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...

References used in "foreach" loops should be "unset"
Bug

Using clear-text protocols is security-sensitive
Security Hotspot

Expanding archive files without controlling resource consumption is security-sensitive
Security Hotspot

Signalling processes is security-sensitive
Security Hotspot

Configuring loggers is security-sensitive
Security Hotspot

Using weak hashing algorithms is security-sensitive
Security Hotspot

Disabling CSRF protections is security-sensitive
Security Hotspot

Using pseudorandom number generators (PRNGs) is security-sensitive
Security Hotspot

Dynamically executing code is security-sensitive
Security Hotspot

`str_replace` should be preferred to `preg_replace`
Code Smell

"default" clauses should be first or last
Code Smell

Variable variables should not be used

Analyze your code

Code Smell   Blocker   brain-overload

PHP's "variable variables" feature (dynamically-named variables) is temptingly powerful, but can lead to unmaintainable code.

Noncompliant Code Example

```
$var = 'foo';
$$var = 'bar';      //Noncompliant
$$$var = 'hello';  //Noncompliant

echo $foo; //will display 'bar'
echo $bar; //will display 'hello'
```

Available In:

sonarlint   sonarcloud   sonarqube

A conditionally executed single line should be denoted by indentation

Code Smell

Conditionals should start on new lines

Code Smell

Cognitive Complexity of functions should not be too high

Code Smell

Parentheses should not be used for calls to "echo"

Code Smell

Functions should not be nested too deeply