Java: which APIs and framework defaults turn into vulnerabilities?

Before your next release, move to JDK 25 or 21 and Spring Boot 4.1[5], stop deserializing outside data into classes it names, and require Find Security Bugs or CodeQL in CI. Judge everything else by whether outside data can pick a class, become SQL, an expression or a command, reach an XML parser, a path or a URL, or slip past an authorization rule.

Reviewed	Next review	Version
5 Nov 2026	**5 Dec 2026**	**1.1**

Written for
Developers and tech leads writing production Java, including Spring Boot services, and the security people who review it

Latest version
sheets.einblick.dev/secure-java/

If you only do five things

1. **Stay on supported lines and take every update.**
   OpenJDK ships fixes each January, April, July and October, plus three extra releases in 2026[3]. Spring Boot 3.5 lost free fixes in June 2026, 4.0 loses them in December and 4.1 in July 2027[,5]. Patch within the patching sheet's deadlines.
2. **Never let outside data name the class.**
   Java deserialization creates the classes the stream names, and no filter runs unless you set one[12]; Jackson's default typing lets JSON do the same[,21]. Read outside data as JSON into records.
3. **Keep input out of queries, expressions and commands.**
   Use parameters for SQL and JPQL, never parse SpEL or OGNL from input, and give `ProcessBuilder` an argument list. SQL, OS command and code injection rank 2nd, 9th and 10th in the CWE Top 25[25].
4. **Deny by default and keep actuator private.**
   End your rules with `.anyRequest().denyAll()`[8] and expose only `health`, Boot's default[9]. ActiveMQ's Jolokia endpoint let logged-in users run code (CVE-2026-34197, exploited)[,23,24].
5. **Make the analyzers a required check.**
   Run Find Security Bugs or CodeQL on every pull request[28,30]; CodeQL on private code needs paid Code Security[32]. Set Dependency-Check's `failBuildOnCVSS`: by default it never fails[31].

Review a Java pull request

1. **Where does outside data enter?**
   Find the controllers, listeners, uploads and batch jobs it touches. Each should bind a record made for that request, as Spring advises[11], not an entity: Spring4Shell was code execution through data binding[,24].
2. **Who may call it, on which record?**
   Ask why for each new `permitAll()`, each matcher placed after a broader one[8] and each `csrf` change. Missing authorization ranks 4th in the CWE Top 25[25]: check who owns the record.
3. **Can data choose a class or code?**
   Search for `ObjectInputStream`, `XMLDecoder`, `enableDefaultTyping`, `activateDefaultTyping`, `Id.CLASS`, `Class.forName`, `parseExpression` and `lookup(`.
4. **Does it reach SQL, a process, XML, a path or a URL?**
   Check `createQuery` and `@Query` strings built with +, `ProcessBuilder`, XML parser factories, `resolve(` and HTTP clients given URLs from input.
5. **What does it expose?**
   Check `exposure.include`, new `SecurityFilterChain` beans, which replace Boot's actuator rules[9], error details and logged secrets.

