WPSecureStack

wp-config.php Hardening Snippets

wpsecurestack.com

9 ready-to-paste PHP snippets for hardening your WordPress configuration file. Each snippet includes a plain-English explanation, a warning where relevant, and a practical tip. Replace every value marked ← **REPLACE** before saving. Always back up your wp-config.php before making any changes.

HOW TO USE THESE SNIPPETS

Step 1	Download a backup of your current wp-config.php before touching anything.
Step 2	Open wp-config.php via FTP, SFTP, or your hosting file manager.
Step 3	Copy the snippets you want from this guide.
Step 4	Paste them ABOVE the line that says: /* That's all, stop editing! Happy publishing. */
Step 5	Replace every value marked ← REPLACE with your real values.
Step 6	Save the file and immediately test your site in a browser.
Step 7	If anything breaks, restore your backup and contact us.

■■ DISCLAIMER

1. Your Responsibility	You are solely responsible for reviewing, testing, and adapting these settings to your specific hosting environment and WordPress setup before deploying to any live site.
2. No Guarantee of Security	Applying these settings does not guarantee your website will be free from vulnerabilities or breaches. No configuration file alone can provide complete security.
3. Always Backup First	WPSecureStack accepts no responsibility for downtime, data loss, or broken functionality caused by applying these settings without first taking a full backup.
4. Replace All Placeholders	This file contains placeholder values marked ← REPLACE. Deploying placeholder values to a live site leaves your site misconfigured. WPSecureStack is not liable for any consequence of deploying placeholder values.
5. Server Compatibility	Some settings depend on your server type (Apache, Nginx, LiteSpeed), PHP version, and hosting configuration. Test on a staging site before applying to production.
6. Not Legal or Compliance Advice	Nothing in this file constitutes legal, regulatory, or compliance advice. If your site handles personal data or operates in a regulated industry, consult a qualified professional.
7. Third-Party Services	References to third-party tools (Wordfence, Sucuri, Cloudflare, etc.) are informational only. WPSecureStack has no affiliation with these services and does not endorse them.
8. No Liability	WPSecureStack, Emmanuel Tatyabala, and any affiliated contributors shall not be held liable for any direct, indirect, incidental, or consequential damages from use of this file.

This file is shared in good faith to help site owners improve their security posture. If you are unsure about any setting, contact us before applying it. Email: info@wpsecurestack.com | WhatsApp: +256 785 832 643 | wpsecurestack.com/resources

1 SECURITY KEYS & SALTS

These random strings make your login cookies and sessions secure. Think of them as secret passwords WordPress uses behind the scenes to verify every logged-in session. Without fresh, unique keys, attackers can forge login cookies and bypass authentication entirely.

■■ ***Note:*** *Never reuse keys across sites. After a hack, replace these immediately — they invalidate all active sessions and kick out anyone still logged in.*

```
define( 'AUTH_KEY',         'replace-with-generated-key' ); // ← REPLACE
define( 'SECURE_AUTH_KEY',  'replace-with-generated-key' ); // ← REPLACE
define( 'LOGGED_IN_KEY',    'replace-with-generated-key' ); // ← REPLACE
define( 'NONCE_KEY',        'replace-with-generated-key' ); // ← REPLACE
define( 'AUTH_SALT',        'replace-with-generated-key' ); // ← REPLACE
define( 'SECURE_AUTH_SALT', 'replace-with-generated-key' ); // ← REPLACE
define( 'LOGGED_IN_SALT',   'replace-with-generated-key' ); // ← REPLACE
define( 'NONCE_SALT',       'replace-with-generated-key' ); // ← REPLACE
```

■ *Get a fresh set in one click: https://api.wordpress.org/secret-key/1.1/salt/ — copy the entire output and paste it here, replacing all 8 lines. Refresh every 6–12 months as good practice.*

2 DATABASE TABLE PREFIX

By default, WordPress names all database tables starting with 'wp_' — for example wp_users and wp_options. Automated hacking tools know this and use it in SQL injection attacks. Changing the prefix to something random makes it much harder for automated tools to guess your table names.

