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
Code Smell

Creating cookies without the "HttpOnly" flag is security-sensitive

Analyze your code

Security Hotspot | Minor | cwe sans-top25 privacy owasp

When a cookie is configured with the `HttpOnly` attribute set to *true*, the browser guaranties that no client-side script will be able to read it. In most cases, when a cookie is created, the default value of `HttpOnly` is *false* and it's up to the developer to decide whether or not the content of the cookie can be read by the client-side script. As a majority of Cross-Site Scripting (XSS) attacks target the theft of session-cookies, the `HttpOnly` attribute can help to reduce their impact as it won't be possible to exploit the XSS vulnerability to steal session-cookies.

Ask Yourself Whether

- the cookie is sensitive, used to authenticate the user, for instance a *session-cookie*
- the `HttpOnly` attribute offer an additional protection (not the case for an *XSRF-TOKEN cookie* / CSRF token for example)

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

Recommended Secure Coding Practices

- By default the `HttpOnly` flag should be set to *true* for most of the cookies and it's mandatory for session / sensitive-security cookies.

Sensitive Code Example

In *php.ini* you can specify the flags for the session cookie which is security-sensitive:

```
session.cookie_httponly = 0;  // Sensitive: this sensitive s
```

Same thing in PHP code:

```
session_set_cookie_params($lifetime, $path, $domain, true, f
```

If you create a custom security-sensitive cookie in your PHP code:

```
$value = "sensitive data";
setcookie($name, $value, $expire, $path, $domain, true, fals
```

By default `setcookie` and `setrawcookie` functions set `httpOnly` flag to *false* (the seventh argument) and so cookies can be stolen easily in case of XSS vulnerability:

```
$value = "sensitive data";
setcookie($name, $value, $expire, $path, $domain, true); //
setrawcookie($name, $value, $expire, $path, $domain, true);
```

Compliant Solution

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

```
session.cookie_httponly = 1; // Compliant: the sensitive coo
```

```
session_set_cookie_params($lifetime, $path, $domain, true, t
```

```
$value = "sensitive data";
setcookie($name, $value, $expire, $path, $domain, true, true
setrawcookie($name, $value, $expire, $path, $domain, true, t
```

See

- OWASP Top 10 2021 Category A5 - Security Misconfiguration
- OWASP HttpOnly
- OWASP Top 10 2017 Category A7 - Cross-Site Scripting (XSS)
- MITRE, CWE-1004 - Sensitive Cookie Without 'HttpOnly' Flag
- SANS Top 25 - Insecure Interaction Between Components
- Derived from FindSecBugs rule HTTPONLY_COOKIE

Available In:

sonarcloud | sonarqube