

- 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...



of code

Code Smell

Lines should not be too long

Code Smell

HTTP response headers should not be vulnerable to injection attacks

Vulnerability

"sleep" should not be called

Vulnerability

Static members should be referenced with "static::"

Bug

"require_once" and "include_once" should be used instead of "require" and "include"

Bug

Errors should not be silenced

Bug

Files should not contain characters before "<?php"

Bug

Method visibility should be explicitly declared

Bug

Controlling permissions is security-sensitive

Security Hotspot

Writing cookies is security-sensitive

Security Hotspot

Framework-provided functions should be used to test exceptions

Code Smell

Method arguments with default values should be last

Analyze your code

Code Smell Major ? psr2

The ability to define default values for method arguments can make a method easier to use. Default argument values allow callers to specify as many or as few arguments as they want while getting the same functionality and minimizing boilerplate, wrapper code.

But all method arguments with default values should be declared after the method arguments without default values. Otherwise, it makes it impossible for callers to take advantage of defaults; they must re-specify the defaulted values in order to "get to" the non-default arguments.

Noncompliant Code Example

```
function makeyogurt($type = "acidophilus", $flavor){...
    makeyogurt("raspberry")}} // Runtime error: Missing ar
```

Compliant Solution

```
function makeyogurt($flavor, $type = "acidophilus", ){...
    makeyogurt("raspberry")}} // Works as expected
```

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](#)

Unicode-aware versions of character classes should be preferred

⚠ Code Smell

Alias functions should not be used

⚠ Code Smell

Perl-style comments should not be used

⚠ Code Smell

Superglobals should not be accessed directly

⚠ Code Smell

Colors should be defined in upper