordPress plugins, is not permitted before remediation or expiry of the 90-day responsible disclosure period. After remediation, the client may decide independently whether to disclose selected aspects of the assessment publicly for marketing or reputational purposes.

3. Technology Stack and Asset Map

The following technical configuration of the target resource was identified. The asset map is based on passive reconnaissance (DNS, TLS, HTTP headers and content analysis) and does not use active port scanning or high-intensity service fingerprinting.

3.1. Network Infrastructure

Component	Value	Source
IPv4 address	[REDACTED]	dig A
IPv6 address	[REDACTED]	dig AAAA
Hosting	[REDACTED]	ipinfo.io
Reverse DNS	[REDACTED]	dig -x
DNS servers (NS)	[REDACTED]	dig NS
SOA	[REDACTED]	dig SOA
Mail (MX)	[REDACTED]	dig MX
Hosting type	[REDACTED]	DNS pattern

3.2. Web Stack

Layer	Component	Version	Status
Frontend proxy	nginx	Not disclosed	OK
Web server	Apache	Not disclosed	OK
PHP	PHP	[REDACTED]	EOL (CRIT-02)
CMS	WordPress	[REDACTED]	EOL branch (CRIT-03)
Theme	Avada	Not determined	Requires verification
Plugin	Slider Revolution	[REDACTED]	Outdated (CRIT-03)
Plugin	Yoast SEO	[REDACTED]	Outdated

Layer	Component	Version	Status
Plugin	Gtranslate	Not determined	Requires verification
TLS certificate	Sectigo (wildcard)	[REDACTED]	OK

3.3. Discovered User Accounts

The /wp-json/wp/v2/users REST API exposed two user accounts (see HIGH-01). This can assist targeted login attempts and social engineering.

ID	Login (slug)	Display name	Note
1	[REDACTED]	[REDACTED]	ID=1 is usually the account created at installation; its role is not established in this sample (see HIGH-01)
[REDACTED]	[REDACTED]	[REDACTED]	Second account; role not established

3.4. External Integrations (Third Parties)

The site is integrated with the following external services, identified through homepage content analysis and HTTP headers:

- Google Workspace: corporate email ([REDACTED] through aspmx.l.google.com), also including google-site-verification in DNS TXT records.
- Gravatar: the WordPress avatar service (secure.gravatar.com).
- Yoast SEO: an SEO plugin that automatically generates schema.org JSON-LD.
- Social media: links to Facebook ([REDACTED]), Instagram ([REDACTED]), Pinterest ([REDACTED]) and Yelp.
- IONOS Web Hosting: infrastructure provider with its own WAF/rate limiter (x-ws-origin and x-ws-ratelimit-* headers).

4. Summary of Findings (Risk Matrix)

The matrix covers the 16 retained entries. Original IDs and section order are preserved for traceability; ID prefixes are historical labels and do not indicate the current rating. Scores with supplied vectors have been recalculated using CVSS v3.1. These are source-based estimates, not a new validation of impact; the CRIT-02 vector is an editorial estimate added in edition 1.2. LOW-01 to LOW-03 have no supplied vectors; their source estimates are not reproducible CVSS scores. HIGH-02 and MED-05 are not CVSS-scored. Entries marked * in the ID column were re-rated in edition 1.2 (section 4.1).

2	2	4	6	2	16
CRITICAL	HIGH	MEDIUM	LOW	INFO	RETAINED

ID	Finding	Severity	CVSS	CWE	Status
CRIT-01	CORS Misconfiguration with Credentials	CRITICAL	9.3	CWE-942	Open
CRIT-02*	PHP 7.4.33 – End-of-Life	MEDIUM	5.6	CWE-1104	Open
CRIT-03	WordPress 6.0.x EOL + outdated plugins	CRITICAL	9.1	CWE-1104	Open
HIGH-01	User Enumeration via REST API	HIGH	7.5	CWE-200	Open
HIGH-02*	Missing Security Headers (HSTS/CSP/X-Frame/etc.), hardening	LOW	-	CWE-693	Open
HIGH-03	Email: no DMARC/DKIM/MTA-STS; SPF ~all	HIGH	8.1	CWE-16	Open
HIGH-04	Directory listing in wp-content/plugins and themes	MEDIUM	5.3	CWE-538	Open
MED-01	Missing CAA DNS records	MEDIUM	6.5	CWE-295	Open
MED-02	Accessible readme.html, license.txt, wp-cron.php	MEDIUM	5.3	CWE-200	Open
MED-03	No OCSP Stapling	LOW	3.7	CWE-299	Open
MED-04	Wildcard certificate *[REDACTED]	LOW	3.8	CWE-295	Open
MED-05*	.git/HEAD returns 403 (potential .git)	INFO	-	CWE-538	Further verification required
LOW-01	Missing /.well-known/security.txt	LOW	2.0	CWE-200	Open
LOW-02	Missing BIML record	LOW	1.5	-	Open
LOW-03	RSS feed generates fingerprinting patterns	LOW	2.7	CWE-200	Open
INFO-01	Google site verification in TXT (normal)	INFO	-	-	OK

Retained entries by revised rating: 2 Critical, 2 High, 4 Medium, 6 Low and 2 Info. The Low group includes three source priority estimates without CVSS vectors and one unscored hardening entry (HIGH-02); the Info group includes MED-05, which records an unconfirmed observation. Ratings remain provisional where evidence is redacted; informational and hardening observations are not automatically confirmed vulnerabilities. Original IDs and section groupings are retained even where a recalculated rating differs. No average score or billing total is inferred from this sample.

4.1. Rating Revisions in Edition 1.2

Three entries were re-rated from the retained evidence. No new testing was performed; each change lowers a rating that the documented observation did not support. CVSS v3.1 scores specific vulnerabilities, so HIGH-02 (a missing protection) and MED-05 (an unconfirmed observation) are rated without a score, and the CRIT-02 vector is an editorial estimate. Remediation advice is unchanged except for the scheduling of HIGH-02 (sections 11 and 13) and dated notes in CRIT-02.

