

- 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...

be removed	Code Smell
Unused "private" methods should be removed	Code Smell
Functions should not contain too many return statements	Code Smell
Track uses of "FIXME" tags	Code Smell
Generic exceptions <code>ErrorException</code> , <code>RuntimeException</code> and <code>Exception</code> 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

All branches in a conditional structure should not have exactly the same implementation

Analyze your code

Bug

Major

Having all branches in a switch or if chain with the same implementation is an error. Either a copy-paste error was made and something different should be executed, or there shouldn't be a switch/if chain at all.

Noncompliant Code Example

```
if ($b == 0) { // Noncompliant
    doOneMoreThing();
} else {
    doOneMoreThing();
}

$b = $a > 12 ? 4 : 4; // Noncompliant

switch ($i) { // Noncompliant
    case 1:
        doSomething();
        break;
    case 2:
        doSomething();
        break;
    case 3:
        doSomething();
        break;
    default:
        doSomething();
}
```

Exceptions

This rule does not apply to if chains without else-s, or to switch-es without default clauses.

```
if($b == 0) { //no issue, this could have been done on p
    doSomething();
} elseif($b == 1) {
    doSomething();
}
```


Available In:

sonarlint

sonarcloud

sonarqube

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 empty string

 Bug

Function and method parameters' initial values should not be ignored