




 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... 🔍

 Code Smell


Primitive types should be omitted from initialized or defaulted declarations




Non-null assertions should not be used




"undefined" should not be assigned




Trailing commas should not be used




Array constructors should not be used




Quotes for string literals should be used consistently




Statements should end with semicolons




Comments should not be located at the end of lines of code




Loops should not contain more than a single "break" or "continue" statement



Variable, property and parameter names should comply with a naming convention





Lines should not end with trailing whitespaces




Disabling resource integrity features is security-sensitive

Analyze your code

 Security Hotspot

 Minor ?

 cwe owasp

Fetching external resources, for example from a CDN, without verifying their integrity could impact the security of an application if the CDN gets compromised and resources are replaced by malicious ones. Resources integrity feature will block resources inclusion into an application if the pre-computed digest of the expected resource doesn't match with the digest of the retrieved resource.

Ask Yourself Whether

- The resources are fetched from external CDNs.

There is a risk if you answered yes to this question.

Recommended Secure Coding Practices

- implement resources integrity checks for all static resources (where "static" means that the resource's content doesn't change dynamically based on the browser)
- use versioned resources instead of using "latest" version of the resources

Sensitive Code Example

```
let script = document.createElement("script"); // Sensitive
script.src = "https://cdnexample.com/script-latest.js";
script.crossOrigin = "anonymous";
document.head.appendChild(script);
```



Compliant Solution

```
let script = document.createElement("script");
script.src = "https://cdnexample.com/script-v1.2.3.js";
script.integrity = "sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/";
script.crossOrigin = "anonymous";
document.head.appendChild(script);
```

See

- [OWASP Top 10 2021 Category A8](#) - Software and Data Integrity Failures
- [MITRE, CWE-353](#) - Missing Support for Integrity Check
- [OWASP Top 10 2017 Category A6](#) - Security Misconfiguration
- [developer.mozilla.org](#) - Subresource Integrity





Available In:

https://rules.sonarsource.com/typescript/RSPEC-5725

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR,

1/2

<div>Files should contain an empty newline at the end</div> <div> Code Smell</div>
<div>An open curly brace should be located at the end of a line</div> <div> Code Smell</div>
<div>Tabulation characters should not be used</div> <div> Code Smell</div>
<div>Function and method names should comply with a naming convention</div> <div> Code Smell</div>

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](#)