CHECK POINT

EXPOSURE MANAGEMENT

CHECK POINT EXPOSURE MANAGEMENT RESEARCH

Java4Shell: Critical Flaw in the Java Runtime

How one malformed string can let an unauthenticated attacker corrupt memory, steal secrets, and run code inside Java 17, 21, 25, and 27.

CVE-2026-114938

November 18, 2026

Critical | Remote code execution | No login required

INSIDE THIS ADVISORY

Contents

01 **Executive Summary** 3
Our Recommendation 3
02 **At a Glance** 4
03 **How Java4Shell Works** 5
Why No Login Is Needed 5
Current Limits on Attackers 5
04 **Exposure and Scope** 6
Which Applications Are at Risk 6
Are You Affected? 6
What About Tomcat, Spring, Elasticsearch, and Atlassian? 7
05 **Remediation** 8
Fixed Versions 8
Temporary Mitigation 8
06 **Recommended Action Plan** 9
07 **Detection and Hunting** 10
08 **Timeline** 10
→ **Contact Us** 11

PUBLISHED BY
Check Point Exposure Management Research Team

DATE
November 18, 2026

THREAT TYPE
Remote code execution · Memory corruption

SOFTWARE AFFECTED
OpenJDK-based Java runtimes

© 2026 Check Point Software Technologies Ltd. All rights reserved.

OVERVIEW

01 Executive Summary

A single malformed string can put your most trusted systems at risk. No login, no malware, no flaw in your own code required.

Java4Shell (CVE-2026-114938) is a critical flaw in the Java runtime itself, not in any one application or library. That means a service can be exposed even when its own code is flawless.

The bug lives in a fast Base64 decoder that Java uses on recent Intel and AMD server processors. A short, malformed Base64 input makes the decoder write past the end of its buffer. With careful preparation, an attacker can turn that into reading secrets from memory and running their own code. Because servers often decode Base64 before they check who you are, no login is needed in many cases.

9.2	6	17.0.6+	0
CVSS v4.0 score assigned by the discovering researchers	Applications with demonstrated pre-authentication code execution	Java 17 builds affected, plus all of Java 21, 25, and 27	Confirmed attacks in the wild as of November 18, 2026

Researchers have demonstrated unauthenticated code execution against default installations of Keycloak, Atlassian Jira, Atlassian Confluence, JFrog Artifactory, Jenkins, and PingFederate. These are identity, collaboration, and software build platforms. Other systems trust what they produce, so a compromise here can spread fast.

Oracle and the OpenJDK project released fixes on November 17, 2026. We're not aware of confirmed exploitation yet. That window won't stay open long: the root cause is public, and a triggering input has been visible since September.

OUR RECOMMENDATION

Update the Java runtime on internet-facing services now, starting with identity providers and CI/CD systems. If you can't update today, apply the temporary JVM setting in section 05 and restart the service. Then work through the rest of your estate in priority order.

© 2026 Check Point Software Technologies Ltd. All rights reserved.

AT A GLANCE

02 Key Facts

CVE ID

CVE-2026-114938

Publicly known as Java4Shell

SEVERITY

Critical

CVSS v4.0 9.2 (researchers) · CVSS v3.1 8.1 (Oracle) · 9.8 preliminary (Red Hat) · 8.8 (Amazon)

FLAW TYPE

Out-of-bounds heap write

Memory corruption that leads to memory disclosure and remote code execution

AUTHENTICATION

None required

Demonstrated on six widely used applications in their default configuration

COMPONENT

Java runtime Base64 decoder

A processor-optimized code path shared by every OpenJDK-based distribution

AFFECTED

Java 17.0.6+, 21, 25, and 27

Unpatched builds running on x86-64 processors with AVX-512 VBMI

FIX STATUS

Available

Oracle, OpenJDK, Amazon, and Azul have shipped fixes. Red Hat and some other vendors are still pending

EXPLOITATION

None confirmed

As of November 18, 2026. Not listed in the CISA Known Exploited Vulnerabilities catalog

BOTTOM LINE

Patch internet-facing Java services first

Identity providers and CI/CD systems carry the highest business risk. Where you can't patch today, a single startup flag closes the vulnerable path until you can

© 2026 Check Point Software Technologies Ltd. All rights reserved.