ID	Edition 1.1	Edition 1.2	Reason
CRIT-02	Critical 9.1 AV:N/AC:L/PR:N/UI:N/ S:U/C:H/I:H/A:N	Medium 5.6 (editorial estimate) AV:N/AC:H/PR:N/UI:N/ S:U/C:L/I:L/A:L	Based only on an X-Powered-By version banner. No applicable CVE was mapped to the deployed build in the retained material and no exploit was demonstrated; the entry's own impact statement says no RCE is established.
HIGH-02	High 8.2 AV:N/AC:L/PR:N/UI:N/ S:U/C:H/I:L/A:N	Low, hardening (not CVSS-scored)	Missing headers are absent defence-in-depth, not a vulnerability by themselves. The direct exposures (HTTPS downgrade without HSTS, framing of public pages) need an attacker on the network path or a separate flaw.
MED-05	Medium 5.3 AV:N/AC:L/PR:N/UI:N/ S:U/C:L/I:N/A:N	Info (not scored)	HTTP 403 means the server refused to return the file: nothing was disclosed. The 403/404 difference may come from a blanket deny rule; it is kept as a verification item.

5. Critical Findings

Grouped by original rating. Revised in edition 1.2: CRIT-02 is now Medium.

5.1. CRIT-01 – CORS Misconfiguration with Credentials

CRITICAL 9.3

SEVERITY	Critical (CVSS 9.3)
CVSS V3.1	AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:N
CWE	CWE-942, Permissive Cross-domain Policy with Untrusted Domains
OWASP	A01:2021 Broken Access Control

Description. The source assessment reports reflection of an arbitrary Origin together with Access-Control-Allow-Credentials: true. This can be unsafe on endpoints that expose sensitive authenticated responses. Headers alone do not establish exploitation: credential delivery, cookie policy, endpoint authentication and WordPress nonce checks must also be verified.

Evidence (PoC):

[REDACTED]

Potential impact. A malicious origin may read sensitive responses if the browser sends valid credentials and the endpoint accepts the request. The redacted sample does not demonstrate those conditions or an administrative action.

Potential follow-on actions depend on the permissions and authentication requirements of individual REST endpoints. Administrative changes or remote code execution would require additional demonstrated conditions. These outcomes are not established by the header observation alone; no exploitation has been added to this sample.

Recommendations:

- In Apache or the nginx frontend, explicitly allowlist only the client's domain as an allowed Origin: [REDACTED].
- Disable Access-Control-Allow-Credentials: true entirely if third-party sites should not make authenticated requests.
- If credentials are required for the site's own frontend, validate Origin on the server and return AC-AO only for allowlisted origins.
- Review plugin settings, particularly REST API extensions, that may generate CORS headers independently.
- After remediation, repeat the curl check with an arbitrary Origin. The expected response contains no Access-Control-Allow-Origin for unauthorized domains.

References: OWASP WSTG-CLNT-07 (Testing Cross Origin Resource Sharing); MDN, "Cross-Origin Resource Sharing (CORS)"; PortSwigger, "Exploiting CORS misconfigurations".

5.2. CRIT-02 – PHP 7.4.33 End-of-Life

MEDIUM 5.6

SEVERITY	Medium (CVSS 5.6, editorial estimate) Revised in 1.2; was Critical 9.1
CVSS V3.1	AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:L
CWE	CWE-1104, Use of Unmaintained Third Party Components
OWASP	A06:2021 Vulnerable and Outdated Components

Description. The X-Powered-By: PHP/7.4.33 header indicates that the server uses PHP 7.4.33. PHP 7.4 officially reached End-of-Life on 28 November 2022 and has received no security updates from the PHP Group since that date. At the time of the assessment ([REDACTED]), this version had not been updated for more than three years and seven months. Any new vulnerability in PHP 7.4 discovered after the EOL date remains unpatched, creating cumulative risk: the longer an EOL version runs, the greater the likelihood of an available exploit.

Evidence (PoC):

[REDACTED]

Potential impact. An unsupported runtime increases maintenance risk. Applicability of any specific CVE depends on the exact version, operating system, SAPI, configuration and any vendor backports. No runtime RCE is established by version disclosure alone. Verify affected-version ranges and prerequisites before assigning a vulnerability-specific impact or treating this entry as confirmed exploitation.

Rating rationale (edition 1.2). The evidence is a version banner; the retained material maps no specific CVE to this build and demonstrates no exploit, and the banner may not reflect vendor backports. CVSS v3.1 scores a specific vulnerability, so the vector above is an editorial estimate for the class of post-EOL runtime flaws that could plausibly apply, used for prioritisation only: reachable over the network (AV:N), dependent on configuration, SAPI or application code paths outside the attacker's control (AC:H), with partial impact (C:L/I:L/A:L). Once an applicable CVE is confirmed, replace it with that CVE's vector. Remediation priority is unchanged: the upgrade stays in Week 1 of the roadmap, because an unsupported runtime accumulates unpatched flaws over time.

Recommendations:

- Plan an upgrade to PHP 8.3 within seven days. Check WordPress plugin and Avada theme compatibility with PHP 8.3 in a staging environment.
- If an immediate upgrade is not possible, install at least PHP 8.2 (security fixes only) for short-term protection. Edition note (2026-10): PHP 8.2 receives security fixes only until 31 December 2026 and PHP 8.3 until 31 December 2027; where the theme and plugins support it, PHP 8.4 or later gives a longer support window.
- Hide X-Powered-By in php.ini: `expose_php = Off`.
- Hide or minimize the Server header, for example using `mod_security` or `ServerTokens Prod` in Apache.
- Enable automatic alerts for new CVEs affecting the installed PHP version: follow php.net release announcements and set up NVD/CVE alerting for the installed version. (`security@php.net` is the PHP project's address for reporting vulnerabilities, not an alert feed.)

References: <https://www.php.net/supported-versions.php>; <https://www.php.net/manual/en/migration83.php>; CIS Apache Benchmark, Section 2.4 (Minimize Server Banner).

