

- 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...

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
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
Code Smell
Duplicate values should not be passed as arguments
Code Smell
Configuration should not be changed dynamically

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

Analyze your code

Code Smell

Major

regex

When a reluctant (or lazy) quantifier is followed by a pattern that can match the empty string or directly by the end of the regex, it will always match zero times for *? or one time for +?. If a reluctant quantifier is followed directly by the end anchor (\$), it behaves indistinguishably from a greedy quantifier while being less efficient.

This is likely a sign that the regex does not work as intended.

Noncompliant Code Example

```
preg_replace("/start\\w*(end)?/", "x", "start123endstart456")
preg_match("/^d*$/", "123456789"); // Noncompliant. Matche
```

Compliant Solution





```
preg_replace("/start\\w*(end|$)/", "x", "start123endstart456")
preg_match("/^d*$$/", "123456789");
```

Available In:

sonarlint

sonarcloud

sonarqube

symfony
 Code Smell
Class constructors should not create other objects  Code Smell
PHP parser failure  Code Smell
The names of methods with boolean return values should start with "is" or "has"  Code Smell
"php_sani_name()" should not be used