TECHNICAL ANALYSIS

03 How Java4Shell Works

Java speeds up common operations with hand-tuned processor code, known as intrinsics. On x86-64 chips that support the AVX-512 VBMI instruction set, Java switches to an accelerated Base64 decoder once that code runs often enough to be compiled. Busy servers reach that point quickly.

STEP 1	STEP 2	STEP 3	STEP 4	RESULT
Malformed input	**Fast decoder**	**Overflow**	**Heap shaping**	**Read, write, run**
Short Base64 string with extra padding	AVX-512 code path handles it	62–63 bytes written past the buffer	Neighboring array is enlarged	Secrets exposed, code

Java rejects the malformed input, but only after the out-of-bounds write. The overflowing bytes are fixed, so early analysis pointed to a crash. The researchers showed otherwise: by arranging the server's memory, they use the overflow to enlarge a neighboring array, gain read and write access to memory, and from there run code.

1. **Remote code execution** · Full takeover
   Full control of the Java process and every system that trusts it.

2. **Memory disclosure** · Before any code runs
   Private keys, session tokens, credentials, and personal data can leak from memory.

3. **Denial of service** · Crash on failure
   Failed attempts often crash the Java process, and attackers can simply try again.

4. **Client-side exposure** · Outbound TLS
   A Java program can be compromised just by connecting over TLS to a malicious server.

ATTACK PATH

Why no login is needed

Applications decode Base64 everywhere: HTTP Basic authentication headers, SAML artifacts, CSRF tokens, and file uploads. Much of that happens before the application checks whether you're signed in, giving an unauthenticated attacker a direct path to the vulnerable code.

REALITY CHECK

Current limits on attackers

There's no universal exploit. Each product needs its own memory layout and timing. That slows mass exploitation, but it doesn't protect a specific, high-value target such as your identity provider. Independent teams have already reproduced the memory corruption in lab conditions.

© 2026 Check Point Software Technologies Ltd. All rights reserved.

EXPOSURE

04 Exposure and Scope

Which Applications Are at Risk

The disclosing researchers achieved code execution against the products below in their default configurations. They stress the list is not complete. They also reached code execution in sample applications built on Apache Struts and Spring Boot. Treat any Java service that accepts Base64 from users as potentially exposed.

ACCESS REQUIRED	DEMONSTRATED PRODUCTS
None (pre-authentication)	Keycloak, Atlassian Jira, Atlassian Confluence, JFrog Artifactory, Jenkins, PingFederate
Authenticated user	Apache Kafka broker, Apache Solr, Elasticsearch, Gerrit, Spinnaker

Are You Affected?

A system is vulnerable only when all three conditions are true.

CONDITION	AFFECTED	NOT AFFECTED
Java runtime	Unpatched OpenJDK-based builds from 17.0.6 onward, and all Java 21, 25, and 27 builds. Also Java SE 8 Performance Pack 8u511b33-perf and GraalVM 17.0.21.0.3, 21.0.13.0.3, and EE 21.3.20.0.3	Java 17 releases before 17.0.6, standard Java 8 and Java 11, and any build that contains the fix
Processor	x86-64 with AVX-512 VBMI: Intel Xeon Ice Lake, Sapphire Rapids, Emerald Rapids, and Granite Rapids; AMD EPYC and Ryzen, Zen 4 and newer	ARM64, and x86-64 processors without AVX-512 VBMI
Reachability	Attacker-controlled Base64 reaches Java's standard decoder, for example through login headers, tokens, or uploads	No path for external input to reach the standard Base64 decoder

The runtime condition applies to every OpenJDK-based distribution, from Oracle and Red Hat to Temurin, Corretto, Zulu, Liberica, SapMachine, and Microsoft. To check the processor on Linux, run the command below. The host is affected only if all four flags appear.

LINUX

```
grep -m1 '^flags' /proc/cpuinfo | tr ' ' '\n' | grep -E '^(avx512f|avx512bw|avx512vbmi|bmi2)$'
```

BEFORE YOU CLOSE ANY FINDINGS

Check the runtime each application actually uses. Bundled runtimes don't get OS updates. **Use the processor check to prioritize, not to dismiss.** A cloud resize can move a workload onto an affected processor without warning.

