Waratek releases virtual patch for Java4Shell, the critical Java Base64 flaw now under active attack

The patch closes CVE-2026-114938 inside the JVM with no restart, no code changes and no performance penalty, while teams roll out vendor JDK updates.



Waratek logo

DUBLIN, IRELAND, November 24, 2026 /EINPresswire.com/ -- Waratek today announced a virtual patch for Java4Shell (CVE-2026-114938), the critical memory-corruption flaw in the HotSpot Java Virtual Machine's Base64 decoder that is now under active exploitation. The patch closes the flaw at runtime with no JVM restart, no source code changes, no recompilation and no downtime, and without the performance cost of switching off the JVM's Base64 optimisation.

> Legacy patching cycles are simply too slow for critical threats like Java4Shell. Virtual runtime patches give you instant protection without altering a single line of code or taking systems offline."
>
> *Apostolos Giannakidis, Waratek CTO*

Updating the JDK remains the full fix, but many organisations cannot update immediately. Most companies run free Java distributions, so many teams may face a patching gap of days or longer. The Waratek virtual patch closes that gap and can stay in place after the JDK is updated.

"Legacy patching cycles are simply too slow for critical threats like Java4Shell," said Apostolos Giannakidis, CTO of Waratek. "Our virtual runtime patch immediately neutralizes CVE-2026-114938 inside the JVM, giving organizations instant, foolproof protection without altering a single line of code or taking systems offline."

About the vulnerability

Java4Shell was disclosed on 17 November 2026, when Oracle and the OpenJDK project shipped fixes and the security firm Calif, which discovered the flaw and named it, published its research. The bug is a heap out-of-bounds write in the optimised Base64 decoding routine that HotSpot's JIT compiler uses on x86-64 processors with AVX-512 VBMI support.

The routine miscalculates the store mask for the padded tail of the input, so it writes past the end of the decoded output array. In Calif's demonstration, a 65-character string led to a 50-byte result array being overwritten with 64 bytes. Calif reports unauthenticated remote code execution against widely used Java products.

Oracle's list of affected releases includes Java 17, 21, 25 and 27 and GraalVM, and Calif reports that OpenJDK from 17.0.6 onward is affected. Oracle's list names only one Java 8 build, the Enterprise Performance Pack build 8u511b33-perf, and does not list Java 11. ARM64 systems and x86-64 CPUs without AVX-512 VBMI are not affected. Oracle rates the flaw 8.1 (High) under CVSS 3.1, reflecting high attack complexity, while Calif scores it 9.2 (Critical) under CVSS 4.0. CISA has urged immediate patching.

An attacker needs only to send a crafted Base64 string anywhere an application decodes attacker-supplied input, such as an Authorization header, a JWT segment, a SAML assertion or a file upload. If the JVM has JIT-compiled the decoder, which is true of any busy service, the vulnerable routine runs and corrupts the heap. A failed attempt usually crashes the JVM. A successful one can give the attacker code execution.

How the virtual patch works

The flaw sits in HotSpot's native Base64 routine, which goes wrong only when the input ends in a short padded tail. Just before the JDK's Java code calls that routine, the Waratek virtual patch detects and neutralises the Base64 payload by shortening the input to whole groups of four characters, and the JDK's own Java code decodes the rest as it always did. Decoding results and error messages do not change. Waratek tested the patch on JDK 17, 21 and 25 on a processor with AVX-512 VBMI.

Waratek's file, network and process hardening rules add a second layer. With them enabled, attackers cannot compromise the system by spawning OS processes, writing executable files or opening unexpected outbound connections.

Why the decision has to be made inside the JVM

Java4Shell is hard for conventional tools to catch. There is no vulnerable dependency for software composition analysis to flag and no line of application code for static analysis to point at, so a clean scan result says nothing about this bug. A web application firewall does not know which JDK runs behind it, whether a payload will reach the vectorised decoder, or what the buffer offsets are at execution time. Updating an application also does not update the runtime beneath it, and JDKs bundled in container images and application distributions will not appear in package manager queries.

Waratek RASP operates inside the JVM rather than at the network perimeter. Because it enforces on what the application is actually doing at runtime, it addresses classes of unsafe behaviour including deserialisation of untrusted data, server-side request forgery, path traversal, command

injection, JNDI and LDAP lookups, SQL injection and XML external entity processing, independent of which specific CVE number is behind the behaviour. Waratek applies these controls with no source code changes, no recompilation and no downtime, so exposure can be reduced the day an advisory lands while regression testing and vendor patching proceed on a schedule the business can absorb.

Get the virtual patch
The virtual patch for CVE-2026-114938 is available now to Waratek customers. Details are at vrt.services.waratek.com/cve/CVE-2026-114938. Waratek recommends that organisations inventory their runtimes by vendor and exact build string, prioritise internet-facing Java services on affected hardware, and apply vendor JDK updates as they become available.

Doug Ennis
Waratek
DougE@waratek.com
Visit us on social media:
LinkedIn

This press release can be viewed online at: https://www.einpresswire.com/article/958576982

EIN Presswire's priority is source transparency. We do not allow opaque clients, and our editors try to be careful about weeding out false and misleading content. As a user, if you see something we have missed, please do bring it to our attention. Your help is welcome. EIN Presswire, Everyone's Internet News Presswire™, tries to define some of the boundaries that are reasonable in today's world. Please see our Editorial Guidelines for more information.
© 1995-2026 Newsmatics Inc. All Right Reserved.