

- 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
Back references in regular expressions should only refer to capturing groups that are matched before the reference
Bug
Regex boundaries should not be used in a way that can never be matched
Bug
Regex patterns following a possessive quantifier should not always fail
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

Tests should include assertions

Analyze your code

Code Smell

Blocker ?

tests

phpunit

A test case without assertions ensures only that no exceptions are thrown. Beyond basic runnability, it ensures nothing about the behavior of the code under test.

This rule raised an issue when no assertions are found within a PHPUnit test method.

Noncompliant Code Example

```
public function testDoSomething() { // Compliant
    $myClass = new MyClass();
    $myClass->getSomething();
}
```

Compliant Solution

```
public function testDoSomething() { // Noncompliant
    $myClass = new MyClass();
    $this->assertEquals("foo", $myClass->getSomething());
}
```

Available In:

sonarlint

sonarcloud

sonarqube


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