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...

Bug

Zero should not be a possible denominator

Bug

Locks should be released

Bug

"runFinalizersOnExit" should not be called

Bug

"ScheduledThreadPoolExecutor" should not have 0 core threads

Bug

"Random" objects should be reused

Bug

The signature of "finalize()" should match that of "Object.finalize()"

Bug

Jump statements should not occur in "finally" blocks

Bug

"super.finalize()" should be called at the end of "Object.finalize()" implementations

Bug

Using slow regular expressions is security-sensitive

Security Hotspot

Using publicly writable directories is security-sensitive

Security Hotspot

Using clear-text protocols is security-sensitive

Security Hotspot

Accessing Android external storage is

A new session should be created during user authentication

Analyze your code

Vulnerability | Critical | cwe spring owasp

Session fixation attacks occur when an attacker can force a legitimate user to use a session ID that he knows. To avoid fixation attacks, it's a good practice to generate a new session each time a user authenticates and delete/invalidate the existing session (the one possibly known by the attacker).

Noncompliant Code Example

In a Spring Security's context, session fixation protection is enabled by default but can be disabled with `sessionFixation().none()` method:

```
@Override
protected void configure(HttpSecurity http) throws Exception
   http.sessionManagement()
     .sessionFixation().none(); // Noncompliant: the existin
}
```

Compliant Solution

In a Spring Security's context, session fixation protection can be enabled as follows:

```
@Override
protected void configure(HttpSecurity http) throws Exception
  http.sessionManagement()
     .sessionFixation().newSession(); // Compliant: a new se

  // or

  http.sessionManagement()
     .sessionFixation().migrateSession(); // Compliant: a ne
}
```

See

- OWASP Top 10 2021 Category A7 - Identification and Authentication Failures
- OWASP Top 10 2017 Category A2 - Broken Authentication
- OWASP Sesssion Fixation
- MITRE, CWE-384 - Session Fixation

Available In:

sonarlint sonarcloud sonarqube

Accessing Android external storage is security-sensitive

Security Hotspot

Receiving intents is security-sensitive

Security Hotspot

Broadcasting intents is security-sensitive

Security Hotspot

Expanding archive files without controlling resource consumption is security-sensitive

Security Hotspot