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

TS

TypeScript static code analysis

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

All rules 279 | Vulnerability 27 | Bug 51 | Security Hotspot 43 | Code Smell 158 | Quick Fix 50

Tags ⌄ | Search by name...

Delivering code in production with debug features activated is security-sensitive
Security Hotspot

Creating cookies without the "HttpOnly" flag is security-sensitive
Security Hotspot

Creating cookies without the "secure" flag is security-sensitive
Security Hotspot

Using hardcoded IP addresses is security-sensitive
Security Hotspot

Regular expression quantifiers and character classes should be used concisely
Code Smell

Regular expression literals should be used when possible
Code Smell

"await" should not be used redundantly
Code Smell

Redundant casts and non-null assertions should be avoided
Code Smell

Type aliases should be used
Code Smell

Type guards should be used
Code Smell

"module" should not be used
Code Smell

"for of" should be used with Iterables
Code Smell

Variables should not be self-assigned

Analyze your code

Bug Major

There is no reason to re-assign a variable to itself. Either this statement is redundant and should be removed, or the re-assignment is a mistake and some other value or variable was intended for the assignment instead.

Noncompliant Code Example

```
function setName(name) {
    name = name;
}
```

Compliant Solution

```
function setName(name) {
    this.name = name;
}
```

Available In:

sonarlint sonarcloud sonarqube

Imports from the same modules should be merged

Code Smell

Jump statements should not be redundant

Code Smell

Default export names and file names should match

Code Smell

The global "this" object should not be used

Code Smell