

- 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-sensitive
Security Hotspot
Using Sockets is security-sensitive
Security Hotspot
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

Constants should not be redefined

Analyze your code

Code Smell Major


PHP doesn't fail if two `define` statements assign a value to the same constant name: in such a case, PHP only issues a warning and ignores the second `define`. However, that code should be fixed. In the best case, the second `define` tries to assign the same value to the constant: it's just useless code which can be removed. In the worst case, the second `define` attempts to assign a different value and is followed by code which wrongly assumes the value of the constant: it can be a nasty bug.

Noncompliant Code Example


```
define( 'WP_DEBUG', 1 );
define( 'SCRIPT_DEBUG', 1 );
define( 'WP_DEBUG', 0 ); // Noncompliant, redefines con
echo WP_DEBUG; // output "1"
```

Available In:


sonarlint | sonarcloud | sonarqube

 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