■■ ***Note:*** *Set this BEFORE installing WordPress. Changing it on an existing live site requires also renaming all database tables. Use a plugin like 'Change Table Prefix' to do this safely on an existing site, or contact us for help.*

```
// Replace 'xk92m_' with your own random 5-6 character prefix
// Example random prefixes: 'p7vw2_', 'zt83k_', 'mn19x_'
$table_prefix = 'xk92m_'; // ← REPLACE with your own random prefix
```

3 DEBUG SETTINGS

Never show PHP errors on screen on a live site — error messages expose your server file paths, database name, plugin details, and PHP version to anyone who visits. Instead, log errors privately to a file outside the public web folder so you can diagnose problems silently without exposing anything publicly. The log path below points outside the public folder — adjust it to match your server.

■■ ***Note:*** *Do NOT set WP_DEBUG_LOG to true without specifying a path outside the webroot. The default debug.log inside wp-content/ is publicly readable at yourdomain.com/wp-content/debug.log — a goldmine for attackers.*

```
// Never show errors on screen — hides server details from visitors
define( 'WP_DEBUG',         false );
define( 'WP_DEBUG_DISPLAY', false );
@ini_set( 'display_errors', 0 );

// Log errors privately OUTSIDE the public webroot ← REPLACE path
define( 'WP_DEBUG_LOG', '/home/yourusername/private-logs/wp-errors.log' );
```

■ *Common private log paths: cPanel = /home/yourusername/private-logs/wp-errors.log | VPS/Ubuntu = /var/log/wordpress/wp-errors.log | Create the folder first: mkdir -p /home/yourusername/private-logs && chmod 750 /home/yourusername/private-logs*

4 FILE EDITOR & FILE MODIFICATIONS

DISALLOW_FILE_EDIT removes the built-in theme and plugin code editor from the WordPress dashboard. If an attacker gets into your admin panel, they cannot inject malicious code directly into your theme or plugin files. Always keep this true — there is no downside. DISALLOW_FILE_MODS blocks ALL file changes through the dashboard including plugin and theme updates. This is the most secure option, but only works if you commit to updating manually via FTP/SFTP. If updates will be missed, Option B (auto-patching) is safer in practice.

■■ ***Note:*** *DISALLOW_FILE_MODS and WP_AUTO_UPDATE_CORE conflict with each other. Use Option A OR Option B — never both at the same time.*

```
// Always keep true — removes dashboard code editor, no downside
define( 'DISALLOW_FILE_EDIT', true );

// ■■ CHOOSE ONE OPTION — do not use both ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

// Option A: Maximum lockdown — you manage ALL updates manually via FTP/SFTP
// define( 'DISALLOW_FILE_MODS', true );   // ← Uncomment for Option A
// (Remove or comment out WP_AUTO_UPDATE_CORE if using Option A)

// Option B (DEFAULT): Auto security patches ON — recommended for most sites
// define( 'DISALLOW_FILE_MODS', false );  // or remove this line entirely
define( 'WP_AUTO_UPDATE_CORE', 'minor' );  // ← Remove if using Option A
```

■ *Option A (max lockdown) = DISALLOW_FILE_MODS true + no auto-updates. Best for developers who update via FTP on a schedule. | Option B (auto-patch) = DISALLOW_FILE_MODS off + WP_AUTO_UPDATE_CORE minor. Best for client sites where manual updates may be forgotten.*

5 FORCE HTTPS FOR ADMIN & LOGINS

Forces WordPress to use HTTPS — the encrypted, secure connection — whenever someone logs in or uses the admin area. This protects your username and password from being intercepted on public Wi-Fi networks, shared connections, or anywhere traffic could be monitored.

■■ ***Note:*** *Only add this if your site already has an SSL certificate (HTTPS) installed and working. If your site does not have SSL, this setting will break your admin login page entirely.*

```
// Requires a valid SSL certificate to be installed on your domain
define( 'FORCE_SSL_ADMIN', true );
```

6 COOKIE SECURITY

Scopes your WordPress login cookies to your specific domain so they cannot leak to other applications or subdomains sharing the same hosting server. The leading dot (.) before the domain name is important — it covers both yourdomain.com and www.yourdomain.com. Without the dot, logging in on the www version of your site will break.