5.3. CRIT-03 – WordPress 6.0.x EOL and Outdated Plugins

CRITICAL 9.1

SEVERITY	Critical (CVSS 9.1)
CVSS V3.1	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
CWE	CWE-1104, Use of Unmaintained Third Party Components
OWASP	A06:2021 Vulnerable and Outdated Components

Description. The source identifies WordPress 6.0.12, Slider Revolution 6.5.11 and Yoast SEO 20.6 from public metadata. Confirm the installed versions and applicable vendor advisories. Version age alone does not establish an exploitable vulnerability or the exact support status; this sample does not assert a WordPress branch end-of-support date.

Evidence (PoC):

[REDACTED]

Potential impact. Components missing applicable security fixes may expose the site to known attacks. Map each installed component to a verified affected-version range and its exploitation prerequisites. The redacted evidence does not establish a specific plugin RCE or core compromise. The retained vector is a source estimate, not independent confirmation of those consequences.

Edition 1.2 note. This rating is retained as supplied by the source. Unlike CRIT-02, the source names specific component versions (WordPress 6.0.12, Slider Revolution 6.5.11, Yoast SEO 20.6) whose advisories still have to be mapped. If no applicable vulnerability exploitable without authentication is confirmed, the entry should be re-rated on the same basis as CRIT-02.

Recommendations:

- Before updating, create a complete backup of the site (files and database) and test updates in staging.
- Update WordPress to the latest stable 6.5+ version through wp-admin > Updates or WP-CLI: wp core update.
- Update every plugin to its latest version, particularly Slider Revolution and Yoast SEO. Check installed plugins through directory listing (HIGH-04); the actual list may exceed what is visible on the homepage.
- Remove the generator meta tag by adding `remove_action("wp_head", "wp_generator");` to the theme's functions.php.
- Disable XML-RPC entirely if no mobile application uses it: `add_filter("xmlrpc_enabled", "__return_false");`. The xmlrpc.php endpoint is already partially blocked by the IONOS WAF, but not completely.
- Subscribe to WordPress security announcements at wordpress.org/news/category/security/ and enable automatic notifications of new versions.
- Install an update-management plugin, such as WPVulnerability, to scan automatically for known vulnerabilities in installed components.

References: <https://wordpress.org/news/category/security/>; <https://wpscan.com/> (WordPress vulnerability database); <https://www.sliderrevolution.com/changelog/>; CWE-1104, Use of Unmaintained Third Party Components.

6. High-Severity Findings

Grouped by original rating. HIGH-04 has been rated Medium since revision 1.1; revised in edition 1.2: HIGH-02 is now Low (hardening).

6.1. HIGH-01 – User Enumeration via REST API

HIGH 7.5

SEVERITY	High (CVSS 7.5)
CVSS V3.1	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
CWE	CWE-200, Information Exposure
OWASP	A01:2021 / A08:2021

Description. The source reports two account records exposed through `/wp-json/wp/v2/users` without authentication. Public author metadata can be intentional in WordPress; determine whether the response includes sensitive data beyond the intended public profile. User ID 1 does not by itself prove an administrator role. Roles and the completeness of the returned account list are not established in this sample.

Evidence (PoC):

[REDACTED]

Potential impact. Exposed account metadata may assist reconnaissance and targeted login attempts. The actual impact depends on whether private information is returned and on authentication controls. Gaps in user IDs do not establish account history, and the sample does not verify whether 2FA is enabled. No account compromise is demonstrated here.

Recommendation. Disable the `/wp-json/wp/v2/users` REST API endpoint for unauthenticated requests. Add the following to `functions.php` or a must-use plugin. It disables ALL unauthenticated REST API requests, if they are not needed publicly:

```
add_filter("rest_authentication_errors", function($result) {
    if (!is_user_logged_in()) return new WP_Error(
        "rest_forbidden", "Authentication required", ["status"=>401]
    );
    return $result;
});
```

- Alternatively, disable only the users endpoint:

```
add_filter("rest_endpoints", function($endpoints) {
    if (is_user_logged_in()) return $endpoints; // keep the editor's author list working
    if (isset($endpoints["/wp/v2/users"]))
        unset($endpoints["/wp/v2/users"]);
    if (isset($endpoints["/wp/v2/users/(?P<id>[\d]+)"]))
        unset($endpoints["/wp/v2/users/(?P<id>[\d]+)"]);
    return $endpoints;
});
```

- Change the administrator login from [REDACTED] to a nonstandard value that differs from the display name, such as a random 12-character string.
- Enable 2FA for all administrative accounts through Wordfence, Two Factor or Google Authenticator.

- Limit login attempts using Limit Login Attempts or an IONOS-level WAF rule: five attempts per 15 minutes, with a one-hour IP block.

References: OWASP WSTG-IDNT-04 (Testing for User Enumeration); WordPress Codex, "Hardening WordPress"; <https://developer.wordpress.org/rest-api/reference/users/>.

6.2. HIGH-02 – Missing Security Headers (Hardening)

LOW

SEVERITY	Low, hardening Revised in 1.2; was High 8.2
CVSS V3.1	Not CVSS-scored: a missing protection, not a vulnerability on its own
CWE	CWE-693, Protection Mechanism Failure
OWASP	A05:2021 Security Misconfiguration

Description. The homepage and most other pages lack all key security headers. Their absence removes browser-side defence-in-depth against XSS, clickjacking, MIME sniffing, SSL stripping and Referer leakage; it does not by itself create any of these flaws, and none of them was demonstrated in this assessment. Only wp-login.php returns X-Frame-Options: SAMEORIGIN, preventing clickjacking on the login form but not elsewhere on the site.

