CHECKLIST · SECURITY AUDIT

The WordPress security audit checklist.

This is the working outline of what a professional WordPress security audit actually reviews, published as a checklist you can run against your own site. It will not replace an audit — a checklist records questions, an audit verifies answers with evidence — but working through it honestly will surface the gaps that matter most, and it is free to read and download with no email, form, or account.

Prepared by G. Schad, WordPress and Linux security specialist

Published · Last reviewed

FORMAT Free checklist — this page and a printable PDF	SCOPE Five control areas, from access to monitoring and response
SOURCE The methodology used in real audit engagements	HONEST LIMIT A checklist asks the questions; an audit verifies the answers

---

HOW TO USE IT

Answer with evidence, not with memory

For each item, the standard is not “I think so” but “I checked, and here is what I saw”. Most items take a few minutes to verify; the ones you cannot verify are findings in themselves, because they usually mean the access or the records do not exist.

Items involving the server side may need your hosting provider’s help on shared plans. If the host cannot answer them either, that too is worth knowing before an incident makes it urgent.

SCOPE NOTE

The checklist covers the assessment questions, not the fixes. Where an answer is “no” or “unknown”, resist quick config changes on production — understand the dependency first, or hand the finding to whoever operates the system.

AREA 1

Access and identity

- ✓ Every administrator account maps to a named, current person — none are shared, none belong to former staff or agencies
- ✓ The user list in the admin panel matches the users table in the database, with no unexplained accounts at any role
- ✓ All administrators use strong, unique passwords and two-factor authentication where available
- ✓ Application passwords, API keys, and REST access are inventoried and revoked where unused
- ✓ Login endpoints have brute-force protection: rate limiting, lockouts, or an access restriction in front of wp-login.php
- ✓ Hosting-panel, SFTP/SSH, and database credentials are unique, current, and held only by people who need them
- ✓ Self-registration is disabled, or the default role for new registrations is the lowest available

---

AREA 2

Software and configuration

- ✓ WordPress core, every plugin, and every theme are current — and every installed plugin and theme is actually in use
- ✓ No abandoned components: everything installed has received a vendor update within a reasonable period
- ✓ File editing from the dashboard is disabled (DISALLOW_FILE_EDIT), and debug output is off in production
- ✓ Secret keys and salts in wp-config.php are set, unique, and rotated after any suspected compromise
- ✓ XML-RPC is disabled or restricted unless something genuinely depends on it
- ✓ The REST API does not expose user enumeration to anonymous visitors
- ✓ Comments, forms, and uploads are restricted to what the site actually needs

AREA 3

Hosting and server

- ✓ PHP is a supported version, and dangerous functions and settings are not enabled beyond what the site requires
- ✓ TLS covers the whole site with a valid certificate, and HTTP redirects to HTTPS everywhere
- ✓ File and directory permissions follow least privilege — no world-writable paths, and wp-config.php is not readable over the web
- ✓ Each site on the server runs in its own account or isolation boundary, so one compromise cannot spread to neighbors
- ✓ SSH access uses keys, not passwords, and administrative services are not open to the whole internet

- ✓ Server and application error messages do not disclose paths, versions, or credentials to visitors
- ✓ Web-server and PHP logs exist, cover enough history to investigate an incident, and someone can actually access them

AREA 4

Recovery readiness

- ✓ Backups run automatically, cover both files and the database, and are stored somewhere the web server cannot overwrite or delete
- ✓ A restore has actually been tested — recently enough that the result is still believable
- ✓ Backup retention reaches back further than the longest plausible undetected compromise, not just a few days
- ✓ Access needed for recovery — hosting, DNS, backups, licenses — is documented and reachable by more than one person
- ✓ The site can be rebuilt from trusted sources: original plugin/theme licenses and any custom code are in version control or safe storage

AREA 5

Monitoring and response

- ✓ Something watches file integrity or unexpected changes, and its alerts go somewhere a human actually reads
- ✓ Administrative actions are logged with enough detail to answer “who changed what, when” after the fact

- ✓ Search Console is claimed by the owner, so security flags and indexing anomalies reach you and not only an attacker
- ✓ Uptime and certificate expiry are monitored externally
- ✓ It is decided, in writing, who responds when the site is compromised at 2 a.m. — owner, developer, host, or a retained specialist

THE HONEST LIMIT

What a checklist cannot tell you

A checklist verifies presence, not adequacy. It can confirm that backups run; it cannot judge whether the restore would survive a real incident, whether the WAF rule actually blocks the exploit it names, or whether the custom checkout logic leaks what it should protect. Those answers come from evidence: configuration review, log analysis, and controlled testing against the specific system.

That is the difference between this page and the audit engagement — and it is why the audit produces a findings report with evidence, business impact, and retested fixes rather than a column of checkmarks. If the checklist has surfaced more “unknown” than “verified”, that is the honest signal to scope one.

What the full audit engagement covers →

Read a sanitized sample audit report →

COMMON QUESTIONS

Common questions about the checklist

Is this the complete audit methodology? –

It is the honest outline of the control areas. The engagement itself goes deeper in each area, verifies answers with evidence rather than self-assessment, covers WooCommerce and custom functionality where present, and ends in a prioritized findings report with retesting.

How often should I run through it? +

I cannot verify the server items on shared hosting. What now? +

If I can tick everything, do I still need an audit? +