Hive Pro

Threat Level

Red

Hiveforce Labs

THREAT ADVISORY

VULNERABILITY REPORT

FastJson 1.x's Parting Gift: An Unpatched, Unauthenticated RCE

Date of Publication	Admiralty Code	TA Number
July 29, 2026	A1	TA2026215

Summary

First Seen: July 2026
Targeted Countries: US, Singapore, and Canada
Targeted Industries: Financial Services, Healthcare, Computing, Retail, Business, Entertainment, Telecom, and ISPs
Affected Products: Alibaba FastJson 1.2.68 through 1.2.83 (Spring Boot executable fat-JAR deployments)
Impact: A critical remote code execution flaw, CVE-2026-16723, has been disclosed in FastJson, Alibaba's widely used Java library for reading and writing JSON, and attackers are already exploiting it in the wild; the weakness lets an unauthenticated attacker run code on a vulnerable server simply by sending it a malicious JSON message, it works under FastJson's default settings without AutoType turned on or any third-party gadget class present, it affects versions 1.2.68 through 1.2.83 (the last of the 1.x line) when the application runs as a Spring Boot executable fat JAR, it was confirmed actively exploited against organizations across financial services, healthcare, computing, retail, and business sectors (mostly in the United States, with smaller volumes in Singapore and Canada) and, most importantly, there is no patched 1.x release available, leaving migration to FastJson 2.x and immediate mitigation as the only real fixes.

CVE

CVE	NAME	AFFECTED PRODUCT	ZERO-DAY	CISA KEV	PATCH
CVE-2026-16723	Alibaba FastJson Remote Code Execution Vulnerability	Alibaba FastJson	✓	✗	✗

Vulnerability Details

#1

Fastjson is an open-source Java library, originally developed by Alibaba, used to convert Java objects into JSON and to parse JSON back into Java objects. CVE-2026-16723 is a remote code execution flaw in Fastjson 1.x, caused by the way the library turns incoming JSON into Java objects. Fastjson relies on a special type-identifier field within the JSON to decide which Java class to create. Version 1.x turns off AutoType by default and adds checks to block untrusted classes, yet researchers found a way around them: the library's type-resolution logic performs attacker-controlled lookups before those AutoType restrictions ever take effect.

#2

In an attack, an adversary sends JSON carrying a malicious type value. Fastjson then looks up a resource based on that attacker-chosen name, and in a Spring Boot fat-JAR application the attacker can abuse nested JAR URL handling to pull in their own code. A type annotation placed on that code is treated as trustworthy, so it slips past the checks and runs with no traditional deserialization gadget needed. A separate newer-JDK variant instead downloads a remote JAR and loads it through a system file-descriptor reference. The flaw is reachable through everyday JSON parsing calls, and even pinning the target class does not fully help, because the payload can hide inside an object or map field.

#3

The flaw affects Fastjson 1.2.68 through 1.2.83, including 1.2.83, the last release in the 1.x line. It was confirmed on Spring Boot 2.x, 3.x, and 4.x with JDK 8, 11, 17, and 21, and the one firm requirement is that the application runs as a Spring Boot executable fat JAR launched directly from the command line. Versions 1.2.60 and earlier, plain non-fat JARs, uber-JARs, and Tomcat or Jetty WAR deployments are not affected, and neither is Fastjson 2.x, which uses an allowlist-first approach and does not treat the type annotation as a trust signal.

#4

Attacks so far have hit financial services, healthcare, computing, retail, and business-services organizations, almost entirely in the United States, with smaller numbers in Singapore and Canada, and this is expected to spread as public proof-of-concept details circulate. In-the-wild activity was reported from the week of July 20, though a July 23 CISA-ADP assessment still marked exploitation as "none," meaning the reporting reflects attempted exploit activity rather than a confirmed breach. Either way, this is a high-severity, actively targeted flaw in an end-of-life library with no patch coming, so exposed deployments should be treated as urgent and moved off Fastjson 1.x.

Vulnerability

CVE ID	AFFECTED PRODUCTS	AFFECTED CPE	CWE ID
CVE-2026-16723	Alibaba FastJson 1.x (1.2.68 through 1.2.83)	cpe:2.3:a:alibaba:fastjson:*:*:*:*:*:*:*:*	CWE-502, CWE-20

Recommendations

Enable FastJson SafeMode Immediately: If you cannot migrate right away, turn on SafeMode using the JVM option -Dfastjson.parser.safeMode=true, the fastjson.parser.safeMode=true property, or ParserConfig.getGlobalInstance().setSafeMode(true). Alternatively, switch to the com.alibaba:fastjson:1.2.83_noneautotype build, which strips the vulnerable AutoType-related code at compile time. This is the fastest way to close the exposure while you plan a full move.

Hunt for Signs of Exploitation: Review application and WAF logs for suspicious JSON requests containing "@type" fields and jar:http or jar:file URL patterns, and inspect affected systems for unexpected process execution, unusual outbound connections, unauthorized file changes, and web shells. Because the exploit fetches attacker-controlled bytecode, treat any matching activity as a potential compromise and investigate accordingly.

Inventory Your FastJson Dependencies: Identify both direct and transitive FastJson references across your Java and Spring Boot applications, since the library is often pulled in indirectly. You cannot mitigate what you do not know you are running, so an accurate dependency inventory is the foundation for everything else.

Migrate to FastJson 2.x: FastJson 1.x is no longer actively maintained and will not receive a fix for this issue, so plan and execute a migration to FastJson 2.x after appropriate compatibility testing. This is the only durable, long-term resolution rather than a stopgap.

Use Perimeter Protection as a Stopgap, Not a Cure: WAF rules that inspect incoming requests for malicious JSON payloads, suspicious @type values, and nested JAR URL patterns can blunt opportunistic scanning while you remediate. Do not treat this as a substitute for fixing the underlying deployment, as motivated attackers can adapt around signatures.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Initial Access	T1190: Exploit Public-Facing Application	
Execution	T1059: Command and Scripting Interpreter	
Persistence	T1505: Server Software Component	T1505.003: Web Shell
Resource Development	T1588: Obtain Capabilities	T1588.006: Vulnerabilities

Patch Details

No vendor patch link is available. Alibaba has confirmed that no fixed release exists for the Fastjson 1.x branch as of the publication of the referenced sources and recommends SafeMode activation or migration to Fastjson 2.x in place of a patch.

https://github.com/alibaba/fastjson2/wiki/Security-Advisory:-Remote-Code-Execution-in-fastjson-1.2.68%E2%80%931.2.83

References

https://www.imperva.com/blog/imperva-customers-protected-against-cve-2026-16723-critical-fastjson-1-x-zero-day-rce/

https://threatbook.io/blog/fastjson-rce-1.2.83-active-exploitation-detected-detection-mitigation

What Next?

At **Hive Pro**, it is our mission to detect the most likely threats to your organization and to help you prevent them from happening.

Book a Demo of **HivePro**.

REPORT GENERATED ON

July 29, 2026 • 09:30 AM

More at www.hivepro.com