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...

Vulnerability

Reflection should not be vulnerable to injection attacks
Vulnerability

Authorizations should be based on strong decisions
Vulnerability

OpenSAML2 should be configured to prevent authentication bypass
Vulnerability

Server-side requests should not be vulnerable to forging attacks
Vulnerability

Collections should not be modified while they are iterated
Bug

Equals method should be overridden in records containing array fields
Bug

Reflection should not be used to increase accessibility of records' fields
Bug

AssertJ assertions with "Consumer" arguments should contain assertion inside consumers
Bug

The regex escape sequence \cX should only be used with characters in the @-_ range
Bug

Regular expressions should not overflow the stack
Bug

Tests method should not be annotated with competing

"runFinalizersOnExit" should not be called

Analyze your code

Bug Critical cert

Running finalizers on JVM exit is disabled by default. It can be enabled with `System.runFinalizersOnExit` and `Runtime.runFinalizersOnExit`, but both methods are deprecated because they are inherently unsafe.

According to the Oracle Javadoc:

> It may result in finalizers being called on live objects while other threads are concurrently manipulating those objects, resulting in erratic behavior or deadlock.

If you really want to execute something when the virtual machine begins its shutdown sequence, you should attach a shutdown hook.

Noncompliant Code Example

```
public static void main(String [] args) {
  ...
  System.runFinalizersOnExit(true);  // Noncompliant
  ...
}

protected void finalize(){
  doSomething();
}
```

Compliant Solution

```
public static void main(String [] args) {
  Runtime.addShutdownHook(new Runnable() {
    public void run(){
      doSomething();
    }
  });
  //...
```

See

- CERT, MET12-J. - Do not use finalizers

Available In:
sonarlint sonarcloud sonarqube

annotations

Bug

Assertions should not be used in production code

Bug

DateTimeFormatters should not use mismatched year and week numbers

Bug

Unicode Grapheme Clusters should be avoided inside regex character classes

Bug