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

Functions should not have too many lines of code
Code Smell

Track comments matching a regular expression
Code Smell

Statements should be on separate lines
Code Smell

Magic numbers should not be used
Code Smell

Collapsible "if" statements should be merged
Code Smell

Standard outputs should not be used directly to log anything
Code Smell

Files should not have too many lines of code
Code Smell

Lines should not be too long
Code Smell

Debugger statements should not be used
Vulnerability

"alert(...)" should not be used
Vulnerability

Regular expressions using Unicode character classes or property escapes should enable the unicode flag
Bug

The base should be provided to

"arguments.caller" and "arguments.callee" should not be used

Analyze your code

Code Smell Major obsolete

Both `arguments.caller` and `arguments.callee` make quite a few optimizations impossible so they were deprecated in latest versions of JavaScript. In fact, EcmaScript 5 forbids the use of both in `strict` mode, according to the docs:

> Arguments objects for strict mode functions define non-configurable accessor properties named "caller" and "callee" which throw a TypeError exception on access.

The same restriction applies to the function's `caller` and `arguments` properties in `strict` mode.

Noncompliant Code Example

```
function whoCalled() {
   if (arguments.caller == null)   //Noncompliant
      console.log('I was called from the global scope.');
   else
      console.log(arguments.caller + ' called me!');  // Non

  console.log(whoCalled.caller);  // Noncompliant
  console.log(whoCalled.arguments);  // Noncompliant
}
```

Available In:
sonarlint sonarcloud sonarqube

"parseInt"

Bug

Function declarations should not be made within blocks

Bug

Writing cookies is security-sensitive

Security Hotspot

"continue" should not be used

Code Smell

Trailing commas should be used

Code Smell