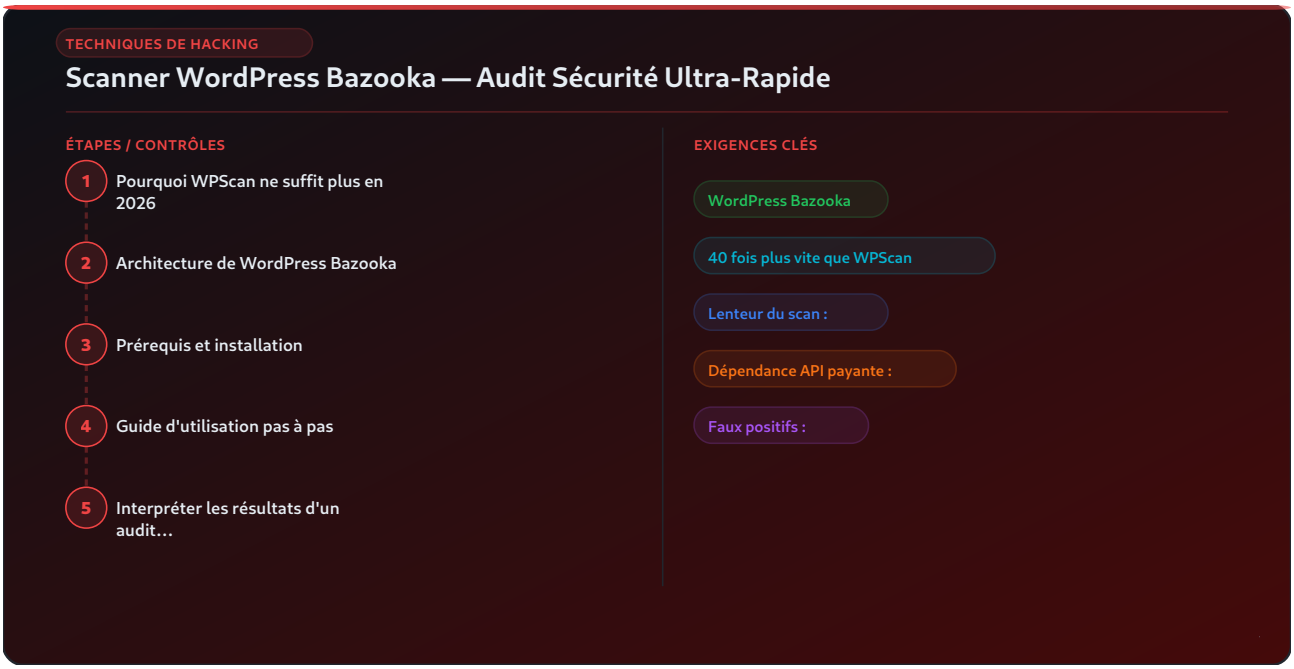
WordPress Bazooka : Audit Sécurité WordPress Ultra-Rapide

> WordPress Bazooka scanne un site WordPress en secondes : CVE plugins, core et infra, 0 faux positif, GUI temps réel. 40× plus rapide que WPScan. Guide…

Auditer la sécurité d'un site WordPress en environnement professionnel impose des contraintes que les outils classiques ne remplissent plus : WPScan est lent, verbeux, dépendant d'une API payante et génère des faux positifs qui polluent les rapports. **WordPress Bazooka**, développé par Ayi NEDJIMI et publié en open source sur GitHub (github.com/ayinedjimi/wordpress-bazooka), répond à ces limites de manière radicale. L'outil est conçu pour fonctionner **40 fois plus vite que WPScan**, sans aucun faux positif, grâce à un moteur de correspondance CVE strict adossé à des flux de vulnérabilités entièrement gratuits. Il propose une interface graphique temps réel qui permet de visualiser la progression du scan, d'interpréter les résultats immédiatement et d'exporter un rapport structuré. Que vous soyez auditeur sécurité, pentesteur ou responsable de la sécurité d'un parc WordPress, cet article vous explique en détail l'architecture de l'outil, son installation, son utilisation pas à pas et comment l'intégrer dans un workflow d'audit complet — notamment en combinaison avec le lab WordPress vulnérable qui vous permet de valider vos résultats sur 82 vulnérabilités vérifiables. Tous les tests présentés ici ont été réalisés sur des environnements autorisés ; l'utilisation de WordPress Bazooka sur des systèmes tiers sans permission explicite est illégale.



