

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...



Secret keys and salt values should be robust

Vulnerability

Authorizations should be based on strong decisions

Vulnerability

Server-side requests should not be vulnerable to forging attacks

Vulnerability

The number of arguments passed to a function should match the number of parameters

Bug

Non-empty statements should change control flow or have at least one side-effect

Bug

Variables should be initialized before use

Bug

Replacement strings should reference existing regular expression groups

Bug

Alternation in regular expressions should not contain empty alternatives

Bug

Unicode Grapheme Clusters should be avoided inside regex character classes

Bug

Assertions should not compare an object to itself

Bug

Regex alternatives should not be redundant

Bug

## Regular expressions should have valid delimiters

Analyze your code

Bug Critical regex

The pattern provided to PHP regular expression functions is required to be enclosed in valid delimiters. Failing to do so will result in a PHP warning and the pattern never matching. Since the warning only appears during runtime when the pattern is evaluated, such a mistake risks to get unnoticed into production.

A delimiter can be any character that is not alphanumeric, a backslash, or a whitespace. Bracket style delimiters are also allowed.

### Noncompliant Code Example

```
// Condition will always evaluate to false
if (preg_match(".*", $input)) {
    echo "true";
} else {
    echo "false";
}

// Unclosed bracket delimiters
$result = preg_match("[abc", $input);
```

### Compliant Solution

```
if (preg_match("/.*/", $input)) {
    echo "true";
} else {
    echo "false";
}

$result = preg_match("[abc]", $input);
```

### See

- [Delimiters](#) - PHP Documentation

Available In:

sonarlint | sonarcloud | sonarqube

Alternatives in regular expressions should be grouped when used with anchors

🐛 Bug

Array values should not be replaced unconditionally

🐛 Bug

Exceptions should not be created without being thrown

🐛 Bug

Array or Countable object count comparisons should make sense

🐛 Bug