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...

be used to end lines
Code Smell

More than one property should not be declared per statement
Code Smell

The "var" keyword should not be used
Code Smell

"<?php" and "<?=" tags should be used
Code Smell

File names should comply with a naming convention
Code Smell

Comments should not be located at the end of lines of code
Code Smell

Local variable and function parameter names should comply with a naming convention
Code Smell

Field names should comply with a naming convention
Code Smell

Lines should not end with trailing whitespaces
Code Smell

Files should contain an empty newline at the end
Code Smell

Modifiers should be declared in the correct order
Code Smell

An open curly brace should be located at the beginning of a line

"allow_url_fopen" and "allow_url_include" should be disabled

Analyze your code

Vulnerability | Blocker | cwe owasp sans-top25 php-ini

`allow_url_fopen` and `allow_url_include` allow code to be read into a script from URL's. The ability to suck in executable code from outside your site, coupled with imperfect input cleansing could lay your site bare to attackers. Even if your input filtering is perfect today, are you prepared to bet your site that it will always be perfect in the future?

This rule raises an issue when either property is explicitly enabled in *php.ini* and when `allow_url_fopen`, which defaults to enabled, is not explicitly disabled.

Noncompliant Code Example

```
; php.ini  Noncompliant; allow_url_fopen not explicitly
allow_url_include=1  ; Noncompliant
```

Compliant Solution

```
; php.ini
allow_url_fopen=0
allow_url_include=0
```

See

- OWASP Top 10 2021 Category A3 - Injection
- OWASP Top 10 2021 Category A8 - Software and Data Integrity Failures
- OWASP Top 10 2017 Category A1 - Injection
- MITRE, CWE-829 - Inclusion of Functionality from Untrusted Control Sphere
- SANS Top 25 - Risky Resource Management

Available In:

sonarlint | sonarcloud | sonarqube

Code Smell

An open curly brace should be located at the end of a line

Code Smell

Tabulation characters should not be used

Code Smell

Method and function names should comply with a naming convention

Code Smell

Creating cookies with broadly defined "domain" flags is security-sensitive

Security Hotspot