Evidence (PoC):

[REDACTED]

Analysis of missing headers:

Header	Purpose	Risk when absent
Strict-Transport-Security	Enforce HTTPS; protect against SSL stripping	SSL stripping on public Wi-Fi
Content-Security-Policy	Allowlist script/style sources	Less mitigation if an XSS flaw exists
X-Frame-Options	Prevent iframe embedding	Clickjacking on any page except wp-login.php
X-Content-Type-Options	nosniff prevents MIME sniffing	Uploaded files may be interpreted as scripts
Referrer-Policy	Control Referer leakage	Leakage of URLs and session parameters
Permissions-Policy	Control browser API access	Malicious scripts may use camera/microphone/geolocation

Rating rationale (edition 1.2). The edition 1.1 vector (C:H, no attacker position, no user interaction) implied that any remote attacker could read high-value data directly, which missing headers do not allow. Two effects are direct. Without HSTS, any visit that starts over plain HTTP (a typed address or an old link) can be downgraded by an attacker on the network path, for example on public Wi-Fi; the HTTP-to-HTTPS redirect does not prevent this, because the redirect itself travels over HTTP. Without X-Frame-Options, pages other than wp-login.php can be framed, which matters only where a framed page offers a state-changing action. Both require an attacker position or user interaction; the remaining headers only limit the impact of other flaws, none of which was found in scope. The entry is therefore rated as hardening work (Low) rather than as a vulnerability with a CVSS score.

Recommendation. Add the headers through .htaccess (Apache) or nginx configuration. Example for .htaccess:

[REDACTED]

After implementation, check [https://securityheaders.com/?\[REDACTED\]](https://securityheaders.com/?[REDACTED]) (expected grade: A+). CSP requires careful tuning for the Avada theme's scripts and plugins. Start in Report-Only mode to collect violations, then switch to enforcement.

6.3. HIGH-03 – Email Security Misconfiguration

HIGH 8.1

SEVERITY	High (CVSS 8.1)
CVSS V3.1	AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:N
CWE	CWE-16, Configuration
OWASP	A05:2021 Security Misconfiguration

Description. The domain [REDACTED] uses Google Workspace for corporate email (MX points to aspmx.l.google.com), but lacks critical DNS records for protection against phishing and sender spoofing. SPF uses soft fail (~all instead of -all); DKIM is not configured for the google selector; DMARC is entirely absent; and MTA-STS is missing. CAA records restricting TLS certificate issuance for the domain are also absent.

Evidence (PoC):

[REDACTED]

Impact. An attacker can send phishing messages in the name of any [REDACTED] address, for example info[REDACTED]. Since SPF uses ~all (soft fail), most email clients and servers will accept spoofed messages: soft fail marks them as spam but does not reject them.

Without DMARC, recipients receive no instructions for handling messages that fail SPF/DKIM. Missing MTA-STS permits a TLS downgrade MITM attack on the email channel. Missing CAA records do not by themselves allow anyone to obtain a certificate for [REDACTED]: a CA must still validate domain control (see section 12). CAA narrows which CAs may issue, which limits exposure to a mis-issuing or compromised CA account.

Attack scenario. An attacker registers the domain [REDACTED] and sends a message in the name of [REDACTED] to an employee of [REDACTED], asking them to "update the WordPress administrator password" and linking to a fake wp-login.[REDACTED] page. Without DMARC, SPF/DKIM checks fail, but the receiving server accepts the message because SPF uses soft fail. The employee enters their password, and the attacker gains administrative access to the site.

Recommendations:

- SPF: change ~all to -all (hard fail). First ensure that SPF includes every legitimate sender.
- DKIM: enable signing in Google Workspace (Admin > Apps > Gmail > Authenticate email), generate a 2048-bit key and publish the TXT record for google._domainkey.[REDACTED].
- DMARC: start with monitoring: v=DMARC1; p=none; rua=mailto:dmarc-reports@[REDACTED]; ruf=mailto:dmarc-forensics@[REDACTED]; adkim=s; aspf=s;. After two to four weeks of report analysis, switch to p=quarantine, then to p=reject after another two weeks.
- MTA-STS: publish _mta-sts.[REDACTED] IN TXT "v=STSv1; id=[REDACTED]" and host an enforce-mode policy at .well-known/mta-sts.txt.
- CAA: add records for permitted CAs: [REDACTED] IN CAA 0 issue "sectigo.com"; [REDACTED] IN CAA 0 issue "letsencrypt.org"; [REDACTED] IN CAA 0 iodef "mailto:hostmaster@[REDACTED]".
- TLS-RPT: add _smtp._tls.[REDACTED] IN TXT "v=TLSRPTv1; rua=mailto:tls-rpt@[REDACTED]".

References: RFC 7208 (SPF), RFC 6376 (DKIM), RFC 7489 (DMARC), RFC 8461 (MTA-STS), RFC 8659 (CAA), RFC 8460 (TLS-RPT); <https://dmarc.org/>; <https://sslmate.com/certspotter/> (CAA monitoring).

6.4. HIGH-04 – Directory Listing in wp-content

MEDIUM 5.3

SEVERITY	Medium (CVSS 5.3)
CVSS V3.1	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
CWE	CWE-538, File and Directory Information Exposure
OWASP	A05:2021 Security Misconfiguration

Description. Apache returns HTTP 200 with directory listings for /wp-content/plugins/ and /wp-content/themes/. This lets an attacker obtain the full list of installed plugins and themes without analyzing the homepage. The /wp-content/uploads/ directory returns 403 (protected), showing inconsistent configuration.

Evidence (PoC):

[REDACTED]

Impact. An attacker obtains: (1) the exact list of installed plugins and their directory names; (2) direct access to plugin files such as readme.txt and changelog.txt, which often contain precise versions; and (3) the theme list, facilitating searches for vulnerable versions. This substantially accelerates reconnaissance during a real attack and lets automated bots immediately identify targets with known CVEs.

Recommendations:

- Disable directory listing globally in .htaccess: Options -Indexes.
- Deny direct access to sensitive files (readme.txt, changelog.txt and .md): Require all denied.
- Deny direct access to /wp-content/plugins/ and /wp-content/themes/ as a whole, except static resources (.css, .js, .png, .jpg, .gif, .svg, .woff and .woff2): Require all granted / Require all denied.
- Consider Wordfence or iThemes Security, which automatically hide WordPress and plugin fingerprints.

References: OWASP WSTG-CONF-05 (Test File Permission); CWE-538; WordPress Codex, "Hardening WordPress", File Permissions section.

7. Medium-Severity Findings

Grouped by original rating. MED-03 and MED-04 have been rated Low since revision 1.1; revised in edition 1.2: MED-05 is now Informational.

