




 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



TypeScript static code analysis

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

All rules279

Vulnerability27

Bug51

Security Hotspot43


Code Smell158

Quick Fix50


Tags ▾

Search by name... 🔍


expected fields

 Bug


"super()" should be invoked appropriately

 Bug


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

 Bug


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

 Bug


Jump statements should not occur in "finally" blocks

 Bug


Using slow regular expressions is security-sensitive

 Security Hotspot


Using publicly writable directories is security-sensitive

 Security Hotspot


Using clear-text protocols is security-sensitive

 Security Hotspot


Expanding archive files without controlling resource consumption is security-sensitive

 Security Hotspot


Using weak hashing algorithms is security-sensitive

 Security Hotspot

Disabling CSRF protections is security-sensitive


 Security Hotspot


Using pseudorandom number generators (PRNGs) is security-sensitive


 Security Hotspot

Octal values should not be used

Analyze your code

 Code Smell

 Blocker

 pitfall

Integer literals starting with a zero are octal rather than decimal values. While using octal values is fully supported, most developers do not have experience with them. They may not recognize octal values as such, mistaking them instead for decimal values.




Noncompliant Code Example

```
var myNumber = 010; // Noncompliant. myNumber will hold 8,
```

Compliant Solution

```
var myNumber = 8;
```





Available In:

 |  | 

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE 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](#)

https://rules.sonarsource.com/typescript/RSPEC-1314

1/2

sensitive  Security Hotspot
Dynamically executing code is security-sensitive  Security Hotspot
Equality operators should not be used in "for" loop termination conditions  Code Smell
Tests should not execute any code after "done()" is called  Code Smell
Union and intersection types should