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 SÉCURITÉ

Java4Shell : exécution de code à distance dans la JVM HotSpot

Contenu

I. Contexte 3
II. Vulnérabilité identifiée 3
Versions et plateformes affectées 3
III. Vecteur d'attaque 4
IV. Menaces actives 4
V. Exposition de l'Afrique et du Cameroun 5
VI. Conclusion et actions recommandées 5
VII. Références 6

I. Contexte

Le 17 novembre 2026, Oracle a publié des correctifs pour une vulnérabilité critique de la machine virtuelle Java HotSpot, référencée CVE-2026-114938 et surnommée « Java4Shell ». Le même jour, l'agence américaine CISA a publié une alerte invitant toutes les organisations à appliquer ces correctifs immédiatement.

La faille a été découverte par les chercheurs de Calif. Elle affecte le décodeur Base64 optimisé de la JVM HotSpot, présent dans toutes les distributions OpenJDK depuis la version 17.0.6 (Oracle, Eclipse Temurin, Amazon Corretto, Azul Zulu, Red Hat, Microsoft, paquets Linux, etc.). Une donnée Base64 spécialement conçue peut provoquer le plantage de la JVM et permettre une exécution de code à distance sans authentification.

II. Vulnérabilité identifiée

La vulnérabilité est une écriture hors limites dans le tas (heap out of bounds write) au sein du code assembleur qui accélère le décodage Base64 grâce aux instructions processeur AVX512 VBMI. Une entrée Base64 malformée et courte amène ce code à écrire 62 octets au delà de la fin d'un tampon mémoire, avant même que Java ne rejette l'entrée.

CVE	CVE-2026-114938
Nom de code	Java4Shell
Composant	JVM HotSpot, décodeur Base64 (java.util.Base64)
Type	Corruption mémoire, écriture hors limites dans le tas
Impact	Exécution de code à distance sans authentification, fuite de mémoire
Score CVSS	**9.2 Critique (CVSS v4.0), 8.1 Élevé selon Oracle (CVSS v3.1)**
Statut	**Alerte CISA du 17 novembre 2026, aucune exploitation confirmée à ce jour**
Date de publication	17 novembre 2026
Correctif	Builds Oracle du 17 novembre 2026 (suffixe .1)

Versions et plateformes affectées

Sont concernées les JVM HotSpot à partir de la version 17.0.6, ainsi que les versions 21, 25 et 27, uniquement lorsqu'elles s'exécutent sur des processeurs x86_64 disposant des instructions AVX512 VBMI : Intel Xeon (Ice Lake, Sapphire Rapids, Emerald Rapids, Granite Rapids) et

AMD EPYC ou Ryzen (Zen 4 et plus récents). Les versions standard de Java 8 et 11 ainsi que les serveurs ARM64 ne sont pas affectés.

Produit	Versions affectées	Versions corrigées
Oracle Java SE 17 / OpenJDK 17	17.0.6 à 17.0.21	17.0.21.1 ou ultérieure
Oracle Java SE 21 / OpenJDK 21	Jusqu'à 21.0.13	21.0.13.1 ou ultérieure
Oracle Java SE 25 / OpenJDK 25	Jusqu'à 25.0.5	25.0.5.1 ou ultérieure
Oracle Java SE 27 / OpenJDK 27	Jusqu'à 27.0.1	27.0.1.1 ou ultérieure
Java SE 8 Enterprise Performance Pack	8u511b33 perf	8u513 perf
Amazon Corretto 17	Antérieures au correctif	17.0.21.1.2 ou ultérieure
Autres distributions OpenJDK	Builds sans correctif	Selon l'avis de chaque éditeur

III. Vecteur d'attaque

Le décodage Base64 est omniprésent dans les applications Java. L'exploitation se déroule comme suit :

- L'attaquant envoie une donnée Base64 malformée à une application Java exposée, par exemple dans un en tête d'authentification, un jeton (JWT, SAML) ou un fichier téléversé.
- Ce décodage intervient souvent avant toute vérification d'identité, ce qui rend l'attaque possible sans authentification.
- Le décodeur optimisé écrit en dehors de la mémoire prévue, ce qui permet de lire la mémoire de la JVM (clés, jetons, données personnelles) puis d'exécuter du code arbitraire sur le serveur.
- Un scénario côté client est également signalé : un programme Java qui se connecte en TLS à un serveur malveillant peut être compromis.