■■ ***Note:*** *Replace yourdomain.com with your actual domain before saving. If you experience login problems after adding this, remove the COOKIE_DOMAIN line and let WordPress manage cookies automatically.*

```
// Note the leading dot — covers both yourdomain.com and www.yourdomain.com
define( 'COOKIE_DOMAIN',  '.yourdomain.com' ); // ← REPLACE with your domain
define( 'COOKIEPATH',      '/' );
define( 'SITECOOKIEPATH',  '/wp-admin' );
```

7 DATABASE PERFORMANCE & CLEANUP

Keeps your WordPress database lean and fast. WordPress saves a copy of every post every time you edit it. Without limits, a site with active editors can accumulate thousands of unnecessary database rows. A bloated database is slower to back up, slower to restore after an incident, and slower for malware scanners to search through.

```
// Keep only the last 5 saved versions of each post
define( 'WP_POST_REVISIONS', 5 );

// Auto-save drafts every 2 minutes instead of the default 60 seconds
define( 'AUTOSAVE_INTERVAL', 120 );

// Permanently delete items from the trash after 7 days
define( 'EMPTY_TRASH_DAYS', 7 );
```

8 DIRECT ACCESS GUARD

This guard exits immediately if someone tries to load wp-config.php directly via a browser URL, rather than through WordPress's normal loading process. WordPress itself defines the ABSPATH constant during its bootstrap — if it is not defined, the file is being accessed directly and should exit.

■■ ***Note:*** *Place this at the very bottom of your wp-config.php additions, just before the 'That's all, stop editing' line. Do not place it inside a comment block.*

```
// Exits immediately if this file is accessed directly via URL
if ( ! defined( 'ABSPATH' ) ) {
    exit;
}
```

9 RECOMMENDED SERVER-LEVEL STEPS

These steps are done outside wp-config.php — on your server, via FTP, or in your .htaccess file. They complement the settings above and close gaps that wp-config.php cannot address on its own.

```
// A) Set file permissions — read-only for owner, no access for anyone else
//    Run this via SSH or your host's terminal:
//    chmod 400 wp-config.php

// B) Block browser access to wp-config.php — add to root .htaccess (Apache):
//    <Files wp-config.php>
//        order allow,deny
//        deny from all
//    </Files>

// C) Move wp-config.php one level ABOVE the webroot (optional but recommended)
//    WordPress automatically looks one directory up — no code change needed.
//    /home/yourusername/wp-config.php       ← move it here (not public)
//    /home/yourusername/public_html/        ← WordPress files stay here

// D) Block xmlrpc.php — add to .htaccess (Apache):
//    <Files xmlrpc.php>
//        order deny,allow
//        deny from all
//    </Files>

// E) Block the debug log if it must stay in wp-content — add to wp-content/.htaccess:
//    <Files debug.log>
//        deny from all
//    </Files>
```

■ *For the full .htaccess and Nginx/IIS security rules, download our separate Web Server Security Rules guide at wpsecurestack.com/resources*

■ PRE-DEPLOYMENT CHECKLIST — Tick before saving

[ ]	Backed up the original wp-config.php before making any changes
[ ]	Replaced all 8 Security Keys & Salts with fresh values from the WordPress API
[ ]	Changed the database table prefix from the default wp_ to a random prefix
[ ]	Set WP_DEBUG_LOG to a private path outside the public webroot — not inside wp-content/
[ ]	Chose either Option A (manual updates) OR Option B (auto-patch) — not both
[ ]	Replaced yourdomain.com in COOKIE_DOMAIN with your actual domain
[ ]	Added the ABSPATH exit guard at the bottom of your additions
[ ]	Confirmed SSL certificate is installed before using FORCE_SSL_ADMIN
[ ]	Pasted snippets ABOVE the 'That's all, stop editing' line
[ ]	Tested the site in a browser immediately after saving — homepage, login, and uploads

Emmanuel Tatyabala | Co-Founder, WPSecureStack

info@wpsecurestack.com | WhatsApp: +256 785 832 643

wpsecurestack.com/resources