Java4Shell (CVE-2026-114938): who is affected and how to patch it

27 November 2026 · 5 min read · blogs.iictp.org/java4shell-cve-2026-114938-patch-guide

Short answer: Java4Shell (CVE-2026-114938) is a critical memory-corruption flaw in the Base64 decoder of the x86-64 HotSpot Java virtual machine, disclosed by the security company Calif on 17 November 2026. OpenJDK published an advisory and Oracle released patched builds the same day, and the US agency CISA urged immediate patching. It matters because malformed Base64 sent by an unauthenticated user can lead to remote code execution.

TL;DR The short version

- Java4Shell (CVE-2026-114938) is a critical memory-corruption flaw in the Base64 decoder of the x86-64 HotSpot JVM, disclosed on 17 November 2026.
- OpenJDK and Oracle published fixes the same day, and CISA urged immediate patching because malformed Base64 from an unauthenticated user can lead to remote code execution.
- Firewall rules are not a reliable workaround: find every Java runtime, patch internet-facing systems first, and rebuild images.

Read online, listen to it, or change the text size: https://blogs.iictp.org/java4shell-cve-2026-114938-patch-guide

What is Java4Shell?

Base64 turns binary data into text, and Java applications decode it in authentication headers, tokens, uploads and network messages. To go faster, OpenJDK replaces busy code with hand-written machine code called an intrinsic. Cloudflare, summarising the disclosure, says the flaw sits in the part of that routine that handles the end of the input. It reports a CVSS 4.0 score of 9.2.

How the flaw works in outline

Certain invalid padding makes the routine compute a negative output length, which results in a write past the end of the output array. Java then rejects the input and throws an exception, but neighbouring memory has already been overwritten. For that reason, validation that runs after decoding does not help. Building a working exploit needs work specific to each application.

Who is affected

- **Runtime:** OpenJDK 17.0.6 or newer, including distributions built from it. Cloudflare names Amazon Corretto, Eclipse Temurin, Microsoft Build of OpenJDK, Azul Zulu, BellSoft Liberica, SAP SapMachine, Red Hat OpenJDK and Oracle builds.

- **Processor:** x86-64 with AVX-512 VBMI support, found in several recent Intel Xeon and AMD Zen 4 or newer parts. Cloudflare says ARM64 is not affected.
- **Input:** the application decodes attacker-controlled data with the standard Java Base64 decoder.
- Check Oracle's advisory for the exact list of affected Oracle builds.

Impact reported by the sources

- Cloudflare reports that Calif demonstrated remote code execution without authentication against products including Keycloak, Jira, Confluence, Artifactory, Jenkins and PingFederate in default configurations.
- It also reports a client-side case: a Java program opening a TLS connection to a malicious server can be compromised even if it exposes no service.
- Cloudflare said it had seen no sign of exploitation on its own infrastructure when it wrote.
- CISA lists unexpected JVM crashes, rogue runtimes and suspicious Java processes as signs to watch for.

Limitations of firewall rules and workarounds

- A web application firewall cannot see Base64 wrapped in another encoding, traffic on non-HTTP protocols, outbound connections or traffic that bypasses it.
- Cloudflare enabled its new rules in log mode because some legitimate token formats can look similar.
- The JVM options -XX:-UseBASE64Intrinsics or -XX:UseAVX=2 remove the vulnerable path according to Calif's documentation. The second applies only to x86-64, and both can slow Base64-heavy work.
- Workarounds buy time. They do not replace the update.

Sources were read on 27 November 2026. Products, rules and data change, so check the linked originals before you rely on any detail.

How to patch Java4Shell

1. Inventory every Java runtime, including those bundled in applications, container images and appliances. They may not show up as separate software entries.
2. Patch internet-facing systems first with the November 2026 Java updates or your vendor's fixed build.
3. Rebuild container images that pin an older JDK. Updating the host package does not change them.
4. Restart every service, because a running JVM keeps the old code.
5. Where patching must wait, apply a JVM option, limit unauthenticated access and enforce input size limits at the boundary.
6. Review crash logs and process activity since the disclosure.

What it means for platform and application teams

The hard part is finding copies of the runtime. A software bill of materials and an image scanner turn that search from weeks into a query. Record which services run on x86-64 with the affected instruction sets, but do not use hardware as a reason to skip patching. See what is an SBOM for the inventory side.

Frequently asked questions

- **What is Java4Shell?** CVE-2026-114938, a memory-corruption flaw in the HotSpot Base64 decoder that can allow remote code execution.
- **Are ARM servers affected?** Cloudflare says ARM64 processors are not affected.
- **Does input validation in my code fix it?** Checks after decoding do not, because memory is already overwritten.
- **Is a WAF enough?** No. It covers only some request paths. Update the Java runtime.

Related reading

- What is an SBOM?
- NGINX2Root and nginx patching
- Shirudo eBPF shields for unpatched Linux
- Certified Technical Practitioner

Related credential: Certified Technical Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. CISA. Critical Java Vulnerability Requires Immediate Patching. 17 November 2026. cisa.gov/news-events/alerts/2026/11/17/critical-java-vulnerability-requires-immediate-patching
2. Cloudflare. How Cloudflare WAF protects against Java4Shell (CVE-2026-114938). 18 November 2026. blog.cloudflare.com/java4shell-waf-protection

Published by IICTP, an independent technical credential body, on 27 November 2026. The current version of this post is at https://blogs.iictp.org/java4shell-cve-2026-114938-patch-guide. More guides: blogs.iictp.org.