

































-  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


Assignments should not be made from within sub-expressions





 Labels should not be used





 Variables should not be shadowed





 Redundant pairs of parentheses should be removed





 Nested blocks of code should not be left empty





 Functions should not have too many parameters





 OS commands should not be vulnerable to argument injection attacks





 Repeated patterns in regular expressions should not match the empty string





 Empty collections should not be accessed or iterated





 "delete" should be used only with object properties



 Function parameters, caught exceptions and foreach variables' initial values should not be ignored



### Comma and logical OR operators should not be used in switch cases

 Bug  Major ?

The comma operator (,) evaluates its operands, from left to right, and returns the second one. That's useful in some situations, but just wrong in a switch case. You may think you're compactly handling multiple values in the case, but only the last one in the comma-list will ever be handled. The rest will fall through to the default.

Similarly the logical OR operator (||) will not work in a switch case, only the first argument will be considered at execution time.




#### Noncompliant Code Example

```
switch a {
  case 1,2: // Noncompliant; only 2 is ever handled by this
    doTheThing(a);
  case 3 || 4: // Noncompliant; only '3' is handled
    doThatThing(a);
  case 5:
    doTheOtherThing(a);
  default:
    console.log("Neener, neener!"); // this happens when a=
}
```

#### Compliant Solution

```
switch a {
  case 1:
  case 2:
    doTheThing(a);
  case 3:
  case 4:
    doThatThing(a);
  case 5:
    doTheOtherThing(a);
  default:
    console.log("Neener, neener!");
}
```






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-3616

1/2

 Bug
<div>Forwarding client IP address is security-sensitive</div> <div> Security Hotspot</div>
<div>Allowing confidential information to be logged is security-sensitive</div> <div> Security Hotspot</div>
<div>Allowing browsers to perform DNS prefetching is security-sensitive</div> <div> Security Hotspot</div>
<div>Disabling Certificate Transparency monitoring is security-sensitive</div> <div></div>