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

PHP static code analysis

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

All rules 268 | Vulnerability 40 | Bug 51 | Security Hotspot 33 | Code Smell 144

Tags

Search by name...

Bug

Alternation in regular expressions should not contain empty alternatives
Bug

Unicode Grapheme Clusters should be avoided inside regex character classes
Bug

Assertions should not compare an object to itself
Bug

Regex alternatives should not be redundant
Bug

Alternatives in regular expressions should be grouped when used with anchors
Bug

Array values should not be replaced unconditionally
Bug

Exceptions should not be created without being thrown
Bug

Array or Countable object count comparisons should make sense
Bug

All branches in a conditional structure should not have exactly the same implementation
Bug

The output of functions that don't return anything should not be used
Bug

Regex patterns following a possessive quantifier should not always fail

Analyze your code

Bug | Critical | regex

Possessive quantifiers in Regex patterns like below improve performance by eliminating needless backtracking:

```
?+ , *+ , ++ , {n}+ , {n,}+ , {n,m}+
```

But because possessive quantifiers do not keep backtracking positions and never give back, the following sub-patterns should not match only similar characters. Otherwise, possessive quantifiers consume all characters that could have matched the following sub-patterns and nothing remains for the following sub-patterns.

Noncompliant Code Example

```
"/a++abc/"              // Noncompliant, the second 'a'
"/\d*+[02468]/"         // Noncompliant, the sub-patter
```

Compliant Solution

```
"/aa++bc/"              // Compliant, for example it ca
"/\d*+(?<=[02468])/"    // Compliant, for example it ca
```

Available In:

sonarlint | sonarcloud | sonarqube

Unary prefix operators should not be repeated

Bug

"=+" should not be used instead of "+="

Bug

A "for" loop update clause should move the counter in the right direction

Bug

Return values from functions without side effects should not be ignored

Bug

Values should not be uselessly