AGENCE NATIONALE DES TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Centre de Réponse aux Incidents de Sécurité Informatique

NATIONAL AGENCY FOR INFORMATION AND COMMUNICATION TECHNOLOGIES

Computer Incident Response Team

ALERTE DE SECURITE

Vulnérabilités critiques PHP2Shell dans le cœur de PHP

Contenu

I. Contexte 3
II. Vulnérabilités identifiées 3
III. Vecteur d'attaque 4
IV. Menaces actives 4
V. Exposition de l'Afrique et du Cameroun 4
VI. Conclusion 5

I. Contexte

Le Centre de Réponse aux Incidents de Sécurité Informatique (CIRT.cm) alerte les organisations camerounaises sur la divulgation, le 19 octobre 2026, de plusieurs vulnérabilités critiques affectant le moteur du langage PHP. Regroupées sous le nom de code **PHP2Shell**, ces failles permettent à un attaquant non authentifié d'exécuter du code arbitraire sur un serveur web, quelle que soit l'application qui y tourne.

PHP alimente environ **70 % des sites web dans le monde**, dont WordPress, Drupal, Joomla et MediaWiki, l'infrastructure qui fait tourner Wikipédia. **Toutes les versions publiées depuis PHP 5.4 sont touchées.** Une faille dans le moteur concerne donc des millions de serveurs, indépendamment de la qualité du code applicatif qu'ils hébergent. Un simple programme affichant « Bonjour le monde » est vulnérable s'il tourne sur une version non corrigée.

Les vulnérabilités ont été découvertes par les équipes de recherche de Wiz, avec l'appui de modèles d'intelligence artificielle de dernière génération. Elles dormaient dans la base de code depuis plus de dix ans. Ce qui a changé, c'est le coût de l'exploitation : là où un chercheur humain devait enchaîner plusieurs bugs et adapter un exploit à chaque configuration mémoire, un modèle d'IA le fait aujourd'hui en quelques heures.

II. Vulnérabilités identifiées

CVE	Score CVSS	Nature de la faille	Impact
CVE-2026-109618	**9.8 (Critique)**	Use after free dans le désérialiseur de progression d'upload. Une erreur de mémoire insuffisante (OOM) interrompt le traitement et laisse l'état interne de PHP corrompu.	Divulgation de la mémoire du processus (mots de passe, jetons, clés API, certificats) et exécution de code à distance sans authentification.
CVE-2026-109619	**9.8 (Critique)**	Use after free sur l'état global conservé entre les requêtes. Un worker réutilise un pointeur vers une mémoire déjà libérée, combiné à une confusion de type dans OPcache.	Corruption de la mémoire partagée et exécution de code à distance sans authentification.

Versions corrigées : PHP 8.2.35, 8.3.36, 8.4.27 et 8.5.12. Les branches 8.1 et inférieures ne reçoivent plus de correctifs de sécurité et doivent être abandonnées.

III. Vecteur d'attaque

Il ne s'agit pas d'une faille dans WordPress, Drupal ou Joomla : c'est le moteur commun sur lequel reposent ces plateformes qui est en cause. Les chercheurs décrivent trois aspects majeurs :

- **Exécution de code à distance :** Un script unique, baptisé php2shell, est capable d'exécuter du code à distance sur des déploiements WordPress, Drupal et Joomla à jour, en utilisant leurs images de conteneurs officielles les plus récentes.
- **Lecture de la mémoire du processus :** Un second scénario, plus discret mais tout aussi grave, permet de lire la mémoire du processus PHP sans exécuter de code. Dans un hébergement mutualisé où un même worker est partagé entre plusieurs sites, un attaquant peut récupérer les identifiants de connexion, les clés d'API ou les certificats d'un autre client hébergé sur la même machine. C'est le même mécanisme que Heartbleed en 2014, appliqué à PHP.
- **Faible traçabilité :** Ces attaques s'exécutent sous la couche applicative. Les journaux de l'application peuvent ne conserver presque aucune trace exploitable. Il convient de surveiller les requêtes multipart inhabituelles suivies d'erreurs de limite mémoire ou de réponses HTTP 500 répétées.

IV. Menaces actives

- **PHP2Shell :** Chaîne d'exploitation complète publiée, avec démonstrations vidéo sur WordPress, Drupal et Joomla. Le code n'est pas encore largement distribué, mais la publication des détails techniques ouvre la voie à des adaptations rapides.
- **Divulgation mémoire à grande échelle :** Technique d'extraction de secrets depuis la mémoire des workers PHP, sans authentification. Risque majeur pour les hébergeurs mutualisés et les plateformes partagées entre plusieurs clients.
- **Variantes non identifiées :** Les deux CVE ne couvrent que quatre instances exploitables d'un même défaut architectural. Les chercheurs signalent l'existence d'autres variantes, suivies et probablement non suivies, dans la base de code.

V. Exposition de l'Afrique et du Cameroun

Aucune campagne spécifiquement dirigée contre le Cameroun ou l'Afrique de l'Ouest n'a été rapportée à ce stade, la divulgation datant de moins de 24 heures. La nature de la faille rend toutefois le continent particulièrement exposé : le parc web africain repose massivement sur PHP, souvent via WordPress, et une part importante des sites institutionnels et commerciaux

est hébergée sur des offres mutualisées où la version du moteur n'est ni connue ni maîtrisée par le propriétaire du site.

Un scan de masse sur les adresses africaines est le scénario le plus probable dans les prochains jours. Le CIRT.cm maintient sa veille et diffusera une mise à jour si des indicateurs de compromission sont identifiés sur le territoire.

VI. Conclusion

Il ne s'agit pas d'une faille dans un plugin, un thème ou un module : c'est le moteur qui exécute le code de votre site qui est atteignable. Le correctif fourni par l'équipe PHP traite le défaut à sa racine et élimine l'ensemble des variantes connues. La mise à jour vers **PHP 8.2.35, 8.3.36, 8.4.27 ou 8.5.12** est impérative.

Pour les organisations qui ne peuvent pas appliquer le correctif sans délai, les grands fournisseurs de cloud et de CDN (Cloudflare, AWS, Microsoft, Google, Akamai, Fastly, Imperva, Vercel, Automattic) ont mis en place des règles de protection. Ces mesures d'atténuation ne remplacent pas la mise à jour.

Le CIRT.cm recommande de prendre les précautions suivantes :

1. **Inventorier :** Identifier la version PHP de chaque serveur web et application exposée. Toute version inférieure à 8.2.35, 8.3.36, 8.4.27 ou 8.5.12 doit être considérée comme vulnérable ;
2. **Corriger :** Appliquer les correctifs vers PHP 8.2.35, 8.3.36, 8.4.27 ou 8.5.12 sans attendre le cycle de maintenance habituel ;
3. **Migrer :** Abandonner les branches 8.1 et antérieures, qui ne reçoivent plus de mises à jour de sécurité ;
4. **Interroger l'hébergeur :** Sur une offre mutualisée, demander à votre hébergeur la version du moteur PHP utilisée et son calendrier de mise à jour ;
5. **Atténuer :** Activer les règles de protection proposées par votre fournisseur de CDN ou de WAF pour les applications qui ne peuvent pas être corrigées immédiatement ;
6. **Surveiller :** Suivre les requêtes multipart inhabituelles suivies d'erreurs de limite mémoire, de redémarrages de workers ou de séries de réponses HTTP 500 ;
7. **Renouveler les secrets :** Par précaution, renouveler les clés d'API, jetons d'accès et identifiants de base de données des services exposés sur un serveur PHP non corrigé, la divulgation mémoire permettant leur extraction.