
































-  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... 🔍

Files should not contain inline HTML
 Code Smell
Files should contain only one top-level class or interface each
 Code Smell
Classes should not have too many fields
 Code Smell
Track uses of "NOSONAR" comments
 Code Smell
Statements should be on separate lines
 Code Smell
Classes should not be coupled to too many other classes (Single Responsibility Principle)
 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

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

Analyze your code

 Code Smell

 Major

 design suspicious

Having two cases in a switch statement or two branches in an if chain with the same implementation is at best duplicate code, and at worst a coding error. If the same logic is truly needed for both instances, then in an if chain they should be combined, or for a switch, one should fall through to the other.

### Noncompliant Code Example









```
switch ($i) {
    case 1:
        doFirst();
        doSomething();
        break;
    case 2:
        doSomethingDifferent();
        break;
    case 3: // Noncompliant; duplicates case 1's implementation
        doFirst();
        doSomething();
        break;
    default:
        doTheRest();
}

if ($a >= 0 && $a < 10) {
    doFirst();
    doTheThing();
}
else if ($a >= 10 && $a < 20) {
    doTheOtherThing();
}
else if ($a >= 20 && $a < 50) {
    doFirst();
    doTheThing(); // Noncompliant; duplicates first condition
}
```

### Exceptions

Blocks in an if chain that contain a single line of code are ignored, as are blocks in a switch statement that contain a single line of code with or without a following break.

```
if ($a >= 0 && $a < 10) {
    doTheThing();
}
else if ($a >= 10 && $a < 20) {
```

<div> Vulnerability</div>	
<div>Static members should be referenced with "static::"</div> <div> Bug</div>	
<div>"require_once" and "include_once" should be used instead of "require" and "include"</div> <div> Bug</div>	
<div>Errors should not be silenced</div> <div> Bug</div>	
<div>Files should not contain characters before "&lt;?php"</div> <div> Bug</div>	<div> <pre>doTheOtherThing(); } else if (\$a &gt;= 20 &amp;&amp; \$a &lt; 50) {     doTheThing(); // no issue, usually this is done on p }</pre> </div> <div> <p>But this exception does not apply to if chains without else-s, or to switch-es without default clauses when all branches have the same single line of code. In case of if chains with else-s, or of switch-es with default clauses, rule {rule:php:S3923} raises a bug.</p> </div> <div> <pre>if (\$a &gt;= 0 &amp;&amp; \$a &lt; 10) {     doTheThing(); } else if (\$a &gt;= 20 &amp;&amp; \$a &lt; 50) {     doTheThing(); //Noncompliant; this might have been }</pre> </div> <div> <p>Available In:</p> <p>    </p> </div>