7.1. MED-01 – Missing CAA DNS Records

MEDIUM 6.5

SEVERITY	Medium (CVSS 6.5)
CVSS V3.1	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
CWE	CWE-295, Improper Certificate Validation

Description. DNS lacks CAA records (Certification Authority Authorization, RFC 8659). CAA lets a domain explicitly specify which certificate authorities may issue certificates for it. Without CAA, any publicly trusted CA, of which there are dozens, can issue a certificate for [REDACTED] if an account at that CA is compromised or social engineering succeeds.

Recommendation. Add CAA records restricting issuance to the authorities actually used: Sectigo for the wildcard certificate and, optionally, Let's Encrypt for future certificates. Configuration: [REDACTED]. Verify through <https://caatestsuite.com/> or ssllabs.com.

Edition 1.2 note: this rating is retained as supplied by the source. Sections 6.3 and 12 note that missing CAA records do not by themselves allow certificate issuance; the rating should be reviewed on that basis.

7.2. MED-02 – Accessible readme.html, license.txt, wp-cron.php

MEDIUM 5.3

SEVERITY	Medium (CVSS 5.3)
CVSS V3.1	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
CWE	CWE-200, Information Exposure

Description. Standard WordPress files are accessible through direct URLs: /readme.html (HTTP 200; exposes the WordPress version), /license.txt (HTTP 200; GPLv2 license text) and /wp-cron.php (HTTP 200; potentially usable for DoS). These files are included with WordPress and are not required for the site's operation.

[REDACTED]

Recommendation. Delete readme.html and license.txt from the server or block them in .htaccess with Require all denied. Block external access to wp-cron.php and move execution to system cron: add `define("DISABLE_WP_CRON", true);` to wp-config.php, then configure system cron to run every 15 minutes:

```
*/15 * * * * curl -s https://[REDACTED]/wp-cron.php?doing_wp_cron >/dev/null 2>&1
```

Block direct access in .htaccess using Require local (requires configuration for the cron server's IP). Note that Require local also blocks the curl request above when it goes through the public address; in that case run cron through WP-CLI instead (`wp cron event run --due-now`).

7.3. MED-03 – Missing OCSP Stapling

LOW 3.7

SEVERITY	Low (CVSS 3.7)
CVSS V3.1	AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
CWE	CWE-299, Improper Check for Certificate Revocation

Description. During the TLS handshake, the server does not return an OCSP response ("OCSP response: no response sent"). Without OCSP stapling, the client must contact the CA's OCSP server to check revocation status. This (1) increases handshake time, (2) reveals visits to [REDACTED] to the OCSP server, affecting privacy, and (3) prevents revocation checking when that server is unavailable.

Recommendation. Enable OCSP stapling in Apache: `SSLUseStapling On` and `SSLStaplingCache shmcb:/var/run/ocsp(128000)`. Ensure the certificate contains Authority Information Access (AIA) with an OCSP URL. Check using:

```
echo | openssl s_client -connect [REDACTED]:443 -status 2>/dev/null | grep "OCSP Response Status"
```

Expected result: OCSP Response Status: successful.

7.4. MED-04 – Wildcard Certificate *. [REDACTED]

LOW 3.8

SEVERITY	Low (CVSS 3.8)
CVSS V3.1	AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:N
CWE	CWE-295, Improper Certificate Validation

Description. The server uses a wildcard certificate for *. [REDACTED], issued [REDACTED]. It is valid for any subdomain one level below the domain ([REDACTED], and so on). This is convenient for administration, but creates risk: if the private key is compromised, an attacker can impersonate any subdomain. The certificate also covers future subdomains that may fall outside the owner's control, for example through subdomain takeover.

Recommendation. Consider separate certificates for each subdomain using Let's Encrypt and automated ACME issuance. If a wildcard certificate is necessary for convenience, strictly control the private key (HSM and key rotation every 12 months), monitor CT logs (crt.sh and CertSpotter) for unauthorized issuance, and add CAA records (MED-01).

7.5. MED-05 – .git/HEAD Returns 403

INFO

SEVERITY	Info (not scored) Revised in 1.2; was Medium 5.3
CVSS V3.1	Not scored: no content was disclosed
CWE	CWE-538, File and Directory Information Exposure (applies only if a repository is confirmed in the web root)

Description. A request to `/.git/HEAD` returned HTTP 403 rather than 404. This response alone does not prove that a repository exists: a blanket web-server or WAF rule may produce the same result. No source-code disclosure is demonstrated in this sample. Confirm repository presence and access controls with the system

owner before treating the observation as a vulnerability.

[REDACTED]

Rating rationale (edition 1.2). HTTP 403 means the server refused to return the file, so no repository content was exposed; the edition 1.1 vector assumed a confidentiality impact (C:L) that did not occur. The only information gained is that the path may exist, which a blanket deny rule would also produce. The entry is kept as an informational verification item. If a repository is found in the document root, assess it separately, because other files may be served even when HEAD is blocked.

Recommendation. Check for .git in the document root using SFTP/SSH. If present, delete it or move it outside the web root. If it is required for deployment, add this to .htaccess:

```
RedirectMatch 404 /\.(git|svn|hg|env|DS_Store)
```

This returns 404 for sensitive dotfiles and dot-directories, hiding their existence.

8. Low-Severity Findings and Informational Notes

8.1. LOW-01 – Missing /.well-known/security.txt

LOW 2.0

SEVERITY	Low (source estimate 2.0)
CVSS V3.1	No vector supplied (source priority estimate)
CWE	CWE-200

The security.txt file is defined by RFC 9116 for publishing vulnerability reporting contacts. A request to /.well-known/security.txt returns a WordPress 404 page. Independent researchers therefore have no official reporting channel, which may lead to uncoordinated publication. Create security.txt with a contact email, such as security@[REDACTED], preferably with PGP encryption support, and a responsible disclosure policy. See <https://securitytxt.org/>.

