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...

Constants should not be redefined
Code Smell

Regular expressions should not contain empty groups
Code Smell

Regular expressions should not contain multiple spaces
Code Smell

Single-character alternations in regular expressions should be replaced with character classes
Code Smell

Reluctant quantifiers in regular expressions should be followed by an expression that can't match the empty string
Code Smell

Character classes in regular expressions should not contain the same character twice
Code Smell

Regular expressions should not be too complicated
Code Smell

PHPUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion
Code Smell

Methods should not have identical implementations
Code Smell

Functions should use "return" consistently
Code Smell

Assertion arguments should be passed in the correct order

Secret keys and salt values should be robust

Analyze your code

Vulnerability | Major | owasp

Secret keys are used in combination with an algorithm to encrypt data. A typical use case is an authentication system. For such a system to be secure, the secret key should have a value which cannot be guessed and which is long enough to not be vulnerable to brute-force attacks.

A "salt" is an extra piece of data which is included when hashing data such as a password. Its value should have the same properties as a secret key.

This rule raises an issue when it detects that a secret key or a salt has a predictable value or that it's not long enough.

Noncompliant Code Example

WordPress:

```
define('AUTH_KEY', 'hello'); // Noncompliant
define('AUTH_SALT', 'hello'); // Noncompliant
define('AUTH_KEY', 'put your unique phrase here'); // No
```

Compliant Solution

WordPress:

```
define('AUTH_KEY', 'D&ovlU#|CvJ##uNq}bel+^MFtT&.b9{UvR]g
define('AUTH_SALT', 'FIsAsXJKL5ZlQo)iD-pt??eUbdc{_Cn<4!d
```

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- wordpress.org - WordPress Security Keys

Available In:

sonarlint | sonarcloud | sonarqube

passed in the correct order

☢ Code Smell

Ternary operators should not be nested

☢ Code Smell

Reflection should not be used to increase accessibility of classes, methods, or fields

☢ Code Smell

Multiline blocks should be enclosed in curly braces

☢ Code Smell

Parameters should be passed in the correct order