

- 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... 🔍

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
"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
Vulnerability
"exit(...)" and "die(...)" statements should not be used
Bug

WordPress theme and plugin editors are security-sensitive

Analyze your code

Security Hotspot

Major ?

cwe owasp

WordPress makes it possible to edit theme and plugin files directly in the Administration Screens. While it may look like an easy way to customize a theme or do a quick change, it's a dangerous feature. When visiting the theme or plugin editor for the first time, WordPress displays a warning to make it clear that using such a feature may break the web site by mistake. More importantly, users who have access to this feature can trigger the execution of any PHP code and may therefore take full control of the WordPress instance. This security risk could be exploited by an attacker who manages to get access to one of the authorized users. Setting the `DISALLOW_FILE_EDIT` option to `true` in `wp-config.php` disables this risky feature. The default value is `false`.

Ask Yourself Whether

- You really need to use the theme and plugin editors.
- The theme and plugin editors are available to users who cannot be fully trusted.
- There's a chance that the accounts of authorized users get compromised.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

- Modify the theme and plugin files using a local editor and deploy them to the server in a secure way.
- Make sure that `DISALLOW_FILE_EDIT` is defined in `wp-config.php`.
- Make sure that `DISALLOW_FILE_EDIT` is set to `true`.

Sensitive Code Example

```
define( 'DISALLOW_FILE_EDIT', false ); // Sensitive
```


Compliant Solution

```
define( 'DISALLOW_FILE_EDIT', true );
```


See

- [OWASP Top 10 2021 Category A3](#) - Injection
- [OWASP Top 10 2021 Category A4](#) - Insecure Design
- [OWASP Top 10 2021 Category A5](#) - Security Misconfiguration
- [wordpress.org](#) - Disable the Plugin and Theme Editor
- [OWASP Top 10 2017 Category A1](#) - Injection
- [OWASP Top 10 2017 Category A6](#) - Security Misconfiguration
- [OWASP Top 10 2017 Category A7](#) - Cross-Site Scripting (XSS)


Functions and variables should not be defined outside of classes

 Code Smell


Track lack of copyright and license headers

 Code Smell

Octal values should not be used

 Code Smell

Switch cases should end with an unconditional "break" statement

 Code Smell

Session-management cookies should not be persistent

- [MITRE, CWE-79](#) - Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- [MITRE, CWE-94](#) - Improper Control of Generation of Code ('Code Injection')
- [MITRE, CWE-95](#) - Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')

Available In:

**sonarcloud**  | **sonarqube** 

© 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](#)