

- 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...

Operator loop

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 unfiltered HTML content in WordPress is security-sensitive

Analyze your code

Security Hotspot Major ? cwe owasp

By default, the WordPress administrator and editor roles can add unfiltered HTML content in various places, such as post content. This includes the capability to add JavaScript code.

If an account with such a role gets hijacked, this capability can be used to plant malicious JavaScript code that gets executed whenever somebody visits the website.

Ask Yourself Whether

- You really need the possibility to add unfiltered HTML with editor or administrator roles.
- 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

The `unfiltered_html` capability should be granted to trusted roles that need to use markup when publishing dynamic content to the WordPress website. If this capability is not required for all users, including administrators and editors roles, then it's recommended to set `DISALLOW_UNFILTERED_HTML` to `true`.

Noncompliant Code Example

```
define( 'DISALLOW_UNFILTERED_HTML', false ); // sensitive
```

Compliant Solution





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

See

- [OWASP Top 10 2021 Category A3](#) - Injection
- [OWASP Top 10 2021 Category A5](#) - Security Misconfiguration
- [OWASP Top 10 2017 Category A6](#) - Security Misconfiguration
- [OWASP Top 10 2017 Category A7](#) - Cross-Site Scripting (XSS)
- [MITRE, CWE-79](#) - Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

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

© 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](#)