

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- 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... 🔍

Encrypting data is security-sensitive
Security Hotspot
Using regular expressions is security-sensitive
Security Hotspot
Deserializing objects from an untrusted source is security-sensitive
Security Hotspot
Literal boolean values and nulls should not be used in equality assertions
Code Smell
"global" should not be used
Code Smell
"switch" statements should not be nested
Code Smell
Cyclomatic Complexity of functions should not be too high
Code Smell
Control flow statements "if", "for", "while", "switch" and "try" should not be nested too deeply
Code Smell
Cyclomatic Complexity of classes should not be too high
Code Smell
"if ... else if" constructs should end with "else" clauses
Code Smell
Expressions should not be too complex
Code Smell
"cgi.force_redirect" should be enabled

Regular expressions should not contain empty groups

Analyze your code

Code Smell

Major ?

regex

- There are several reasons to use a group in a regular expression:
- to change the precedence (e.g. `/do(g|or)/` will match 'dog' and 'door')
 - to remember parenthesised part of the match in the case of capturing group
 - to improve readability

In any case, having an empty group is most probably a mistake. Either it is a leftover after refactoring and should be removed, or the actual parentheses were intended and were not escaped.

Noncompliant Code Example

```
"/foo()/" // Noncompliant, will match only 'foo'
```


Compliant Solution

```
"/foo\(\)/" // Matches 'foo()'
```

Available In:

sonarlint | sonarcloud | sonarqube


`ignore_redirect` should be enabled

 Vulnerability


Files that define symbols should not cause side-effects

 Bug

Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression

 Code Smell

Test methods should be discoverable

 Code Smell

Use of namespaces should be preferred to "include" or "require" functions