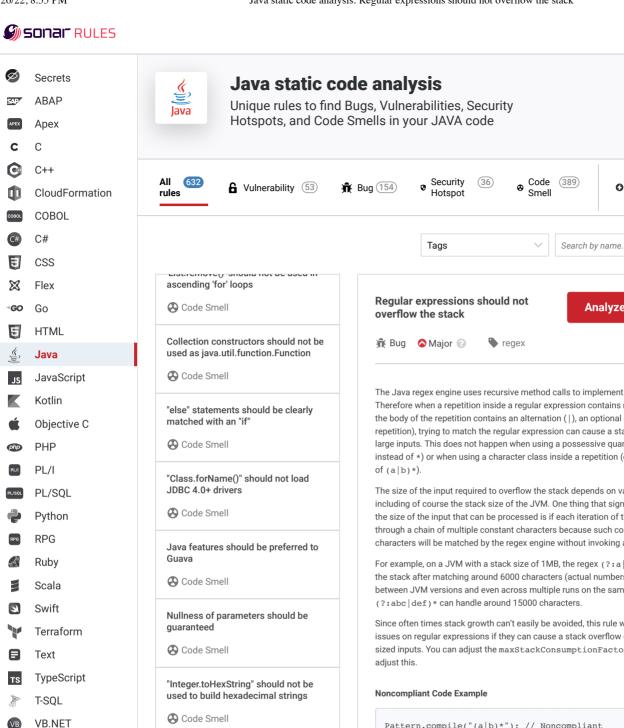
Products ✓

O Quick 42



Asserts should not be used to check the parameters of a public method

Assignments should not be redundant

Methods should not have identical

"java.nio.Files#delete" should be

Unused "private" classes should be

A Code Smell

Code Smell

implementations

Code Smell

Code Smell

preferred

removed Code Smell

Analyze your code The Java regex engine uses recursive method calls to implement backtracking. Therefore when a repetition inside a regular expression contains multiple paths (i.e. the body of the repetition contains an alternation (|), an optional element or another repetition), trying to match the regular expression can cause a stack overflow on large inputs. This does not happen when using a possessive quantifier (such as *+ instead of *) or when using a character class inside a repetition (e.g. [ab]* instead The size of the input required to overflow the stack depends on various factors, including of course the stack size of the JVM. One thing that significantly increases the size of the input that can be processed is if each iteration of the repetition goes through a chain of multiple constant characters because such consecutive characters will be matched by the regex engine without invoking any recursion. For example, on a JVM with a stack size of 1MB, the regex (?:a|b)* will overflow the stack after matching around 6000 characters (actual numbers may differ between JVM versions and even across multiple runs on the same JVM) whereas Since often times stack growth can't easily be avoided, this rule will only report issues on regular expressions if they can cause a stack overflow on realistically sized inputs. You can adjust the maxStackConsumptionFactor parameter to Pattern.compile("(a|b)*"); // Noncompliant Pattern.compile("(.|\n)*"); // Noncompliant Pattern.compile("(ab?)*"); // Noncompliant **Compliant Solution** Pattern.compile("[ab]*"); // Character classes don't cause r Pattern.compile("(?s).*"); // Enabling the (?s) flag makes Pattern.compile("(ab?)*+"); // Possessive quantifiers don't Available In: sonarlint ⊕ | sonarcloud ↔ | sonarcube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONAROUBE 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

VB₆

XML

"Stream.peek" should be used with caution Code Smell "Map.get" and value test should be replaced with single method call Code Smell
"Map.get" and value test should be replaced with single method call
replaced with single method call
Code Smell
"@RequestMapping" methods should not be "private"
Code Smell
Raw types should not be used
☼ Code Smell