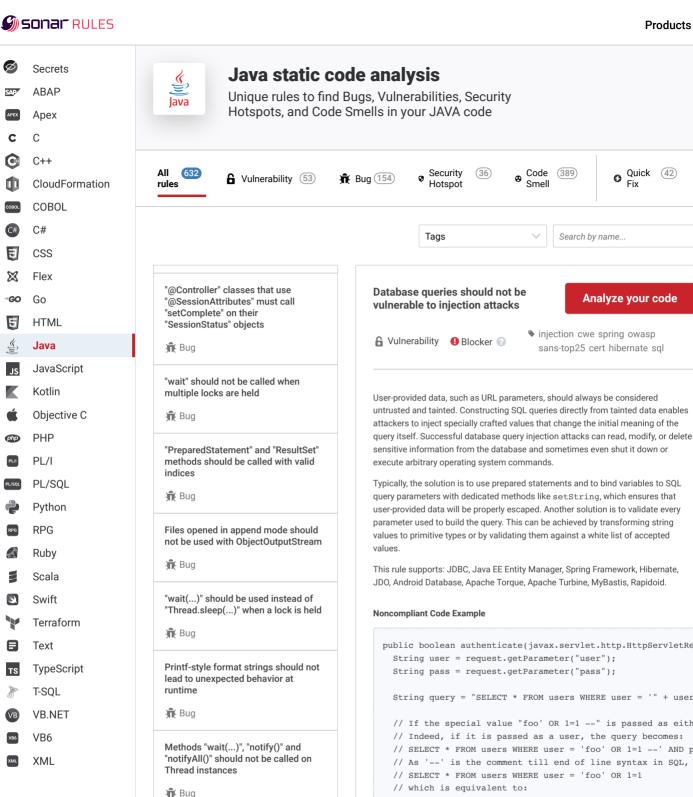


Products ✓

O Quick 42 Fix



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

Recursion should not be infinite

Loops should not be infinite

Double-checked locking should not be

Bug

₩ Bug

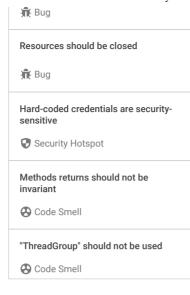
R Bug

used

This rule supports: JDBC, Java EE Entity Manager, Spring Framework, Hibernate, JDO, Android Database, Apache Torque, Apache Turbine, MyBastis, Rapidoid. public boolean authenticate(javax.servlet.http.HttpServletRe String user = request.getParameter("user"): String pass = request.getParameter("pass"); String query = "SELECT * FROM users WHERE user = '" + user // If the special value "foo' OR 1=1 --" is passed as eith // Indeed, if it is passed as a user, the query becomes: // SELECT * FROM users WHERE user = 'foo' OR 1=1 --' AND p // As '--' is the comment till end of line syntax in SQL, // SELECT * FROM users WHERE user = 'foo' OR 1=1 // which is equivalent to: // SELECT * FROM users WHERE 1=1 // which is equivalent to: // SELECT * FROM users java.sql.Statement statement = connection.createStatement(java.sql.ResultSet resultSet = statement.executeQuery(quer return resultSet.next();

Compliant Solution

```
public boolean authenticate(javax.servlet.http.HttpServletRe
 String user = request.getParameter("user");
 String pass = request.getParameter("pass");
 String query = "SELECT * FROM users WHERE user = ? AND pas
 java.sgl.PreparedStatement statement = connection.prepareS
```



statement.setString(1, user); // Will be properly escaped
statement.setString(2, pass);
java.sql.ResultSet resultSet = statement.executeQuery();
return resultSet.next();
}

See

- OWASP Top 10 2021 Category A3 Injection
- OWASP Top 10 2017 Category A1 Injection
- MITRE, CWE-20 Improper Input Validation
- MITRE, CWE-89 Improper Neutralization of Special Elements used in an SQL Command
- MITRE, CWE-564 SQL Injection: Hibernate
- <u>MITRE, CWE-943</u> Improper Neutralization of Special Elements in Data Query Logic
- OWASP SQL Injection Prevention Cheat Sheet
- CERT, IDS00-J. Prevent SQL injection
- SANS Top 25 Insecure Interaction Between Components

Available In:

sonarcloud 🖒 | sonarqube | Developer Edition

© 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