

- 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...

Code Smell
Overriding methods should do more than simply call the same method in the super class
Code Smell
"empty()" should be used to test for emptiness
Code Smell
Interface names should comply with a naming convention
Code Smell
Return of boolean expressions should not be wrapped into an "if-then-else" statement
Code Smell
Boolean literals should not be redundant
Code Smell
Empty statements should be removed
Code Smell
A close curly brace should be located at the beginning of a line
Code Smell
URIs should not be hardcoded
Code Smell
Class names should comply with a naming convention
Code Smell
Track uses of "TODO" tags
Code Smell
"file_uploads" should be disabled
Vulnerability

Allowing unauthenticated database repair in WordPress is security-sensitive

Analyze your code

Security Hotspot Major owasp

WordPress has a database repair and optimization mode that can be activated by setting WP_ALLOW_REPAIR to true in the configuration.

If activated, the repair page can be accessed by any user, authenticated or not. This makes sense because if the database is corrupted, the authentication mechanism might not work.

Malicious users could trigger this potentially costly operation repeatedly slowing down the website, and making it unavailable.

Ask Yourself Whether

- The database is not currently corrupted.

There is a risk if you answered yes to this question.

Recommended Secure Coding Practices

It's recommended to enable automatic database repair mode only in case of database corruption. This feature should be deactivated again when the database issue is resolved.

Noncompliant Code Example

```
define( 'WP_ALLOW_REPAIR', true ); // Sensitive
```

Compliant Solution


```
// The default value is false, so the value does not ha
define( 'WP_ALLOW_REPAIR', false );
```

See


- OWASP Top 10 2021 Category A5 - Security Misconfiguration
- OWASP Top 10 2021 Category A7 - Identification and Authentication Failures
- wordpress.org - Automatic Database Optimizing
- OWASP Top 10 2017 Category A6 - Security Misconfiguration

Available In: sonarcloud sonarqube


"enable_dl" should be disabled

 Vulnerability


"session.use_trans_sid" should not be enabled

 Vulnerability

**"allow_url_fopen" and
"allow_url_include" should be disabled**

 Vulnerability

"open_basedir" should limit file access

 Vulnerability

**Neither DES (Data Encryption
Standard) nor DESede (3DES) should
be used**

© 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](#)