




 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



### Executing XPath expressions is security-sensitive

 Security Hotspot  Critical ⓘ

Executing XPATH expressions is security-sensitive. It has led in the past to the following vulnerabilities:

- [CVE-2016-6272](#)
- [CVE-2016-9149](#)
- [CVE-2012-4837](#)

User-provided data such as URL parameters should always be considered as untrusted and tainted. Constructing XPath expressions directly from tainted data enables attackers to inject specially crafted values that changes the initial meaning of the expression itself. Successful XPath injections attacks can read sensitive information from the XML document.

#### Ask Yourself Whether

- the XPATH expression might contain some unsafe input coming from a user.

You are at risk if you answered yes to this question.

#### Recommended Secure Coding Practices

Sanitize any user input before using it in an XPATH expression.

#### Sensitive Code Example

```
// === Server side ===

var xpath = require('xpath');
var xmlDom = require('xmldom');

var doc = new xmlDom.DOMParser().parseFromString(xml);
var nodes = xpath.select(userinput, doc); // Sensitive
var node = xpath.select1(userinput, doc); // Sensitive

// === Client side ===

// Chrome, Firefox, Edge, Opera, and Safari use the evaluate
var nodes = document.evaluate(userinput, xmlDoc, null, XPath

// Internet Explorer uses its own methods to select nodes:
var nodes = xmlDoc.selectNodes(userinput); // Sensitive
var node = xmlDoc.SelectSingleNode(userinput); // Sensitive
```







#### See

- [OWASP Top 10 2017 Category A1](#) - Injection
- [MITRE, CWE-643](#) - Improper Neutralization of Data within XPath Expressions

#### Deprecated

https://rules.sonarsource.com/typescript/RSPEC-4817

1/2

<div>Files should contain an empty newline at the end</div> <div> Code Smell</div>	<div>This rule is deprecated, and will eventually be removed.</div> <div>Available In:  </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>	

© 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](#)