




 Secrets


 ABAP


 Apex


 C


 C++


 CloudFormation


 COBOL


 C#


 CSS


 Flex


 Go


 HTML


 **Java**


 JavaScript


 Kotlin


 Objective C


 PHP


 PL/I


 PL/SQL


 Python


 RPG


 Ruby


 Scala


 Swift


 Terraform


 Text


 TypeScript

 T-SQL

 VB.NET

 VB6

 XML



Java static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVA code

All rules 632

Vulnerability 53

Bug 154


Security Hotspot 36

Code Smell 389


Quick Fix 42


Tags ▾

Search by name... 🔍


 Bug


InputStream.read() implementation should not return a signed byte







"compareTo" should not be overloaded







"iterator" should not return "this"







Map values should not be replaced unconditionally







Week Year ("YYYY") should not be used for date formatting







Exceptions should not be created without being thrown







Collection sizes and array length comparisons should make sense





Consumed Stream pipelines should not be reused





Intermediate Stream methods should not be left unused

All branches in a conditional structure should not have exactly the same implementation

Optional value should only be accessed after calling isPresent()

Using unsafe Jackson deserialization configuration is security-sensitive

Security Hotspot

Critical

cwe owasp

Using unsafe Jackson deserialization configuration is security-sensitive. It has led in the past to the following vulnerabilities:

- CVE-2017-4995
- CVE-2018-19362

When Jackson is configured to allow Polymorphic Type Handling (aka PTH), formerly known as Polymorphic Deserialization, "deserialization gadgets" may allow an attacker to perform remote code execution.

This rule raises an issue when:

- enableDefaultTyping() is called on an instance of com.fasterxml.jackson.databind.ObjectMapper or org.codehaus.jackson.map.ObjectMapper.
- or when the annotation @JsonTypeInfo is set at class, interface or field levels and configured with use = JsonTypeInfo.Id.CLASS or use = Id.MINIMAL_CLASS.

Ask Yourself Whether

- You configured the Jackson deserializer as mentioned above.
- The serialized data might come from an untrusted source.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

- Use the latest patch versions of jackson-databind blocking the already discovered "deserialization gadgets".
- Avoid using the default typing configuration: ObjectMapper.enableDefaultTyping().
- If possible, use @JsonTypeInfo(use = Id.NAME) instead of @JsonTypeInfo(use = Id.CLASS) or @JsonTypeInfo(use = Id.MINIMAL_CLASS) and so rely on @JsonTypeName and @JsonSubTypes.







Sensitive Code Example

```
ObjectMapper mapper = new ObjectMapper();
mapper.enableDefaultTyping(); // Sensitive

@JsonTypeInfo(use = Id.CLASS) // Sensitive
abstract class PhoneNumber {
}
```

See

- OWASP Top 10 2021 Category A8 - Software and Data Integrity Failures
- OWASP Top 10 2017 Category A8 - Insecure Deserialization
- OWASP - [Deserialization of untrusted data](#)

<p>Overrides should match their parent class methods in synchronization</p> <p> Bug</p>	<ul style="list-style-type: none">• MITRE, CWE-502 - Deserialization of Untrusted Data• On Jackson CVEs: Don't Panic• CVE-2017-1509• CVE-2017-7525• Derived from FindSecBugs rule JACKSON_UNSAFE_DESERIALIZATION <p>Available In:</p> <p> </p>
<p>Value-based classes should not be used for locking</p> <p> Bug</p>	
<p>Expressions used in "assert" should not produce side effects</p> <p> Bug</p>	
<p>"volatile" variables should not be used with compound operators</p> <p> Bug</p>	

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

[Privacy Policy](#)