




 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

Creating cookies without the "secure" flag is security-sensitive

Analyze your code

Security Hotspot

Minor ⓘ

cwe spring privacy sans-top25 owasp

When a cookie is protected with the `secure` attribute set to `true` it will not be send by the browser over an unencrypted HTTP request and thus cannot be observed by an unauthorized person during a man-in-the-middle attack.

Ask Yourself Whether

- the cookie is for instance a *session-cookie* not designed to be sent over non-HTTPS communication.
- it's not sure that the website contains **mixed content** or not (ie HTTPS everywhere or not)

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

Recommended Secure Coding Practices

- It is recommended to use HTTPs everywhere so setting the `secure` flag to `true` should be the default behaviour when creating cookies.
- Set the `secure` flag to `true` for session-cookies.

Sensitive Code Example

If you create a security-sensitive cookie in your JAVA code:

```
Cookie c = new Cookie(COOKIE_NAME, sensitivedata);
c.setSecure(false); // Sensitive: a security-ensitive cooki
```

By default the **secure** flag is set to `false`:

```
Cookie c = new Cookie(COOKIE_NAME, sensitivedata); // Sensit
```

Compliant Solution

```
Cookie c = new Cookie(COOKIE_NAME, sensitivedata);
c.setSecure(true); // Compliant: the sensitive cookie will n
```

See


- [OWASP Top 10 2021 Category A4](#) - Insecure Design
- [OWASP Top 10 2021 Category A5](#) - Security Misconfiguration
- [OWASP Top 10 2017 Category A3](#) - Sensitive Data Exposure
- [MITRE, CWE-311](#) - Missing Encryption of Sensitive Data
- [MITRE, CWE-315](#) - Cleartext Storage of Sensitive Information in a Cookie
- [MITRE, CWE-614](#) - Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
- [SANS Top 25](#) - Porous Defenses

Available In:


https://rules.sonarsource.com/java/RSPEC-2092

1/2


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

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](#)