Java traps and what to use instead

- **`ObjectInputStream.readObject()` and `XMLDecoder`.** The stream picks the classes, and gadgets on the classpath turn that into code execution[12]; OWASP lists `XMLDecoder` as unsafe for untrusted input[26]. If you must read Java serialization, set `jdk.serialFilter` to an allowlist ending in `!*`, with `maxbytes` and `maxdepth` limits[13].
- **`activateDefaultTyping()` and `@JsonTypeInfo(use = Id.CLASS)`.** The JSON then names the class, and Jackson calls the validator you pass critical for security[21]. Use `Id.NAME` with listed subtypes[28]. XStream with a relaxed allowlist, SnakeYAML before 2.0 without `SafeConstructor` and fastjson without safe mode share the problem[26].
- **`DocumentBuilderFactory.newInstance()` defaults.** JDK parsers fetch external DTDs and entities unless you set secure processing explicitly[14], and GeoServer's XXE (CVE-2025-58360) is exploited[,24]. Set `jdk.xml.dtd.support=deny` on JDK 22 and later, or `disallow-doctype-decl` for DOM and SAX and `SUPPORT_DTD` false for StAX[14].
- **`"WHERE id = " + id` in JDBC, JPQL or a native `@Query`.** Hibernate queries are injectable when you concatenate[27]. Bind `?` or `:name` parameters, and map sort columns and directions from a fixed list: parameters can't stand for column names[27].
- **SpEL, OGNL, `Class.forName()` or JNDI `lookup()` fed from input.** Spring says `StandardEvaluationContext` must never evaluate untrusted SpEL and `SimpleEvaluationContext` is only best-effort[10]. Spring Cloud Function (SpEL), Confluence (OGNL) and PaperCut (reflection) flaws are exploited[,24]; Log4Shell ran LDAP code named in a log message[,22]. Keep input in variables; map names from a fixed list.
- **`Runtime.exec(String)` and `sh -c`.** `exec(String)` splits on whitespace and is deprecated since JDK 18[16]. Give `ProcessBuilder` a fixed program and one element per argument, put `--` before input that may start with a dash, and never hand input to `sh -c` or `cmd /c`.
- **`base.resolve(name)`.** An absolute `name` replaces `base`, and `normalize()` ignores symlinks[15]. Require `startsWith(base)` on the normalized path, which compares whole name elements, and on `toRealPath()` where links matter[15]. Check `ZipEntry` names too.
- **`URL.openStream()` and HTTP clients given URLs from input.** URL connections also open `file:` URLs[28]; server-side request forgery ranks 22nd in the CWE Top 25[,25]. Allowlist hosts, check resolved addresses and refuse redirects, the JDK `HttpClient` default[17]. Boot's clients follow them unless `spring.http.clients.redirects` is `dont-follow`[19,18].

Tools that catch it

- **Find Security Bugs.** Add `findsecbugs-plugin` to SpotBugs and fix `OBJECT_DESERIALIZATION`, `XXE_DOCUMENT`, `SQL_INJECTION_JPA` and `SPEL_INJECTION` first[28].
- **Error Prone** flags `BanJNDI` as an error by default; turn on the experimental `BanSerializableRead` and `BanClassLoader` checks[29].
- **CodeQL**'s `java/unsafe-deserialization`, `java/xxe` and `java/jndi-injection` queries cover these traps[30].
- **Dependency-Check** fails builds only once you set `failBuildOnCVSS`[31]. The supply-chain sheet covers the rest.

What you can skip

- **The Security Manager.** JDK 24 disabled it for good, and servers rarely used it[20]. Isolate untrusted code in a process or container.
- **Hunting gadget classes.** OWASP expects more to be found[26]. Stop deserializing outside data instead.
- **Find Security Bugs' inventory patterns.** `SPRING_ENDPOINT` and `SERVLET_PARAMETER` flag every controller and parameter read[28]. Exclude them.
- **Escaping input for SQL, shells or SpEL yourself.** Bind parameters, drop the shell and stop evaluating input instead.

What changed in 2026

- **Oracle JDK 21 updates are no longer no-fee.** Releases since 20 October use the OTN license; Oracle JDK 25 updates stay no-fee through September 2028[1]. Temurin supports 21 until at least December 2029[2].
- **Compromised eMudhra CA keys need another JDK update.** Install a runtime that fixes JDK-8395690 (CVE-2026-114820) in every image, even if you updated in October; on Debian and Ubuntu, also rebuild the Java keystore[4].
- **Spring Boot 4.0.0 to 4.0.5 could open every endpoint** in some apps on Boot's default web security (CVE-2026-40976, critical). 4.0.6 fixes it[6].
- **Free Spring fixes now skip Spring Security 6.5.** August's critical fix for its embedded LDAP server (CVE-2026-59270) shipped free only for 7.0 and 7.1[7].
- **Spring Boot 4.1 added SSRF protection.** An `InetAddressFilter` bean limits the addresses its HTTP clients may call, except through a proxy[18].

What this looks like for your code

New project

- Start on JDK 25 and Spring Boot 4.1, with `.anyRequest().denyAll()` as the last rule.
- Set `jdk.serialFilter=!*` and `jdk.xml.dtd.support=deny` if nothing needs serialization or DTDs.
- Bind every request to a record and keep entities out of controllers.

