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

Track uses of "FIXME" tags

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

The output of functions that don't return anything should not be used

Analyze your code

Bug Major

If a function does not return anything, it makes no sense to use its output. Specifically, passing it to another function, or assigning its "result" to a variable is probably a bug because such functions return nothing, which is probably not what was intended.

Noncompliant Code Example

```
$result = closedir($dir_handle); // Noncompliant, "clos
```

Compliant Solution

```
closedir($dir_handle);
```

Available In:

sonarlint sonarcloud sonarqube

injection attacks

Vulnerability

Repeated patterns in regular expressions should not match the 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