
































-  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... 

 Bug
Assertion failure exceptions should not be ignored
 Bug
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

TestCases should contain tests

Analyze your code tests

 Code Smell

 Blocker 

 tests unused confusing

There's no point in having a PHPUnit test case without any test methods. Similarly, you shouldn't have a file in the tests directory which extends PHPUnit\Framework\TestCase but no tests in the file. Doing either of these things may lead someone to think that uncovered classes have been tested. Add some test method or make the class abstract if it is used by a real test case class.

### Noncompliant Code Example

```
use PHPUnit\Framework\TestCase;

class MyTest extends TestCase {

    protected function setUp() {
        doSomethind();
    }

    private function doSomethind() {
        //...
    }
}
```

### Compliant Solution

```
use PHPUnit\Framework\TestCase;

class MyTest extends TestCase {

    public function testBehaviour() {
        //...
    }

    //...
}


// or

abstract class MyAbstractTest extends TestCase {


    protected function setUp() {
        doSomethind();
    }

    private function doSomethind() {
        //...
    }
}
```


`str_replace` should be preferred to `preg_replace`

 Code Smell


"default" clauses should be first or last

 Code Smell


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

```
}  
}
```

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](#)