

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...

"equals" method parameters should not be marked "@Nonnull"

Code Smell

A conditionally executed single line should be denoted by indentation

Code Smell

Conditionals should start on new lines

Code Smell

Cognitive Complexity of methods should not be too high

Code Smell

Factory method injection should be used in "@Configuration" classes

Code Smell

"static" base class members should not be accessed via derived types

Code Smell

Instance methods should not write to "static" fields

Code Smell

"indexOf" checks should not be for positive numbers

Code Smell

Method overrides should not change contracts

Code Smell

Whitespace and control characters in literals should be explicit

Code Smell

Null should not be returned from a "Boolean" method

Code Smell

Classes should not access their own subclasses during initialization

Code Smell

"Object.wait(...)" and

"HttpServletRequest.getRequestSessionId()" should not be used

Analyze your code

Vulnerability

Critical

cwe sans-top25 owasp

According to the Oracle Java API, the `HttpServletRequest.getRequestSessionId()` method:

Returns the session ID specified by the client. This may not be the same as the ID of the current valid session for this request. If the client did not specify a session ID, this method returns null.

The session ID it returns is either transmitted in a cookie or a URL parameter so by definition, nothing prevents the end-user from manually updating the value of this session ID in the HTTP request.

Here is an example of a updated HTTP header:

```
GET /pageSomeWhere HTTP/1.1
Host: webSite.com
User-Agent: Mozilla/5.0
Cookie: JSESSIONID=Hacked_Session_Value''''>
```

Due to the ability of the end-user to manually change the value, the session ID in the request should only be used by a servlet container (E.G. Tomcat or Jetty) to see if the value matches the ID of an an existing session. If it does not, the user should be considered unauthenticated. Moreover, this session ID should never be logged as is but using a one-way hash to prevent hijacking of active sessions.

Noncompliant Code Example

```
if(isActiveSession(request.getRequestSessionId())) ){
    ...
}
```

See

- OWASP Top 10 2021 Category A4 - Insecure Design
- OWASP Top 10 2017 Category A2 - Broken Authentication
- MITRE, CWE-807 - Reliance on Untrusted Inputs in a Security Decision
- SANS Top 25 - Porous Defenses

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-2254

1/2

<div><div>"Condition.await(...)" should be called inside a "while" loop</div><div> Code Smell</div></div>
<div><div>IllegalMonitorStateException should not be caught</div><div> Code Smell</div></div>
<div><div>JUnit assertions should not be used in "run" methods</div><div> Code Smell</div></div>
<div><div>Placeholder for the next rule</div></div>