Les chercheurs ont démontré une exécution de code sans authentification dans la configuration par défaut de Keycloak, Jira, Confluence, JFrog Artifactory, Jenkins et PingFederate, ainsi qu'une exécution de code par un utilisateur authentifié dans Apache Kafka, Apache Solr, Elasticsearch, Gerrit et Spinnaker. Des applications basées sur Apache Struts et Spring Boot se sont aussi révélées exploitables. Ces listes ne sont pas exhaustives.

IV. Menaces actives

Au 17 novembre 2026, aucune exploitation dans la nature n'a été confirmée et la faille ne figure pas encore au catalogue KEV de la CISA. Toutefois, la cause profonde et la technique

d'exploitation sont publiques, et des détails ont fuité avant la publication du correctif. Les chercheurs indiquent avoir obtenu un premier exploit fonctionnel en quelques heures seulement.

Il n'existe pas d'exploit universel : chaque produit nécessite une chaîne d'exploitation spécifique. Le risque d'exploitation à court terme reste néanmoins élevé, compte tenu du nombre considérable de serveurs Java exposés sur Internet.

V. Exposition de l'Afrique et du Cameroun

Java est au cœur de nombreux systèmes d'information au Cameroun et en Afrique : plateformes bancaires et de monnaie électronique, systèmes des opérateurs de télécommunications, applications métiers des administrations, portails d'authentification (Keycloak), outils de gestion de projets (Jira, Confluence) et chaînes d'intégration continue (Jenkins).

Les serveurs récents hébergés dans les centres de données ou chez les fournisseurs cloud utilisent fréquemment des processeurs Intel Xeon ou AMD EPYC concernés. Par ailleurs, de nombreux produits et images de conteneurs embarquent leur propre JDK, qui n'est pas mis à jour par les correctifs du système d'exploitation, ce qui augmente le risque que des instances vulnérables restent en production.

VI. Conclusion et actions recommandées

Java4Shell est une vulnérabilité critique qui peut permettre la prise de contrôle à distance de serveurs Java sans authentification. Le CIRT recommande vivement les actions suivantes :

1. Inventorier toutes les applications Java en production et identifier la version du JDK réellement utilisée par chacune, y compris les JDK embarqués dans les produits et les conteneurs.
2. Vérifier si les serveurs disposent des instructions AVX512 VBMI (drapeaux avx512f, avx512bw, avx512vbmi et bmi2 dans /proc/cpuinfo sous Linux).
3. Mettre à jour sans délai vers les builds corrigés : 17.0.21.1, 21.0.13.1, 25.0.5.1, 27.0.1.1 ou la version corrigée fournie par votre distribution OpenJDK.
4. En attendant le correctif, désactiver le décodeur Base64 optimisé avec l'option JVM « -XX:-UseBASE64Intrinsics », puis redémarrer chaque processus Java. Ne pas utiliser l'option « -XX:UseAVX=2 » sur des serveurs ARM, où la JVM refuserait de démarrer.
5. Prioriser les applications exposées sur Internet, en particulier Keycloak, Jira, Confluence, Jenkins, Artifactory et PingFederate.
6. Surveiller les plantages inhabituels de la JVM (fichiers hs_err_pid) et les requêtes contenant des données Base64 anormales.

7. Après correction, renouveler les secrets, clés et jetons des applications exposées si une compromission est suspectée.

Pour tout incident ou toute demande d'assistance, contactez le CIRT via ses canaux habituels.

VII. Références

- CISA, Critical Java Vulnerability Requires Immediate Patching : www.cisa.gov
- Qualys, Java4Shell CVE-2026-114938 : blog.qualys.com
- Endor Labs, Java4Shell CVE-2026-114938 : www.endorlabs.com
- Upwind, Java4Shell, What We Know Now : www.upwind.io