




 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 rules632

Vulnerability53

Bug154

Security Hotspot36

Code Smell389

Quick Fix42

Tags ▾

Search by name... 🔍

"setComplete" on their "SessionStatus" objects

Bug

"wait" should not be called when multiple locks are held

Bug

"PreparedStatement" and "ResultSet" methods should be called with valid indices

Bug

Files opened in append mode should not be used with ObjectOutputStream

Bug

"wait(...)" should be used instead of "Thread.sleep(...)" when a lock is held

Bug

Printf-style format strings should not lead to unexpected behavior at runtime

Bug

Methods "wait(...)", "notify()" and "notifyAll()" should not be called on Thread instances

Bug

Methods should not call same-class methods with incompatible "@Transactional" values

Bug

Recursion should not be infinite

Bug

Loops should not be infinite

Bug

Double-checked locking should not be used

Bug

Resources should be closed

A secure password should be used when connecting to a database

Analyze your code

Vulnerability

Blocker

cwe owasp

When relying on the password authentication mode for the database connection, a secure password should be chosen.

This rule raises an issue when an empty password is used.

Noncompliant Code Example

```
Connection conn = DriverManager.getConnection("jdbc:derby:me
```

Compliant Solution

```
String password = System.getProperty("database.password");
Connection conn = DriverManager.getConnection("jdbc:derby:me
```

See

- OWASP Top 10 2021 Category A7 - Identification and Authentication Failures
- OWASP Top 10 2017 Category A2 - Broken Authentication
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- MITRE, CWE-521 - Weak Password Requirements

Available In:





sonarlint | sonarcloud | sonarqube

© 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](#)

https://rules.sonarsource.com/java/RSPEC-2115

1/2

 Bug
Hard-coded credentials are security-sensitive  Security Hotspot
Methods returns should not be invariant  Code Smell
"ThreadGroup" should not be used  Code Smell
"clone" should not be overridden