8.2. LOW-02 – Missing BIMl Record

LOW 1.5

SEVERITY	Low (source estimate 1.5)
CVSS V3.1	No vector supplied (source priority estimate)

BIMI (Brand Indicators for Message Identification) is a DNS record that lets email clients display the brand's logo in the inbox. Although not strictly a security finding, BIMI requires DMARC quarantine/reject, encouraging DMARC adoption. After configuring DMARC (HIGH-03), add a BIMI record:

```
[REDACTED] IN TXT "v=BIMI1; l=https://[REDACTED]/logo.svg; a=https://[REDACTED]/auth.pem"
```

8.3. LOW-03 – RSS Feed Generates Fingerprinting Patterns

LOW 2.7

SEVERITY	Low (source estimate 2.7)
CVSS V3.1	No vector supplied (source priority estimate)
CWE	CWE-200

The /feed/ RSS feed exposes additional metadata: exact publication dates (facilitating timing attacks), author lists and full image URLs revealing the uploads directory structure. If subscribers do not use RSS, disable it:

```
foreach (["do_feed_rdf", "do_feed_rss", "do_feed_rss2", "do_feed_atom",  
         "do_feed_rss2_comments", "do_feed_atom_comments"] as $feed) {  
    add_action($feed, function() { wp_die("No feed available"); }, 1);  
}
```

If RSS is used, retain it but ensure posts do not leak sensitive URL parameters.

8.4. INFO-01 – Google Site Verification in TXT

INFO

SEVERITY

Info

A google-site-verification=... TXT record is normal for domains using Google Workspace or Google Search Console. This is not a vulnerability; it is included as an informational note for completeness.

8.5. Positive Findings (What Is Configured Well)

For completeness, the following configuration aspects are correct and require no changes:

Aspect	Status	Comment
HTTP to HTTPS redirect	OK	301 Moved Permanently from http://[REDACTED] to https://[REDACTED]
TLS 1.0/1.1 disabled	OK	Only TLS 1.2 and 1.3 are supported
Cipher suites (TLS 1.3)	OK	TLS_AES_256_GCM_SHA384, AEAD
IPv6 support	OK	[REDACTED]
TRACE method	OK	HTTP 405 Method Not Allowed
Server header	OK	"Apache", without version (ServerTokens Prod)
xmlrpc.php	OK*	HTTP 503 through IONOS WAF (partial protection)
wp-content/uploads/	OK	HTTP 403; directory listing disabled
.htaccess	OK	HTTP 403
wp-login.php X-Frame-Options	OK	SAMEORIGIN; clickjacking protection on the login form
Certificate validity	OK	Sectigo wildcard, valid until [REDACTED]

9. Detailed TLS/SSL Analysis

This section analyzes the target server's TLS/SSL configuration against Mozilla SSL Configuration Generator best practices (Intermediate level, compatible with browsers from 2018 onward). All checks used openssl s_client and curl, without automated scanners.

9.1. Certificate

The server uses a TLS certificate issued [REDACTED]. The wildcard certificate covers the base domain and all subdomains one level below it. Validity period: [REDACTED], approximately 16 months. It is a Domain Validated (DV) certificate, representing the minimum acceptable level of validation.

[REDACTED]

9.2. Supported Protocols and Ciphers

The server supports only TLS 1.2 and TLS 1.3; obsolete, insecure TLS 1.0 and TLS 1.1 are rejected. This meets modern Mozilla Intermediate recommendations and PCI DSS requirements. TLS 1.3 uses TLS_AES_256_GCM_SHA384, while TLS 1.2 uses ECDHE-RSA-AES128-GCM-SHA256; both provide perfect forward secrecy.

[REDACTED]

9.3. Comparison with Mozilla Intermediate Recommendations

Parameter	Mozilla Intermediate	Observed	Compliance
TLS 1.0	Disabled	Disabled	OK
TLS 1.1	Disabled	Disabled	OK
TLS 1.2	Enabled	Enabled	OK
TLS 1.3	Enabled	Enabled	OK
HSTS	max-age >= 31536000	ABSENT	FAIL (HIGH-02)
OCSP stapling	Enabled	ABSENT	FAIL (MED-03)
CAA records	Recommended	ABSENT	FAIL (MED-01)
Certificate type	ECDSA preferred	RSA (Sectigo DV)	Acceptable
Forward secrecy	Required	ECDHE: yes	OK
Certificate transparency	Embedded SCT	Presumably yes (Sectigo)	OK

9.4. testssl.sh Commands for Retesting

After remediation, a detailed assessment using testssl.sh is recommended; it is the most comprehensive open-source TLS analysis tool. Command for a low-invasive check (a single run, without DoS):

[REDACTED]

10. DNS and Email Infrastructure Analysis

This section combines analysis of the DNS configuration and email infrastructure for [REDACTED]. Since the domain uses Google Workspace for corporate email, particular attention is given to SPF, DKIM, DMARC and MTA-STS. Compromise of the email channel is one of the most likely attack vectors against a small business.

10.1. Complete DNS Record Map

[REDACTED]

10.2. RDAP/WHOIS Analysis

Domain registration information was obtained through the RDAP interface [REDACTED]. By default, [REDACTED] does not publish private individuals' registrant details, but basic information is available for corporate domains. The domain record was last modified on [REDACTED]. Status: active. This means the domain has been registered for more than six years, taking earlier updates into account, and is used consistently.

10.3. Email Protection Analysis and Phishing Attack Scenario

The current email domain configuration is vulnerable to phishing in the name of [REDACTED]. The following step-by-step attack becomes possible through the combined weaknesses of SPF soft fail, missing DKIM and missing DMARC:

- **Step 1.** The attacker selects a target email address, for example [REDACTED]. These are typical addresses published by the client in the site's legal notice.
- **Step 2.** The attacker uses an SMTP server, such as their own VPS or a compromised mail relay, to send a message with a spoofed From: address to an employee of [REDACTED].
- **Step 3.** The employee's receiving mail server checks SPF. The message arrived from the attacker's IP rather than Google Workspace (aspmx.l.google.com). SPF returns softfail because the TXT record uses ~all. The message is marked as "possibly spam" but is not rejected.
- **Step 4.** DKIM checking is not performed because the domain has no public DKIM key: the google._domainkey.[REDACTED] TXT record is missing.
- **Step 5.** DMARC checking is not performed because the _dmarc.[REDACTED] TXT record is missing. The recipient therefore receives no quarantine/reject instructions.
- **Step 6.** The message reaches the employee's inbox, or Spam if the spam filter reacts to softfail. The employee sees a message "from" [REDACTED] and may enter a password on a fake page.

