




 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


 Code Smell


"await" should only be used with promises





 A conditionally executed single line should be denoted by indentation





 Conditionals should start on new lines





 Cognitive Complexity of functions should not be too high





 "void" should not be used





 Loop counters should not be assigned to from within the loop body





 "for" loop increment clauses should modify the loops' counters





 Functions should not be empty





 Server-side requests should not be vulnerable to forging attacks



 Non-empty statements should change control flow or have at least one side-effect





 Regular expressions with the global flag should be used with caution





Origins should be verified during cross-origin communications

Analyze your code

 Vulnerability

 Critical



 cwe

html5

owasp

Browsers **allow message exchanges** between Window objects of different origins.

Because any window can send or receive messages from another window, it is important to verify the sender's/receiver's identity:

- When sending a message with the postMessage method, the identity's receiver should be defined (the wildcard keyword (\*) should not be used).
- When receiving a message with a message event, the sender's identity should be verified using the origin and possibly source properties.

**Noncompliant Code Example**

When sending a message:

```
var iframe = document.getElementById("testiframe");
iframe.contentWindow.postMessage("secret", "*"); // Noncompliant
```

When receiving a message:

```
window.addEventListener("message", function(event) { // Noncompliant
    console.log(event.data);
});
```

**Compliant Solution**

When sending a message:

```
var iframe = document.getElementById("testsecureiframe");
iframe.contentWindow.postMessage("hello", "https://secure.example.com");
```

When receiving a message:

```
window.addEventListener("message", function(event) {
    if (event.origin !== "http://example.org") // Compliant
        return;

    console.log(event.data);
});
```

**See**

- [OWASP Top 10 2021 Category A1](#) - Broken Access Control
- [OWASP Top 10 2017 Category A3](#) - Broken Authentication and Session Management
- [developer.mozilla.org](#) - postMessage API
- [MITRE, CWE-345](#) - Insufficient Verification of Data Authenticity

https://rules.sonarsource.com/typescript/RSPEC-2819

1/2

replacement strings should reference existing regular expression groups



Regular expressions should not contain control characters



Alternation in regular expressions should not contain empty alternatives



Mocha timeout should be disabled by setting it to "0".



Available In:

sonarlint  | sonarcloud  | sonarqube 

© 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](#)