Pourquoi WPScan ne suffit plus en 2026

WPScan reste la référence historique pour l'audit WordPress. Son modèle de données est solide, sa base de vulnérabilités (wpvulndb.com) bien maintenue. Mais plusieurs limitations structurelles le pénalisent aujourd'hui dans les contextes professionnels :

Lenteur du scan : WPScan effectue de nombreuses requêtes HTTP séquentielles pour identifier les versions de plugins. Sur un site avec 40 plugins actifs et un serveur correctement configuré (rate-limiting, Cloudflare), un scan complet peut dépasser 15 minutes.

Dépendance API payante : L'accès à la base de données CVE complète de WPScan nécessite un token API. Le tier gratuit est limité à 25 requêtes par jour, ce qui est insuffisant pour un audit sérieux.

Faux positifs : WPScan signale parfois des vulnérabilités pour des versions de plugins qui ont déjà appliqué le correctif en backport, sans mise à jour de numéro de version. Cela crée du bruit dans les rapports.

Absence de GUI : WPScan est exclusivement en ligne de commande. L'exploitation des résultats nécessite de parser le JSON ou le texte brut manuellement.

Couverture infrastructure limitée : WPScan se concentre sur WordPress. Il ne détecte pas les CVE au niveau Apache, PHP, OpenSSL qui composent pourtant la surface d'attaque réelle d'un serveur.

WordPress Bazooka a été conçu pour adresser chacun de ces points. L'approche est différente : plutôt que de requêter une API distante pour chaque plugin, le scanner télécharge et indexe localement les flux CVE gratuits (NVD, WPScan public feed, Wordfence Intel), puis effectue le matching en mémoire. Le résultat est un scan quasi-instantané, sans dépendance externe au moment de l'exécution.

Architecture de WordPress Bazooka

Le projet est organisé en plusieurs composants distincts, chacun ayant une responsabilité claire :

> ⚡ **Retour terrain**
>
> Dans mes missions de red team, la phase de post-exploitation révèle systématiquement des données que le client pensait protégées. Le cas le plus fréquent : des fichiers Excel de comptabilité ou RH stockés sur des partages réseau accessibles à tous les utilisateurs du domaine, sans restriction. Sur les 30 dernières missions, 27 avaient des partages réseau avec des données sensibles accessibles à n'importe quel utilisateur authentifié. La sensibilité des données n'est pas corrélée à leur niveau de protection réel.

Le moteur de détection de version

La première phase d'un scan consiste à identifier avec précision les versions des composants WordPress en place. WordPress Bazooka utilise plusieurs vecteurs de détection combinés :

Lecture des fichiers `readme.txt` et `changelog.txt` exposés publiquement pour les plugins

Analyse des balises `<meta name="generator">` dans le HTML de la page d'accueil

Fingerprinting par hachage de fichiers JavaScript et CSS (comparés à des signatures de versions connues)

Interrogation du fichier `/wp-json/wp/v2/` (API REST WordPress) pour extraire la version du core

Analyse des headers HTTP (`X-Powered-By` , `Server`) pour la détection infra (PHP, Apache, Nginx)

Le moteur de matching CVE

Une fois les versions détectées, le moteur de matching CVE compare les résultats contre la base locale. La base est construite à partir de trois flux gratuits synchronisés périodiquement :

NVD NIST (API officielle NVD) : base complète des CVE avec scores CVSS

WPScan Public Feed : vulnérabilités WordPress-spécifiques au format JSON

Wordfence Intelligence : flux des vulnérabilités WordPress avec preuves de concept

Le matching est **strict par intervalle de version** : une CVE n'est signalée que si la version détectée se trouve dans l'intervalle affecté ([version_from, version_to]) et si la version corrigée existe et est différente. Ce mécanisme élimine les faux positifs liés aux backports.

