




 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... 🔍

Getters and setters should access the expected fields



"super()" should be invoked appropriately



"Symbol" should not be used as a constructor



Results of "in" and "instanceof" should be negated rather than operands



"in" should not be used with primitive types



A compare function should be provided when using "Array.prototype.sort()"



Jump statements should not occur in "finally" blocks



Using slow regular expressions is security-sensitive



Using publicly writable directories is security-sensitive



Using clear-text protocols is security-sensitive



Expanding archive files without controlling resource consumption is security-sensitive



"future reserved words" should not be used as identifiers

Analyze your code

 Code Smell  Blocker  lock-in pitfall

The following words may be used as keywords in future evolutions of the language, so using them as identifiers should be avoided to allow an easier adoption of those potential future versions:

- await
- class
- const
- enum
- export
- extends
- implements
- import
- interface
- let
- package
- private
- protected
- public
- static
- super
- yield

Use of these words as identifiers would produce an error in JavaScript strict mode code.

Noncompliant Code Example





```
var package = document.getElementsByName("foo"); // Noncompliant
var someData = { package: true };                // Compliant
```

Compliant Solution

```
var elements = document.getElementsByName("foo"); // Compliant
```

Available In:

sonarlint  sonarcloud  sonarqube 

<div>Using weak hashing algorithms is security-sensitive</div> <div> Security Hotspot</div>
<div>Disabling CSRF protections is security-sensitive</div> <div> Security Hotspot</div>
<div>Using pseudorandom number generators (PRNGs) is security-sensitive</div> <div> Security Hotspot</div>
<div>Dynamically executing code is security-sensitive</div> <div> Security Hotspot</div>