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

"switch case" clauses should not have too many lines of code

Code Smell

Assignments should not be made from within sub-expressions

Code Smell

Files should not have too many lines of code

Code Smell

Lines should not be too long

Code Smell

HTTP response headers should not be vulnerable to injection attacks

Vulnerability

"sleep" should not be called

Vulnerability

Static members should be referenced with "static::"

Bug

"require_once" and "include_once" should be used instead of "require" and "include"

Bug

Errors should not be silenced

Bug

Files should not contain characters before "<?php"

Bug

Method visibility should be explicitly declared

Bug

Unused assignments should be removed

Analyze your code

Code Smell | Major | cwe unused

A dead store happens when a local variable is assigned a value that is not read by any subsequent instruction. Calculating or retrieving a value only to then overwrite it or throw it away, could indicate a serious error in the code. Even if it's not an error, it is at best a waste of resources. Therefore all calculated values should be used.

Noncompliant Code Example

```
$i = $a + $b; // Noncompliant; calculation result not u
$i = compute();
```

Compliant Solution

```
$i = $a + $b;
$i += compute();
```

Exceptions

This rule ignores initializations to -1, 0, 1, `null`, `true`, `false`, `""`, `[]` and `array()`.

See

- MITRE, CWE-563 - Assignment to Variable without Use ('Unused Variable')

Available In:

sonarlint sonarcloud sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
Privacy Policy

Controlling permissions is security-sensitive

Security Hotspot

Writing cookies is security-sensitive

Security Hotspot

Framework-provided functions should be used to test exceptions

Code Smell

Unicode-aware versions of character classes should be preferred

Code Smell

Alias functions should not be used

Code Smell