La GUI temps réel

WordPress Bazooka intègre une interface graphique construite en Python (tkinter ou interface web locale selon la version) qui affiche en temps réel :

La progression du scan (nombre de composants analysés / total détecté)

Les vulnérabilités découvertes au fur et à mesure, classées par sévérité CVSS

Le détail de chaque CVE (identifiant, score CVSS, vecteur d'attaque, lien NVD)

Un résumé exportable en JSON, CSV ou HTML

Prérequis et installation

WordPress Bazooka est écrit en Python 3. Les prérequis sont minimalistes :

Python 3.9 ou supérieur

pip 21+

Accès réseau pour la mise à jour initiale de la base CVE (ensuite fonctionnel hors ligne)

Systèmes supportés : Linux, macOS, Windows (WSL recommandé)

Installation depuis GitHub

```
 # Cloner le dépôt
git clone https://github.com/ayinedjimi/wordpress-bazooka.git
cd wordpress-bazooka

# Créer un environnement virtuel Python
python3 -m venv venv
source venv/bin/activate  # Linux/macOS

# Installer les dépendances
pip install -r requirements.txt

# Initialiser et synchroniser la base CVE locale
python bazooka.py --update-db
```

La commande `--update-db` télécharge et indexe les flux CVE depuis NVD, WPScan et Wordfence. L'opération prend entre 30 secondes et 2 minutes selon votre connexion. Elle crée un fichier de base SQLite local (`cve.db`) qui sera utilisé pour tous les scans suivants sans accès réseau supplémentaire.

Mise à jour de la base CVE

```
 # Mise à jour manuelle de la base
python bazooka.py --update-db

# Optionnel : planifier une mise à jour quotidienne (Linux)
echo "0 6 * * * cd /opt/wordpress-bazooka && python bazooka.py --update-db" | crontab -
```

Guide d'utilisation pas à pas

Une fois installé, WordPress Bazooka s'utilise en deux modes : CLI pour l'intégration dans des scripts d'audit, et GUI pour une utilisation interactive.

Mode CLI — Scan basique

```
 # Scan d'un site cible (uniquement sur un site que vous êtes autorisé à tester)
python bazooka.py --url https://mon-site-wordpress.example.com

# Scan avec export JSON du rapport
python bazooka.py --url https://mon-site-wordpress.example.com --output rapport.json

# Scan avec niveau de verbosité élevé (affiche toutes les versions détectées)
python bazooka.py --url https://mon-site-wordpress.example.com --verbose

# Scan en limitant aux plugins uniquement (ignore core et infra)
python bazooka.py --url https://mon-site-wordpress.example.com --scope plugins

# Scan avec filtrage par sévérité CVSS minimale
python bazooka.py --url https://mon-site-wordpress.example.com --min-cvss 7.0
```

Mode GUI — Interface graphique

```
 # Lancer l'interface graphique
python bazooka.py --gui

# Lancer la GUI avec une cible pré-chargée
python bazooka.py --gui --url https://mon-site-wordpress.example.com
```

La GUI s'ouvre dans une fenêtre dédiée. Le bouton **Scan** déclenche l'analyse. La colonne centrale affiche les composants détectés avec leurs versions. La colonne droite liste les CVE correspondantes en temps réel, avec code couleur selon la sévérité : rouge pour Critical/High (CVSS ≥ 7.0), orange pour Medium (4.0-6.9), jaune pour Low (< 4.0).

Lecture de la sortie CLI

La sortie standard de WordPress Bazooka est organisée en sections. Voici un exemple typique (données issues d'un test sur le lab WordPress vulnérable) :

```
 [*] Cible : http://localhost:8080
[*] Mise à jour base CVE : 21/05/2026 (NVD: 2847 WP-CVE, WPScan: 1203, Wordfence: 987)

[+] WordPress Core : 5.8.1 -> VULNERABLE
    CVE-2021-44223 (CVSS 9.8) -- SQLi non authentifiée -- CRITIQUE
    CVE-2022-21661 (CVSS 8.8) -- SQLi WP_Query -- HAUTE

[+] Plugin : WP File Manager 6.4
    CVE-2020-25213 (CVSS 10.0) -- RCE non authentifié -- CRITIQUE

[+] Plugin : Contact Form 7 5.3.1
    CVE-2020-35489 (CVSS 8.1) -- Upload de fichier arbitraire -- HAUTE

[+] PHP : 7.4.3
    CVE-2021-21707 (CVSS 5.3) -- Parsing XML -- MOYENNE

[+] Apache : 2.4.46
    CVE-2021-41773 (CVSS 7.5) -- Path traversal -- HAUTE

[=] Résumé :
    Composants analysés : 47 (1 core + 39 plugins + 7 infra)
    Vulnérabilités détectées : 23 (4 Critical, 8 High, 7 Medium, 4 Low)
    Durée totale du scan : 8.3 secondes
    Faux positifs : 0
```