Existing codebase

- Move Oracle JDK 21 to JDK 25 or an OpenJDK build, and leave JDK 11 and 17 before October 2027[2].
- Search for `ObjectInputStream`, `activateDefaultTyping`, `parseExpression` and `exposure.include`.
- Remove Log4j 1, at end of life since 2015[22].

High-assurance or regulated

- Make CodeQL's `security-extended` suite and Find Security Bugs required checks.
- Run with an allowlist `jdk.serialFilter` and the strict JAXP template as the JAXP configuration file[14].
- Stay on Spring and JDK lines with free fixes, or pay for support on the ones you keep.

Version history

5 Nov 2026, version 1.1 Reviewed for company readers: CodeQL on private code needs a paid GitHub Code Security license.

Sources

1. Oracle Java SE Support Roadmap. Oracle, updated 20 Oct 2026. oracle.com
2. Temurin support roadmap. Adoptium, checked 5 Nov 2026. adoptium.net
3. OpenJDK Vulnerability Advisories. OpenJDK, updated 24 Oct 2026. openjdk.org
4. Compromised Certificates and OpenJDK Java Runtimes. OpenJDK, updated 24 Oct 2026. openjdk.org
5. Spring Boot: support. Spring, checked 5 Nov 2026. spring.io
6. CVE-2026-40976 (Spring Boot). Spring, 23 Apr 2026. spring.io
7. CVE-2026-59270 (Spring Security). Spring, 20 Aug 2026. spring.io
8. Authorize HttpServletRequests. Spring Security, checked 5 Nov 2026. docs.spring.io
9. Actuator: Endpoints. Spring Boot, checked 5 Nov 2026. docs.spring.io
10. SpEL: Evaluation. Spring Framework, checked 5 Nov 2026. docs.spring.io
11. @ModelAttribute. Spring Framework, checked 5 Nov 2026. docs.spring.io
12. Addressing Deserialization Vulnerabilities. JDK 25 docs, checked 5 Nov 2026. docs.oracle.com
13. Creating Pattern-Based Filters. JDK 25 docs, checked 5 Nov 2026. docs.oracle.com
14. JAXP Security Guide. JDK 25 docs, checked 5 Nov 2026. docs.oracle.com
15. Interface Path. JDK 25 API, checked 5 Nov 2026. docs.oracle.com
16. Class Runtime. JDK 25 API, checked 5 Nov 2026. docs.oracle.com
17. Class HttpClient. JDK 25 API, checked 5 Nov 2026. docs.oracle.com
18. Calling REST Services. Spring Boot, checked 5 Nov 2026. docs.spring.io
19. HttpClientSettings. Spring Boot API, checked 5 Nov 2026. docs.spring.io
20. JEP 486: Permanently Disable the Security Manager. OpenJDK, updated 2 Sep 2025. openjdk.org
21. Jackson ObjectMapper 2.22. FasterXML, checked 5 Nov 2026. javadoc.io
22. Apache Log4j security. Apache Logging, checked 5 Nov 2026. logging.apache.org
23. CVE-2026-34197. Apache ActiveMQ, checked 5 Nov 2026. activemq.apache.org
24. Known Exploited Vulnerabilities Catalog. CISA, checked 5 Nov 2026. cisa.gov
25. 2025 CWE Top 25. MITRE CWE, Dec 2025. cwe.mitre.org
26. Deserialization Cheat Sheet. OWASP, checked 5 Nov 2026. cheatsheetseries.owasp.org
27. SQL Injection Prevention Cheat Sheet. OWASP, checked 5 Nov 2026. cheatsheetseries.owasp.org
28. Bug patterns. Find Security Bugs, checked 5 Nov 2026. find-sec-bugs.github.io
29. Bug patterns. Error Prone, checked 5 Nov 2026. errorprone.info
30. CodeQL query help for Java. GitHub, checked 5 Nov 2026. codeql.github.com
31. Maven plugin configuration. OWASP Dependency-Check, 13.0.0, 3 Aug 2026. dependency-check.github.io
32. CodeQL CLI. GitHub Docs, checked 5 Nov 2026. docs.github.com