- 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

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

Analyze your code

Security Hotspot Info

A cookie’s domain specifies which websites should be able to read it. Left blank, browsers are supposed to only send the cookie to sites that exactly match the sending domain. For example, if a cookie was set by *lovely.dream.com*, it should only be readable by that domain, and not by *nightmare.com* or even *strange.dream.com*. If you want to allow sub-domain access for a cookie, you can specify it by adding a dot in front of the cookie’s domain, like so: *.dream.com*. But cookie domains should always use at least two levels.

Cookie domains can be set either programmatically or via configuration. This rule raises an issue when any cookie domain is set with a single level, as in *.com*.

Ask Yourself Whether

- the `domain` attribute has only one level as domain naming.

You are at risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

- You should check the `domain` attribute has been set and its value has more than one level of domain nanimg, like: *sonarsource.com*

Noncompliant Code Example

```
setcookie("TestCookie", $value, time()+3600, "/~path/",
session_set_cookie_params(3600, "/~path/", ".com"); //

// inside php.ini
session.cookie_domain=".com"; // Noncompliant
```

Compliant Solution

```
setcookie("TestCookie", $value, time()+3600, "/~path/",
session_set_cookie_params(3600, "/~path/", ".myDomain.c

// inside php.ini
session.cookie_domain=".myDomain.com";
```

See

- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure

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