




 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... 🔍

Similar tests should be grouped in a single Parameterized test

Code Smell

Tests should be stable

Code Smell

Test methods should not contain too many assertions

Code Smell

AssertJ "assertThatThrownBy" should not be used alone

Code Smell

Character classes in regular expressions should not contain the same character twice

Code Smell

Names of regular expressions named groups should be used

Code Smell

Regexes containing characters subject to normalization should use the CANON_EQ flag

Code Smell

Regular expressions should not be too complicated

Code Smell

JUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion

Code Smell

Only one method invocation is expected when testing runtime exceptions

Code Smell

Exception testing via JUnit ExpectedException rule should not be mixed with other assertions

Code Smell

Server-side requests should not be vulnerable to forging attacks

Analyze your code

VulnerabilityMajor🔍injection cwe sans-top25 owasp

User-supplied data, such as URL parameters, POST data payloads, or cookies, should always be considered untrusted and tainted. Performing requests from user-controlled data could allow attackers to make arbitrary requests on the internal network or to change their original meaning and thus to retrieve or delete sensitive information.

The problem could be mitigated in any of the following ways:

- Validate the user-provided data, such as the URL and headers, used to construct the request.
- Redesign the application to not send requests based on user-provided data.

Noncompliant Code Example

```
protected void doGet(HttpServletRequest req, HttpServletResponse
    URL url = new URL(req.getParameter("url"));
    HttpURLConnection conn = (HttpURLConnection) url.openConne
}
```

Compliant Solution

```
protected void doGet(HttpServletRequest req, HttpServletResponse
    String[] urlWhiteList = { "example.com", "www.example.com"

    String inputUrl = req.getParameter("url");

    URI uri          = new URI(inputUrl);
    String remoteHost = uri.getHost();

    if (!urlWhiteList.contains(remoteHost))
        throw new IOException();

    URL url = uri.toURL();
    HttpURLConnection conn = (HttpURLConnection) url.openConne
}
```

See

- [OWASP Top 10 2021 Category A10](#) - Server-Side Request Forgery (SSRF)
- [OWASP Attack Category](#) - Server Side Request Forgery
- [OWASP Top 10 2017 Category A5](#) - Broken Access Control
- [MITRE, CWE-20](#) - Improper Input Validation
- [MITRE, CWE-641](#) - Improper Restriction of Names for Files and Other Resources
- [MITRE, CWE-918](#) - Server-Side Request Forgery (SSRF)
- [SANS Top 25](#) - Risky Resource Management

Available In:

sonarcloud

sonarqube

Developer Edition

https://rules.sonarsource.com/java/RSPEC-5144

1/2

"@Deprecated" code marked for removal should never be used

 Code Smell

Vararg method arguments should not be confusing

 Code Smell

Whitespace for text block indent should be consistent

 Code Smell

'List.remove()' should not be used in ascending 'for' loops

 Code Smell

© 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](#)