Correctly deployed SPF, DKIM and DMARC enforcement reduce direct domain spoofing, subject to recipient policy and authentication alignment. They do not eliminate phishing, lookalike domains or compromised legitimate accounts. Confirm all legitimate mail streams before enforcement and include staff awareness and incident reporting controls.

10.4. Recommended Target DNS Configuration

[REDACTED]

11. WordPress Hardening Checklist

This consolidated WordPress hardening checklist incorporates all findings. It is divided into four priority groups: critical fixes (24 hours), urgent fixes (one week), scheduled fixes (one month) and recommended improvements (three months).

11.1. Critical Fixes (24 Hours)

No.	Action	Related finding
1	Remove Access-Control-Allow-Credentials: true and allowlist Origin in Apache/nginx	CRIT-01
2	Block /wp-json/wp/v2/users for unauthenticated requests	HIGH-01
3	Disable directory listing (Options -Indexes)	HIGH-04

Edition 1.2: the baseline security headers item (HIGH-02) moved to the one-week group after HIGH-02 was re-rated to Low.

11.2. Urgent Fixes (1 Week)

```
# 1. Upgrade PHP through the IONOS control panel (select PHP 8.3).
# First check plugin compatibility in staging.
# 2. Update WordPress through wp-admin > Dashboard > Updates.
# Or use WP-CLI:
wp core update
wp plugin update --all
wp theme update --all
# 3. Hide the PHP and Apache versions.
# In php.ini:
expose_php = Off
# In Apache configuration:
ServerTokens Prod
# ServerSignature Off
# 4. Remove default files:
rm -f readme.html license.txt
# 5. Install Wordfence or Sucuri for WAF and 2FA.
# Enable 2FA for every administrator account.
# 6. Add baseline security headers (HSTS, X-Frame-Options,
# X-Content-Type-Options) and start CSP in Report-Only mode (HIGH-02).
```

11.3. Scheduled Fixes (1 Month)

- Email security: deploy DKIM (Google Workspace), DMARC (start with p=none, move to p=quarantine after two to four weeks of report analysis, then to p=reject after another two weeks), MTA-STS, CAA and TLS-RPT; change SPF to -all (see HIGH-03 and MED-01).
- CSP: develop Content-Security-Policy in Report-Only mode, collect violations for one to two weeks, then enforce it (HIGH-02).
- OCSP stapling: enable it in Apache (MED-03).
- 2FA: require it for everyone with administrator/editor privileges, using Wordfence Login Security or Two Factor.
- Backup: schedule daily database backups and weekly file backups through UpdraftPlus or IONOS backup. Retain at least four backup generations.

- **Logging:** keep logs outside the web root. In production, leave WP_DEBUG off and let PHP log errors to a non-public file. For temporary debugging, enable WP_DEBUG with WP_DEBUG_LOG pointing to a non-public path (WordPress 5.1+ accepts a file path) instead of the default wp-content/debug.log, which is web-accessible and a common source of data leaks, and set WP_DEBUG_DISPLAY to false explicitly (it defaults to true once WP_DEBUG is on). Never display errors in production. If a log must stay inside the web root, deny web access to it and confirm that a request for it returns 403 or 404. Rotate logs, and send critical event notifications to the administrator's email.

```
; php.ini or .user.ini (production)
display_errors = Off
log_errors = On
error_log = /path/outside/webroot/logs/php-error.log

// wp-config.php, temporary debugging only
define('WP_DEBUG', true);
define('WP_DEBUG_LOG', '/path/outside/webroot/logs/wp-debug.log');
define('WP_DEBUG_DISPLAY', false);

# .htaccess, only if a log file must live under the web root
<FilesMatch "\.log$">
    Require all denied
</FilesMatch>
```

- **.git:** check for .git in the document root, then remove or block it (MED-05).

11.4. Recommended Improvements (3 Months)

- **Staging environment:** create a staging copy on a subdomain, such as staging.[REDACTED], protected by HTTP Basic Auth, for update testing.
- **Continuous monitoring:** subscribe to notifications about new CVEs in installed plugins through WPVulnerability or Patchstack.
- **Web Application Firewall:** consider Cloudflare or IONOS WAF with custom rules for WordPress-specific attack vectors.
- **security.txt:** create /.well-known/security.txt with contacts for responsible vulnerability disclosure (LOW-01).
- **BIMI:** after implementing DMARC p=reject, add BIMI to display the logo in email clients (LOW-02).
- **Penetration test:** after remediation, repeat the external assessment to verify fixes.
- **Incident Response Plan:** develop and document an incident response plan and conduct team exercises.

12. Subdomain Takeover Assessment

Subdomain takeover occurs when an attacker registers an expired service on a third-party platform, such as GitHub Pages, Heroku or S3, and takes control of a subdomain whose CNAME still points to that service. To assess this risk, 16 common subdomains were checked through DNS, alongside a passive Certificate Transparency log search (crt.sh).

12.1. Subdomain Check Results

[REDACTED]

Conclusion. No takeover candidates were reported among the checked subdomains. This is limited to the assessed assets and time. Missing CAA records do not let an attacker obtain a certificate without satisfying a CA's domain-control validation. CAA can restrict authorized issuers; it does not replace DNS lifecycle management or independently prevent subdomain takeover.

13. Remediation Roadmap

The following roadmap lists timeframes, owners and priorities. It is based on CVSS scores, exploitation complexity and business process criticality. Scheduling is not strictly ordered by rating: inexpensive fixes and related upgrades (PHP together with WordPress) are grouped. Before beginning, create a complete site backup (files and database) and test updates in staging.