© 2026 Check Point Software Technologies Ltd. All rights reserved.

What About Tomcat, Spring, Elasticsearch, and Atlassian?

Java4Shell lives in the runtime, so no product is safe just because it isn't on a list. Here's what we know about the Java technologies customers ask about most, and what to do about each.

TECHNOLOGY	WHAT WE KNOW	WHAT TO DO
Atlassian Jira and Confluence CONFIRMED · NO LOGIN NEEDED	Researchers achieved code execution without a login on default installations. The risk applies to self-managed Data Center deployments; for Atlassian Cloud, Atlassian manages the Java runtime. Installers typically ship their own bundled runtime	Patch first, right after identity providers. Apply Atlassian's update or the temporary startup flag. Operating system patches alone won't fix a bundled runtime
Elasticsearch CONFIRMED · LOGIN NEEDED	Researchers achieved code execution with a login. Elasticsearch ships with a bundled Java runtime by default. The attack works in a single complete request, so reverse-proxy request buffering doesn't stop it	Treat clusters with anonymous access or weak credentials as exposed without a login. Apply Elastic's update, or add the startup flag to the JVM options and restart each node
Spring and Spring Boot DEMONSTRATED IN SAMPLE APPS	Researchers achieved code execution in sample Spring Boot applications. Exposure depends on each app: login handling, tokens, and file uploads often decode Base64 before a user is authenticated	Treat every Spring Boot service on an affected runtime as in scope. Check the runtime inside each container image, then patch or apply the flag
Apache Tomcat NOT TESTED · IN SCOPE	Not tested by the researchers, and no advisory links Tomcat itself to this CVE. Tomcat runs a large share of Java web applications and is embedded in Spring Boot by default	Include every Tomcat server in your inventory. The risk follows the applications deployed on it and the Java runtime underneath

ONE RULE FOR EVERYTHING ELSE

Kafka, Solr, Jenkins, Kotlin and Scala services: anything on the Java runtime is in scope when three things are true. It runs an affected Java build, on an affected processor, and attacker input can reach Java's built-in Base64 decoder. If you can't rule out the third, assume it's true.

© 2026 Check Point Software Technologies Ltd. All rights reserved.

REMEDIATION

05 Fix and Mitigate

Fixed Versions

Updating the Java runtime is the only complete fix. The table reflects vendor status as of November 18, 2026. Always confirm against your vendor's own advisory. Products that bundle their own Java runtime need an update from the product vendor.

DISTRIBUTION	FIXED RELEASE	STATUS
Oracle Java SE 17 / 21 / 25 / 27	17.0.21.1 · 21.0.13.1 · 25.0.5.1 · 27.0.1.1	Available
Oracle Java SE 8 Performance Pack	8u513-perf	Available
Oracle GraalVM	GraalVM for JDK 17.0.21.1 and 21.0.13.1; GraalVM Enterprise Edition 21.3.20.1	Available
Amazon Corretto 17, 21, 25, 27	17.0.21.1.2 · 21.0.13.1.2 · 25.0.5.1.2 · 27.0.1.1.4	Available
Azul Zulu 17, 21, 25, 27	17.70.201 · 21.54.201 · 25.38.201 · 27.30.201	Available
SAP SapMachine	17.0.21.1 · 21.0.13.1 · 25.0.5.1 · 27.0.1.1 (CVE not named in notes)	Confirm with SAP
Red Hat Enterprise Linux 8, 9, 10; Red Hat build of OpenJDK 17, 21, 25	Not yet released. Rebuild containers once updated images ship	Pending
Eclipse Temurin, Microsoft Build of OpenJDK, BellSoft Liberica	Not yet listed	Check vendor

Temporary Mitigation

Can't update today? Add one of these Java startup options, restart, and remove it after patching.

OPTION	WHAT IT DOES	THINGS TO KNOW
`-XX:-UseBASE64Intrinsics` **Preferred**	Turns off only the accelerated Base64 decoder. Everything else runs as before	Applications that decode large volumes of Base64 may slow down. Test first
`-XX:UseAVX=2` **Alternative**	Turns off AVX-512 entirely, which removes the vulnerable code along with other optimizations	Broader performance cost. x86-64 only: Java on ARM64 rejects this setting and won't start

DON'T SKIP THESE STEPS

Fixes and flags work only after Java restarts, so verify the running process. A reverse proxy blocks some attack chains, not all. Don't use `-XX:DisableIntrinsic=_base64_decodeBlock` alone: it can stop Java from starting.