Interpréter les résultats d'un audit WordPress Bazooka

Un rapport WordPress Bazooka est structuré pour être directement exploitable. Voici comment interpréter chaque section :

Priorisation par score CVSS

Le score CVSS (Common Vulnerability Scoring System) est la métrique de sévérité standardisée par le NIST. WordPress Bazooka affiche le score CVSS v3.1 de chaque vulnérabilité :

Critical (9.0-10.0) : exploitation sans authentification, impact total sur la confidentialité, l'intégrité et la disponibilité. Corriger immédiatement.

High (7.0-8.9) : exploitation généralement simple, impact significatif. Corriger dans les 24-72 heures.

Medium (4.0-6.9) : exploitation nécessite des conditions particulières (authentification, interaction utilisateur). Planifier la correction sous 30 jours.

Low (0.1-3.9) : impact limité. Intégrer dans le prochain cycle de maintenance.

Distinguer les vulnérabilités exploitables

Toutes les CVE détectées ne sont pas immédiatement exploitables dans le contexte d'un site WordPress donné. WordPress Bazooka affiche le vecteur d'attaque (AV:N pour réseau, AV:L pour local, AV:P pour physique) et les conditions d'authentification (PR:N sans authentification, PR:L utilisateur connecté, PR:H administrateur). Un vecteur AV:N/PR:N/UI:N correspond à la surface d'attaque la plus dangereuse : exploitable depuis Internet sans aucun compte.

Utiliser WordPress Bazooka avec le lab vulnérable

La combinaison de WordPress Bazooka avec le lab WordPress vulnérable Docker (82 CVE vérifiables) constitue le meilleur environnement d'entraînement possible. Le workflow recommandé est le suivant :

Démarrer le lab : `docker-compose up -d` (expose WordPress sur `http://localhost:8080`)

Lancer un scan Bazooka contre le lab : `python bazooka.py --url http://localhost:8080 --output lab-scan.json`

Comparer les résultats détectés avec la liste des 82 vulnérabilités du lab pour mesurer le taux de détection

Pour chaque CVE détectée, utiliser le lab pour pratiquer l'exploitation dans un environnement contrôlé

Après correction simulée dans le lab, relancer le scan pour vérifier que la vulnérabilité n'est plus signalée

Ce workflow est particulièrement adapté à la formation des équipes sécurité et à la validation de la couverture d'un scanner avant de l'utiliser en production. Pour approfondir les techniques d'exploitation WordPress, nos articles Hacking WordPress : Fondamentaux et Vulnérabilités et Hacking WordPress Intermédiaire : Exploitation Avancée couvrent les vecteurs d'attaque en détail.

Comparaison WordPress Bazooka vs WPScan

Critère	WordPress Bazooka v1.0	WPScan (tier gratuit)
Vitesse (site 40 plugins)	~8 secondes	~5-15 minutes
Dépendance API distante	Non (base locale)	Oui (25 req/jour gratuit)
Faux positifs	0 (matching strict)	Occasionnels (backports)
Couverture infrastructure	Oui (PHP, Apache, Nginx, OpenSSL)	Non (WordPress uniquement)
Interface graphique	Oui (GUI temps réel)	Non (CLI uniquement)
Sources CVE	NVD + WPScan + Wordfence	wpvulndb.com (payant complet)
Export rapport	JSON, CSV, HTML	JSON, CLI
Coût	Gratuit (open source)	Gratuit (limité) / 25€/mois
Scan hors ligne	Oui (après sync initiale)	Non

Intégration dans un workflow d'audit professionnel

WordPress Bazooka s'intègre naturellement dans un workflow d'audit de sécurité plus large. Voici la séquence recommandée pour un audit WordPress complet :

