

- 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

Writing cookies is security-sensitive

Analyze your code

Security Hotspot Minor

Using cookies is security-sensitive. It has led in the past to the following vulnerabilities:

- CVE-2018-11639
- CVE-2016-6537

Attackers can use widely-available tools to read cookies. Any sensitive information they may contain will be exposed.

This rule flags code that writes cookies.

Ask Yourself Whether

- sensitive information is stored inside the cookie.

You are at risk if you answered yes to this question.

Recommended Secure Coding Practices

Cookies should only be used to manage the user session. The best practice is to keep all user-related information server-side and link them to the user session, never sending them to the client. In a very few corner cases, cookies can be used for non-sensitive information that need to live longer than the user session.

Do not try to encode sensitive information in a non human-readable format before writing them in a cookie. The encoding can be reverted and the original information will be exposed.

Using cookies only for session IDs doesn't make them secure. Follow OWASP best practices when you configure your cookies.

As a side note, every information read from a cookie should be Sanitized.

Sensitive Code Example

```
$value = "1234 1234 1234 1234";

// Review this cookie as it seems to send sensitive info
setcookie("CreditCardNumber", $value, $expire, $path, $
setrawcookie("CreditCardNumber", $value, $expire, $path
```

See

- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- MITRE, CWE-312 - Cleartext Storage of Sensitive Information
- MITRE, CWE-315 - Cleartext Storage of Sensitive Information in a Cookie
- Derived from FindSecBugs rule COOKIE_USAGE

Deprecated

 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

This rule is deprecated, and will eventually be removed.

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](#)