sonar RULES

Products

- 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...

Code Smell

Generic exceptions ErrorException, RuntimeException and Exception should not be thrown

Code Smell

Local variables should not have the same name as class fields

Code Smell

Redundant pairs of parentheses should be removed

Code Smell

Inheritance tree of classes should not be too deep

Code Smell

Nested blocks of code should not be left empty

Code Smell

Functions should not have too many parameters

Code Smell

Unused "private" fields should be removed

Code Smell

Collapsible "if" statements should be merged

Code Smell

OS commands should not be vulnerable to argument injection attacks

Vulnerability

Logging should not be vulnerable to injection attacks

Vulnerability

Repeated patterns in regular expressions should not match the

"=+" should not be used instead of "+="

Analyze your code

Bug Major

The use of operators pairs (=+, =- or =!) where the reversed, single operator was meant (+=, -= or !=) will compile and run, but not produce the expected results.

This rule raises an issue when =+, =-, or =! is used without any spacing between the two operators and when there is at least one whitespace character after.

Noncompliant Code Example

```
$target = -5;
$num = 3;

$target =- $num;  // Noncompliant; target = -3. Is that
$target =+ $num; // Noncompliant; target = 3
```

Compliant Solution

```
$target = -5;
$num = 3;

$target = -$num;  // Compliant; intent to assign invers
$target += $num;
```

Available In:

sonarlint | sonarcloud | sonarqube

empty string

Bug

Function and method parameters' initial values should not be ignored

Bug

Having a permissive Cross-Origin Resource Sharing policy is security-sensitive

Security Hotspot

Delivering code in production with debug features activated is security-sensitive

Security Hotspot

Creating cookies without the