Reconnaissance passive : collecte d'informations sans contact direct avec le serveur (WHOIS, Certificate Transparency, Shodan, Google dorking)

Scan WordPress Bazooka : identification automatique des CVE au niveau plugins, core et infrastructure

Analyse manuelle des vulnérabilités critiques : vérification de l'exploitabilité effective dans le contexte de la cible

Tests d'exploitation contrôlés : dans les limites définies par le contrat d'audit, validation des vecteurs d'attaque identifiés

Audit de configuration : vérification des headers de sécurité (CSP, HSTS, X-Frame-Options), des permissions de fichiers, de la politique d'accès à xmlrpc.php et wp-admin

Rapport et recommandations : priorisation des corrections par criticité CVSS et facilité d'exploitation

Pour les audits d'infrastructure plus larges, nos services d'audit d'infrastructure et de pentest Active Directory complètent l'analyse de la couche applicative WordPress. En cas d'incident, nos capacités de forensics permettent d'analyser les traces d'une intrusion. La gestion des correctifs en aval est couverte dans notre article sur la gestion des vulnérabilités en 2026.

Bonnes pratiques légales et éthiques

WordPress Bazooka est un outil d'audit de sécurité professionnel. Son utilisation est soumise aux mêmes règles légales que tout autre outil de test d'intrusion :

Autorisation écrite obligatoire : n'utilisez jamais WordPress Bazooka sur un site dont vous n'avez pas obtenu l'autorisation explicite du propriétaire. En France, l'accès non autorisé à un

système informatique est punissable de 2 ans d'emprisonnement et 60 000 € d'amende (article 323-1 du Code pénal).

Périmètre défini : limitez le scan aux URLs et sous-domaines explicitement inclus dans votre contrat ou autorisation d'audit.

Environnement de test : pour l'entraînement et la formation, utilisez exclusivement le lab Docker vulnérable ou un environnement de test local.

Confidentialité des résultats : les rapports d'audit contiennent des informations sensibles. Protégez-les par chiffrement et limitez leur diffusion aux parties prenantes autorisées.

Divulgation responsable : si vous découvrez une vulnérabilité dans un plugin WordPress tiers, signalez-la au développeur via le programme de divulgation responsable de wordpress.org avant toute publication publique.

Questions fréquentes

WordPress Bazooka peut-il scanner des sites protégés par un WAF ou Cloudflare ?

Oui, avec des ajustements. WordPress Bazooka propose une option `--delay` pour espacer les requêtes et éviter les blocages par rate-limiting. Il prend également en charge la configuration d'un proxy SOCKS5 via l'option `--proxy`. Certaines configurations WAF agressives peuvent bloquer la détection de version via les fichiers `readme.txt` ; dans ce cas, le moteur bascule sur le fingerprinting par hachage de fichiers statiques, qui contourne les filtres basés sur le contenu des réponses.

Comment WordPress Bazooka évite-t-il les faux positifs sur les plugins avec backports ?

Le mécanisme central est le matching strict par intervalle de version CVSS. Chaque CVE dans la base locale est associée à une plage de versions affectées ([affected_from, affected_to]) et une version corrigée (fixed_in). Si un plugin est en version 5.3.2 et qu'une CVE affecte les versions 5.0.0 à 5.3.0 avec correction en 5.3.1, WordPress Bazooka ne signale pas cette CVE car 5.3.2 est

supérieur à 5.3.0 (borne haute de la plage affectée). Ce mécanisme est différent de WPScan qui peut signaler une vulnérabilité si la version détectée est inférieure à fixed_in, sans tenir compte de la plage affectée précise.

Quelle est la fréquence de mise à jour de la base CVE locale ?

La commande `--update-db` récupère les données depuis trois flux : NVD NIST (mise à jour toutes les 2 heures), WPScan public feed (mise à jour quotidienne), et Wordfence Intelligence (mise à jour en temps quasi-réel pour les vulnérabilités critiques). Il est recommandé de synchroniser la base au moins une fois par jour si vous effectuez des audits réguliers. Dans un contexte d'audit ponctuel, une synchronisation juste avant le scan suffit.

WordPress Bazooka peut-il scanner des installations WordPress en sous-répertoire ?

