

- 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... 🔍

Security Hotspot
Setting loose POSIX file permissions is security-sensitive
Security Hotspot
Formatting SQL queries is security-sensitive
Security Hotspot
"goto" statement should not be used
Code Smell
Character classes in regular expressions should not contain only one character
Code Smell
Superfluous curly brace quantifiers should be avoided
Code Smell
Non-capturing groups without quantifier should not be used
Code Smell
WordPress option names should not be misspelled
Code Smell
WordPress options should not be defined at the end of "wp-config.php"
Code Smell
Constants should not be redefined
Code Smell
Regular expressions should not contain empty groups
Code Smell
Regular expressions should not contain multiple spaces

References should not be passed to function calls

Analyze your code

Code Smell

Critical

cwe performance obsolete pitfall

Passing a reference to a function parameter means that any modifications the method makes to the parameter will be made to the original value as well, since references have the effect of pointing two variables at the same memory space. This feature can be difficult to use correctly, particularly if the callee is not expecting a reference, and the improper use of references in function calls can make code less efficient rather than more efficient.

Further, according to the PHP manual:

As of PHP 5.3.0, you will get a warning saying that "call-time pass-by-reference" is deprecated... And as of PHP 5.4.0, call-time pass-by-reference was removed, so using it will raise a fatal error.

Noncompliant Code Example

```
myfun(&$name); // Noncompliant
```

Compliant Solution


```
myfun($name);
```

See


- MITRE, CWE-374 - Weakness Base Passing Mutable Objects to an Untrusted Method

Available In:


sonarlint | sonarcloud | sonarqube

 Code Smell


Single-character alternations in regular expressions should be replaced with character classes

 Code Smell

Reluctant quantifiers in regular expressions should be followed by an expression that can't match the empty string

 Code Smell

Character classes in regular expressions should not contain the same character twice

 Code Smell

Regular expressions should not be too complicated