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

JavaScript static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVASCRIPT code

All rules 285 | Vulnerability 29 | Bug 62 | Security Hotspot 43 | Code Smell 151 | Quick Fix 41

Tags | Search by name...

Code Smell

Extra semicolons should be removed
Code Smell

Class names should comply with a naming convention
Code Smell

Track uses of "TODO" tags
Code Smell

Web SQL databases should not be used
Vulnerability

Variables should be defined before being used
Bug

Variables declared with "var" should be declared before they are used
Code Smell

Track lack of copyright and license headers
Code Smell

Reading the Standard Input is security-sensitive
Security Hotspot

Using command line arguments is security-sensitive
Security Hotspot

Using Sockets is security-sensitive
Security Hotspot

Executing XPath expressions is security-sensitive
Security Hotspot

Encrypting data is security-sensitive
Security Hotspot

Regular expressions should not contain empty groups

Analyze your code

Code Smell Major regex

There are several reasons to use a group in a regular expression:

- to change the precedence (e.g. `do(g|or)` will match 'dog' and 'door')
- to remember parenthesised part of the match in the case of capturing group
- to improve readability

In any case, having an empty group is most probably a mistake. Either it is a leftover after refactoring and should be removed, or the actual parentheses were intended and were not escaped.

Noncompliant Code Example

```
const dateRegex = /^(?:0[1-9]|[12][0-9]|3[01])[- /.](?:0[1-9
const methodCallRegex = /foo()/;  // Noncompliant, will matc
```

Compliant Solution

```
const dateRegex = /^(?:0[1-9]|[12][0-9]|3[01])[- /.](?:0[1-9
const methodCallRegex = /foo\(\)/; // OK, matches 'foo()'
```

Available In:

sonarlint sonarcloud sonarqube

Using regular expressions is security-sensitive

Security Hotspot

Class methods should be used instead of "prototype" assignments

Code Smell

Function constructors should not be used

Code Smell

Variables should be declared with "let" or "const"

Code Smell