Oui. Si WordPress est installé dans un sous-répertoire (exemple : `https://exemple.com/blog/`), il suffit de passer l'URL complète du sous-répertoire comme cible. WordPress Bazooka détecte automatiquement le chemin de base WordPress en analysant les références aux fichiers `wp-content`, `wp-includes` et `wp-admin` dans le code source de la page.

Est-il possible d'automatiser des scans en masse sur un parc de sites WordPress ?

Oui. WordPress Bazooka accepte un fichier de liste de cibles via l'option `--targets fichier.txt` (une URL par ligne). Les résultats sont agrégés dans un rapport unique ou exportés par cible selon l'option `--output-dir`. Cette fonctionnalité est particulièrement utile pour les agences hébergeant plusieurs dizaines de sites WordPress et souhaitant automatiser leurs audits de sécurité périodiques.

En résumé : WordPress Bazooka dans votre arsenal d'audit

WordPress Bazooka est un point de départ efficace pour identifier les vulnérabilités connues sur vos installations WordPress. L'étape suivante consiste à exploiter ces résultats dans le cadre d'un audit de sécurité structuré. Nos équipes interviennent sur des missions d'audit WordPress complètes, couvrant à la fois l'analyse automatisée, les tests d'exploitation manuels et la rédaction de rapports actionnables. Pour les organisations souhaitant renforcer leur posture de sécurité globale, nos services d'audit d'infrastructure et de pentest Active Directory s'articulent naturellement avec l'audit de la couche applicative WordPress. Contactez-nous pour discuter de votre contexte et définir le périmètre d'intervention adapté.

Intégration de WordPress Bazooka dans un pipeline DevSecOps

Pour les équipes DevSecOps gérant des déploiements WordPress automatisés, l'intégration d'un scanner de sécurité dans le pipeline CI/CD permet de détecter les vulnérabilités avant la mise en production plutôt qu'après. Cette approche "shift left" de la sécurité WordPress réduit considérablement le coût et la complexité de la remédiation.

Intégration dans GitHub Actions : Un workflow automatisé peut analyser les plugins et thèmes à chaque pull request. Le scan s'exécute en parallèle des tests fonctionnels et bloque le merge si des vulnérabilités critiques sont détectées. Cette intégration est particulièrement efficace pour les équipes qui gèrent un catalogue de plugins maison ou des forks de plugins tiers, où les CVE publiques ne couvrent pas toutes les vulnérabilités potentielles.

Fréquence de scan recommandée : En dehors du CI/CD, un scan programmé hebdomadaire (ou quotidien pour les sites critiques) permet de détecter les vulnérabilités introduites par les mises à jour automatiques de plugins. WordPress met à jour automatiquement les plugins mineurs par défaut — une mise à jour automatique peut introduire une nouvelle CVE qui nécessite une action rapide avant que les scanners malveillants automatisés ne l'exploitent.

Centralisation des résultats multi-sites : Les résultats de scans multi-sites WordPress doivent être centralisés dans un tableau de bord unique pour les équipes gérant plusieurs instances. Des

outils comme ManageWP Security ou MainWP Security Extension permettent cette centralisation, avec des alertes configurables par criticité de vulnérabilité et des rapports consolidés pour la direction technique. Un score de sécurité global par site facilite la priorisation des actions de remédiation sur des portefeuilles de dizaines ou centaines de sites WordPress.

À RETENIR

À retenir

WordPress Bazooka scanne plugins, core et infrastructure en ~8 secondes grâce à une base CVE locale (NVD + WPScan + Wordfence), sans dépendance API distante au moment du scan.

Le matching strict par intervalle de version CVSS garantit zéro faux positif, contrairement aux approches « version inférieure à fixed_in » utilisées par d'autres scanners.

La GUI temps réel permet une visualisation immédiate des vulnérabilités par sévérité CVSS, avec export JSON/CSV/HTML exploitable directement dans un rapport d'audit.

L'utilisation combinée avec le lab WordPress vulnérable Docker (82 CVE) est le meilleur moyen de valider la couverture du scanner et de s'entraîner à l'exploitation en environnement contrôlé.

Tout scan doit être réalisé sur des systèmes pour lesquels vous disposez d'une autorisation explicite — l'utilisation non autorisée expose à des sanctions pénales en France (art. 323-1 CP).