Phase	Timeframe	Actions	Findings	Owner
Day 0 (emergency)	24 hours	Fix CORS; disable REST API users; set Options -Indexes	CRIT-01, HIGH-01, HIGH-04	Web administrator
Week 1	7 days	Upgrade PHP to 8.3 and WordPress to 6.5+; update plugins; hide X-Powered-By; add baseline security headers and start CSP in Report-Only	CRIT-02, CRIT-03, HIGH-02	Web administrator + hosting provider
Week 2	14 days	Set up DKIM; DMARC p=none; MTA-STS; CAA; SPF -all; TLS-RPT	HIGH-03, MED-01	IT administrator
Week 3	21 days	Move CSP from Report-Only to enforcement; OCSP stapling; remove readme/license; disable wp-cron	HIGH-02 (CSP), MED-02, MED-03	Web administrator
Week 4	28 days	Move DMARC from p=none to p=quarantine; administrator 2FA; WAF; monitoring; .git verification (remove if present); security.txt	HIGH-03 (DMARC), MED-05, LOW-01	IT administrator
Week 6	42 days	Move DMARC from p=quarantine to p=reject; BIMi; HSTS preload	HIGH-03 (DMARC), LOW-02, HIGH-02 (HSTS)	IT administrator
Week 8	56 days	Repeat external penetration test to verify remediation	All	External penetration tester

Quality control. After each phase, repeat the relevant commands from Appendix A for verification. Perform a full retest in Week 8 to confirm that all findings are resolved and that no new regressions have appeared. Issue a separate retest report confirming each finding's status: Closed, Risk Accepted or False Positive.

Appendix A. Audit Reproduction Commands

Complete list of commands used during the assessment. All commands are passive or low-invasive (one request per command). They can be run sequentially for a repeat assessment and are grouped by check category.

A.1. DNS Reconnaissance

[REDACTED]

A.2. TLS/SSL Analysis

[REDACTED]

A.3. HTTP Header Audit

[REDACTED]

A.4. WordPress Fingerprinting

[REDACTED]

A.5. CORS Verification

[REDACTED]

A.6. Sensitive Path Probe

[REDACTED]

A.7. Subdomain Enumeration

[REDACTED]

Appendix B. Recommended Tools for Future Assessments

Toolchain for a future, deeper assessment, subject to an expanded scope. All tools are open source except WPScan, which requires an API key for its extended CVE database, and Shodan/Censys, which require API registration.

B.1. Passive OSINT

[REDACTED]

B.2. Active Scanning (Low-Invasive)

[REDACTED]

B.3. WordPress-Specific Tools

[REDACTED]

B.4. Directory Fuzzing (Requires an Expanded Scope)

[REDACTED]

Appendix C. References and Standards

Normative documents, standards and external references used to prepare the report. All references were checked for currency on the report's preparation date ([REDACTED]).

C.1. Standards and Methodologies

- OWASP Web Security Testing Guide (WSTG) v4.2: <https://owasp.org/www-project-web-security-testing-guide/>
- OWASP Top 10 2021: <https://owasp.org/Top10/>
- PTES (Penetration Testing Execution Standard): <http://pentest-standard.org/>
- CIS Benchmarks (Apache, PHP, WordPress): <https://www.cisecurity.org/cis-benchmarks/>
- Mozilla SSL Configuration Generator: <https://ssl-config.mozilla.org/>
- NIST SP 800-115: Technical Guide to Information Security Testing and Assessment.
- Federal Office for Information Security (BSI), TR-O3162: Penetration Testing Best Practices.

C.2. CWE References

CWE	Name	URL
CWE-942	Permissive Cross-domain Policy with Untrusted Domains	https://cwe.mitre.org/data/definitions/942.html
CWE-1104	Use of Unmaintained Third Party Components	https://cwe.mitre.org/data/definitions/1104.html
CWE-200	Information Exposure	https://cwe.mitre.org/data/definitions/200.html
CWE-693	Protection Mechanism Failure	https://cwe.mitre.org/data/definitions/693.html
CWE-16	Configuration	https://cwe.mitre.org/data/definitions/16.html
CWE-538	File and Directory Information Exposure	https://cwe.mitre.org/data/definitions/538.html
CWE-295	Improper Certificate Validation	https://cwe.mitre.org/data/definitions/295.html
CWE-299	Improper Check for Certificate Revocation	https://cwe.mitre.org/data/definitions/299.html

C.3. RFCs

- RFC 7208: Sender Policy Framework (SPF).
- RFC 6376: DomainKeys Identified Mail (DKIM).
- RFC 7489: Domain-based Message Authentication, Reporting, and Conformance (DMARC).
- RFC 8461: SMTP MTA Strict Transport Security (MTA-STS).
- RFC 8659: DNS Certification Authority Authorization (CAA).
- RFC 8460: SMTP TLS Reporting (TLS-RPT).
- RFC 9116: A File Format to Aid in Security Vulnerability Disclosure (security.txt).
- RFC 7919: Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for TLS.

C.4. Legal Acts ([REDACTED])

[REDACTED]

C.5. Tools and Resources

- WPScan: <https://wpscan.com/> (WordPress vulnerability database).
- testssl.sh: <https://github.com/drwetter/testssl.sh>
- Nmap: <https://nmap.org/>
- Nikto: <https://cirt.net/Nikto2>
- Subfinder: <https://github.com/projectdiscovery/subfinder>
- Amass: <https://github.com/owasp-amass/amass>
- ffuf: <https://github.com/ffuf/ffuf>
- dnsx / httpx: <https://github.com/projectdiscovery/dnsx>
- securityheaders.com: <https://securityheaders.com/>
- sslabs.com SSL Test: <https://www.ssllabs.com/ssltest/>
- crt.sh: <https://crt.sh/> (Certificate Transparency logs).
- Shodan: <https://www.shodan.io/>
- Censys: <https://censys.io/>
- securitytxt.org: <https://securitytxt.org/>
- HSTS Preload List: <https://hstspreload.org/>

End of report.



synt world

YOUR TURN

Get your own assessment

Pay only for confirmed vulnerabilities.

This sample shows the format of a Synt World report: scope and rules of engagement, rated findings with the evidence they rest on (redacted in this sample) and their caveats, and a remediation roadmap.

PRICING

<https://synt.world/pricing.html>

CONTACT

ceo@synt.world