sonar RULES

Products

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...

Abstract class names should comply with a naming convention
Code Smell

Strings literals should be placed on the left side when checking for equality
Code Smell

Files should contain an empty newline at the end
Code Smell

Source code should be indented consistently
Code Smell

A close curly brace should be located at the beginning of a line
Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be on two different lines
Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be located on the same line
Code Smell

An open curly brace should be located at the beginning of a line
Code Smell

An open curly brace should be located at the end of a line
Code Smell

Tabulation characters should not be used
Code Smell

Functions should not be defined with a variable number of arguments
Code Smell

Exceptions should not be thrown from servlet methods

Analyze your code

Vulnerability Minor cwe error-handling cert owasp

Even though the signatures for methods in a servlet include `throws IOException, ServletException`, it's a bad idea to let such exceptions be thrown. Failure to catch exceptions in a servlet could leave a system in a vulnerable state, possibly resulting in denial-of-service attacks, or the exposure of sensitive information because when a servlet throws an exception, the servlet container typically sends debugging information back to the user. And that information could be very valuable to an attacker.

This rule checks all exceptions in methods named "do*" are explicitly handled in servlet classes.

Noncompliant Code Example

```
public void doGet(HttpServletRequest request, HttpServletRes
  throws IOException, ServletException {
  String ip = request.getRemoteAddr();
  InetAddress addr = InetAddress.getByName(ip); // Noncompli
  //...
}
```

Compliant Solution

```
public void doGet(HttpServletRequest request, HttpServletRes
  throws IOException, ServletException {
  try {
    String ip = request.getRemoteAddr();
    InetAddress addr = InetAddress.getByName(ip);
    //...
  }
  catch (UnknownHostException uhex) {
    //...
  }
}
```

See

- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- MITRE, CWE-600 - Uncaught Exception in Servlet
- CERT, ERR01-J. - Do not allow exceptions to expose sensitive information

Available In:

sonarlint sonarcloud sonarqube

Local-Variable Type Inference should be used

Code Smell

Migrate your tests from JUnit4 to the new JUnit5 annotations

Code Smell

Track uses of disallowed classes

Code Smell

Track uses of "@SuppressWarnings" annotations

Code Smell