🛡️ WordPress Security Checklist: Information Leakage

Basierend auf den Analysen von wordpress-bremen.de

1. Identität & Versionierung (Footprinting)

Diese Maßnahmen verhindern, dass Angreifer gezielt nach Schwachstellen für deine spezifische Version suchen können.

- [ ] **WP-Generator Tag entfernen:** Den `<meta name="generator" content="WordPress x.y.z">` aus dem Header löschen.
- [ ] **Versions-Strings an Assets:** `?ver=x.y.z` von CSS- und JS-Dateien entfernen (besonders bei Core-Dateien).
- [ ] **Standard-Dateien löschen:** `readme.html`, `license.txt` und `wp-config-sample.php` aus dem Root-Verzeichnis entfernen.
- [ ] **Header-Cleaning:** Entfernen von Link-Tags wie `rsd_link`, `wlwmanifest_link` und `wp_shortlink_wp_head`.

2. Benutzer-Exploration (Enumeration)

Verhindert, dass Login-Namen (Usernames) durch einfache URL-Abfragen oder API-Calls ausgespäht werden.

- [ ] **Author-Scanning blockieren:** Anfragen wie `?author=1` via `.htaccess` oder Plugin abfangen und umleiten.
- [ ] **REST-API User-Endpunkt:** Den Zugriff auf `/wp-json/wp/v2/users` für nicht-authentifizierte Anfragen sperren.
- [ ] **Nickname vs. Login:** Sicherstellen, dass der "Öffentliche Name" im Profil niemals identisch mit dem Login-Usernamen ist.
- [ ] **Author-Slug-Verleierung:** Falls möglich, den URL-Slug `/author/benutzername/` individualisieren (z.B. mit einer ID oder einem Pseudonym).

3. Schnittstellen & Verzeichnisse

Schließt offene Türen, die für moderne Web-Anwendungen oft nicht mehr (in dieser Form) benötigt werden.

- [ ] **XML-RPC deaktivieren:** Die Datei `xmlrpc.php` für externe Zugriffe sperren (außer bei Nutzung der Jetpack-App o.ä.).
- [ ] **Directory Browsing unterbinden:** `Options -Indexes` in der `.htaccess` setzen, damit Ordnerinhalte (wie `/uploads/`) nicht aufgelistet werden.
- [ ] **REST-API einschränken:** Die API global nur für eingeloggte Nutzer oder spezifische App-Anwendungen freigeben.

4. Server-Ebene & Dateischutz

Zusätzliche Härtung, um den Zugriff auf sensible Konfigurationsdaten zu verhindern.

- [ ] **wp-config.php schützen:** Zugriff via `.htaccess` explizit verbieten.
- [ ] **Install-Skripte sperren:** Zugriff auf `wp-admin/install.php` und `upgrade.php` nach der Installation einschränken.
- [ ] **Fehlermeldungen unterdrücken:** `display_errors` in der `php.ini` oder via `wp-config.php` (WP_DEBUG_DISPLAY) auf `false` setzen, um Pfad-Enthüllungen bei Fehlern zu vermeiden.

Technischer Quick-Check

Versuche folgende URLs im Inkognito-Modus aufzurufen. Erhältst du Daten, ist die Lücke noch offen:

1. `deine-domain.de/?author=1`
2. `deine-domain.de/wp-json/wp/v2/users`
3. `deine-domain.de/readme.html`