

- 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... 🔍

corresponding dedicated assertion

Code Smell

Methods should not have identical implementations

Code Smell

Functions should use "return" consistently

Code Smell

Assertion arguments should be passed in the correct order

Code Smell

Ternary operators should not be nested

Code Smell

Reflection should not be used to increase accessibility of classes, methods, or fields

Code Smell

Multiline blocks should be enclosed in curly braces

Code Smell

Parameters should be passed in the correct order

Code Smell

Classes named like "Exception" should extend "Exception" or a subclass

Code Smell

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

Code Smell

Unused assignments should be removed

Code Smell

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

Analyze your code

Bug Major cwe unused

Any statement (other than a null statement, which means a statement containing only a semicolon ;) which has no side effect and does not result in a change of control flow will normally indicate a programming error, and therefore should be refactored.

Noncompliant Code Example

```
$a == 1; // Noncompliant; was assignment intended?
$a < $b; // Noncompliant; have we forgotten to assign t
```

See


- MITRE, CWE-482 - Comparing instead of Assigning

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](#)


Method arguments with default values should be last

 Code Smell


A reason should be provided when skipping a test

 Code Smell

"__construct" functions should not make PHP 4-style calls to parent constructors

 Code Smell

PHP 4 constructor declarations should not be used

 Code Smell

PHPUnit 4-style assertions should be used