WordPress

SECURITY AUDIT

Prepared For:
Example Company

Prepared By:
Webolicious Security Team

WordPress Security Audit Report
Website: example.com
Audit Date: 01/01/2025
Conducted By: Webolicious Security Team

1. Executive Summary

The security audit of example.com identified several vulnerabilities that could compromise website security. Key findings include outdated plugins, weak user authentication, exposed sensitive information, and misconfigurations. Immediate remediation is recommended to prevent potential security breaches.

2. Critical Issues (High Priority)

2.1 Outdated & Vulnerable Plugins

The following plugins are outdated or have known vulnerabilities:

Plugin Name	Installed Version	Latest Version	Risk Level	Vulnerability Details
Contact Form 7	5.4.1	5.8	High	XSS vulnerability (CVE-XXXX-XXXX)
Elementor	3.2.3	3.19.2	High	RCE vulnerability (CVE-XXXX-XXXX)
WP File Manager	6.9	7.1	Critical	Unauthenticated File Upload

Recommendation: Immediately update all vulnerable plugins and remove any unused or abandoned plugins.

2.2 Weak WordPress Core Version

The site is running WordPress **5.7.2**, which has known vulnerabilities. The latest version is **6.4.2**.

Recommendation: Update to the latest WordPress version and enable automatic updates.

3. User Security Risks

3.1 Exposed Usernames

The following usernames were found exposed via author archives, REST API, or login pages:

- admin
- editor
- johndoe

Risk: Attackers can use these usernames for brute-force attacks.

Recommendation: Hide author archives, disable user enumeration, and enforce strong passwords.

3.2 Weak Passwords Detected

Some user accounts have weak passwords or use common passwords.

Recommendation: Implement two-factor authentication (2FA) and enforce strong password policies.

3.3 Unused or Dormant Admin Accounts

The following administrator accounts have not been used in over 90 days:

- oldadmin
- testuser

Recommendation: Remove or disable unused admin accounts.

4. Code & Configuration Vulnerabilities

4.1 Insecure Themes & Custom Code

A manual review of theme files and custom code identified:

- **Hardcoded API keys** in functions.php
- **Direct database queries** without sanitization, leading to SQL Injection risks
- **Use of outdated and vulnerable third-party libraries** in custom theme files
- **Excessive use of inline JavaScript**, increasing the risk of XSS attacks
- **Unvalidated user input** in custom form handlers
- **Unprotected AJAX endpoints**, allowing unauthorized data manipulation
- **Hardcoded admin credentials** in theme or plugin files
- **Use of base64 encoding/decoding**, which is often a red flag for obfuscated malicious code

Recommendation:

- Remove hardcoded sensitive data and store API keys securely.
- Use WordPress functions like prepare() for database queries to prevent SQL injection.

- Regularly update and audit third-party libraries used in themes.
- Sanitize and validate all user inputs before processing.
- Restrict access to AJAX endpoints to authenticated users where applicable.
- Review and remove any unnecessary or suspicious use of base64 encoding.

4.2 Debug Mode Enabled

WordPress debug mode (WP_DEBUG) is enabled in wp-config.php, potentially exposing sensitive information.

Recommendation: Disable debug mode on production sites (define('WP_DEBUG', false);).

4.3 Insecure File Permissions

Detected files with incorrect permissions:

- wp-config.php (should be 600 but is 644)
- /uploads/ directory is writable by all (777 instead of 755)

Recommendation: Adjust file permissions to secure sensitive files.

4.4 Directory Traversal & Sensitive File Exposure

The following files were found publicly accessible:

- /wp-config.php.bak
- /debug.log
- /backup.zip

Recommendation: Remove or restrict access to these files.

5. Server & Hosting Security

5.1 Outdated PHP Version

The site is running PHP **7.4**, which is no longer receiving security updates.

Recommendation: Upgrade to PHP **8.2** for improved security and performance.

5.2 Directory Listing Enabled

Certain directories (/uploads, /wp-includes/) are publicly accessible, exposing files.

Recommendation: Disable directory listing via .htaccess or server configuration.

5.3 Unprotected wp-config.php

wp-config.php is accessible from the web, exposing sensitive database credentials.

Recommendation: Move wp-config.php outside the root directory and block access using .htaccess.

5.4 Unnecessary Services & Ports Open

Detected open ports that increase attack surface:

- FTP (Port 21) - Should be disabled or replaced with SFTP
- XML-RPC (Enabled) - Known to be vulnerable to brute force attacks

Recommendation: Disable unused services and restrict access where possible.

6. Malware & Suspicious Activity

6.1 Suspicious File Modifications

Detected unauthorized changes in:

- wp-includes/functions.php
- index.php

Recommendation: Review file changes, scan for malware, and restore clean backups if necessary.

6.2 Blacklisted by Security Services

The website is blacklisted by Google Safe Browsing.

Recommendation: Remove malware, request a review, and resubmit the site to Google.

7. Security Recommendations & Next Steps

Immediate Actions:
✔ Update all plugins, themes, and WordPress core
✔ Hide usernames and enforce strong passwords
✔ Secure custom code and disable debug mode
✔ Upgrade to PHP 8.2 and disable directory listing
✔ Remove exposed sensitive files
✔ Scan for malware and restore clean backups
✔ Restrict access to wp-config.php

Long-Term Security Measures:
🔹 Implement a Web Application Firewall (WAF)
🔹 Schedule regular security audits & malware scans
🔹 Enable automated backups with offsite storage
🔹 Monitor for unauthorized logins and activity
🔹 Implement Content Security Policy (CSP) to mitigate XSS attacks

Final Assessment

The website has **critical security risks** that require immediate attention. Addressing these vulnerabilities will significantly improve the site's security posture and protect against potential threats.

Report Generated By: Webolicious Security Team
Contact Us: T. 01245 377 635, E. hello@webolicious.co.uk