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...

PHP 4 constructor declarations should not be used

⊗ Code Smell

Deprecated predefined variables should not be used

⊗ Code Smell

"switch" statements should not have too many "case" clauses

⊗ Code Smell

Classes should not have too many methods

⊗ Code Smell

Functions should not have too many lines of code

⊗ Code Smell

"for" loop stop conditions should be invariant

⊗ Code Smell

Sections of code should not be commented out

⊗ Code Smell

Unused function parameters should be removed

⊗ Code Smell

Unused "private" methods should be removed

⊗ Code Smell

Functions should not contain too many return statements

⊗ Code Smell

Track uses of "FIXME" tags

⊗ Code Smell

Generic exceptions ErrorException, RuntimeException and Exception should not be thrown

---

Array values should not be replaced unconditionally

Analyze your code

🐛 Bug   🔺 Major ⦸   🏷 suspicious

It is highly suspicious when a value is saved for a key or index and then unconditionally overwritten. Such replacements are likely errors.

Noncompliant Code Example

```
$someArray[1] = "someValue";
$someArray[1] = "someOtherValue"; // The intention here was
```

Available In:

sonarlint ⊖ | **sonar**cloud ☁ | **sonar**qube

⊗ Code Smell

Local variables should not have the same name as class fields

⊗ Code Smell

Redundant pairs of parentheses should be removed

⊗ Code Smell

Inheritance tree of classes should not be too deep

⊗ Code Smell

Nested blocks of code should not be left empty