© 2026 Check Point Software Technologies Ltd. All rights reserved.

WHAT TO DO NEXT

06 Recommended Action Plan

01 Build your inventory

Find every Java runtime in your environment, including runtimes bundled inside commercial products, application packages, and container images.

02 Scope the hardware

Run the processor check on each host. Flag every affected host, and every workload that could be scheduled onto one.

03 Patch in priority order

Identity providers (Keycloak, PingFederate) first, then Jira, Confluence, Jenkins, and Artifactory. Next, other internet-facing Java services that handle logins, tokens, or uploads. Then internal systems and Java clients.

04 Mitigate where you must wait

Apply the preferred startup flag to any system still waiting on a vendor build. Assign an owner and a deadline to each one.

05 Restart and verify

Restart every Java process. Rebuild and redeploy container images. Confirm the running version and flags with `jcmd <pid> VM.version` and `jcmd <pid> VM.flags`.

06 Reduce exposure

Restrict internet access to Java services and administrative interfaces that don't need it.

07 Rotate secrets if you find signs of attack

Memory disclosure can happen before any code runs. If an exposed system shows the warning signs in section 07, rotate its keys, tokens, and credentials.

08 Remember one rule

A patch isn't applied until Java restarts.

© 2026 Check Point Software Technologies Ltd. All rights reserved.

HUNTING

07 Detection and Hunting

These signals can point to exploitation attempts. None of them proves a compromise, and their absence doesn't prove a system is safe.

HTTP HEADERS

Malformed Base64 tokens

Values of 32 characters or more that end in = and whose length isn't a multiple of four, especially in Authorization headers

APPLICATION LOGS

Decoder exception bursts

Spikes of `IllegalArgumentException` errors raised by `java.util.Base64$Decoder`

HOST

Unexpected JVM crashes

New `hs_err_pid` crash logs, or bursts of Java process restarts

NETWORK

High-volume login noise

Large numbers of malformed login or token requests from a single source in a short window

TIMELINE · 2026

08 Chronology

SEPTEMBER 9, 2026

A proposed code change submitted publicly to OpenJDK describes an overflow in the same AVX-512 Base64 decoder and includes a triggering input.

EARLY NOVEMBER 2026

That proposal is closed without being merged after a period of inactivity.

NOVEMBER 17, 2026

Oracle and the OpenJDK project release fixes. The discovering researchers publish their disclosure. Amazon releases fixed Corretto builds.

NOVEMBER 18, 2026

Azul fixes are available and Red Hat fixes are pending. The Canadian Centre for Cyber Security issues an alert. No confirmed exploitation in the wild, and the CVE is not in the CISA Known Exploited Vulnerabilities catalog.

This advisory reflects information available as of November 18, 2026. Details may change as vendors publish fixes and as threat activity develops. It is provided for information only and does not replace guidance from your software vendors.

© 2026 Check Point Software Technologies Ltd. All rights reserved.

CHECK POINT

EXPOSURE MANAGEMENT

Contact Us

ISRAEL

Tel: +972-73-226-4555
5 Shlomo Kaplan Street, 6789159

USA

Tel: 1-800-429-4391
100 Oracle Parkway, Suite 800
Redwood City, CA 94065

SINGAPORE

Tel: +65-6435-1318
78 Shenton Way, #09-01 Tower 1
Singapore 079120

PHILIPPINES

Tel: +63-2-8465-9200
Unit 2005, 20th Floor, Zuellig Building, Makati Avenue,
corner Paseo de Roxas, Makati City 1223, Metro Manila

UK AND IRELAND

Tel: +44-20-7628-4211
85 London Wall, 4th Floor
London, EC2M 7AD

JAPAN

Tel: +81-3-6205-8340
Toranomon Kotohira Tower 25F, 1-2-8, Toranomon,
Minato-ku, Tokyo 105-0001

For more information visit: **checkpoint.com**

© Check Point, 2026. All Rights Reserved.