

- 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...



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

OS commands should not be vulnerable to argument injection attacks

Analyze your code

Vulnerability Minor injection cwe owasp sans-top25

Applications that allow execution of operating system commands from user-controlled data should control the arguments passed to the command, otherwise an attacker can inject additional arbitrary arguments which can change the behavior of the command.

User-controlled arguments should be sanitized by neutralizing argument delimiters (eg: ' , space, -) and thus preventing injection of unwanted additional arguments. A single user-controlled argument may still lead to vulnerabilities if it corresponds to a dangerous option supported by the command, such as -exec available with find, in that case, mark end of option processing on the command line using -- (double-dash) or restrict the options to only trusted values.

Noncompliant Code Example

escapeshellcmd doesn't prevent additional arguments from being injected:

```
/*
    here the attacker can cat any files, not only the ones su
    by passing additional arguments such as "data_private.csv
*/
exec(escapeshellcmd("/usr/bin/cat /tmp/" . $_GET["arg"] . "_publ
```

escapeshellcmd cancels any prior sanitization with escapeshellarg:

```
exec(escapeshellcmd("/usr/bin/cat " . escapeshellarg("/tmp/" . $_
```

Fifth argument of the mail function accepts arguments that are added to the command line:

```
mail($to, $subject, $message, $params, $_GET["arg"]); // Non
// by default php sanitizes the fifth argument with escapesh
mail($to, $subject, $message, $params, escapeshellarg($_GET[
```

Compliant Solution

escapeshellarg should be used to sanitize a specific argument:

```
exec("/usr/bin/cat " . escapeshellarg("/tmp/" . $_GET["arg"] . "_p
```


Fifth argument of the mail function can be secured with the use of an allow-list:

```
$arg = $_GET["arg"];
if ($arg === "something1" || $arg === "something2") {
    mail($to, $subject, $message, $params, $arg);
}
```

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

#### See

- OWASP OS Command Injection Defense [Cheat Sheet](#)
- [OWASP Top 10 2021 Category A3](#) - Injection
- [OWASP Top 10 2017 Category A1](#) - Injection
- [MITRE, CWE-88](#) - Argument Injection or Modification
- [blog.sonarsource.com](#) - Why mail() is dangerous in PHP
- [SANS Top 25](#) - Insecure Interaction Between Components

Available In:

**sonarcloud**  | **sonarqube**  Developer Edition

© 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](#)