sonar RULES

Products

- 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

"open_basedir" should limit file access

Analyze your code

Vulnerability Blocker cwe owasp php-ini

The `open_basedir` configuration in *php.ini* limits the files the script can access using, for example, `include` and `fopen()`. Leave it out, and there is no default limit, meaning that any file can be accessed. Include it, and PHP will refuse to access files outside the allowed path.

`open_basedir` should be configured with a directory, which will then be accessible recursively. However, the use of `.` (current directory) as an `open_basedir` value should be avoided since it's resolved dynamically during script execution, so a `chdir('/')` command could lay the whole server open to the script.

This is not a fool-proof configuration; it can be reset or overridden at the script level. But its use should be seen as a minimum due diligence step. This rule raises an issue when `open_basedir` is not present in *php.ini*, and when `open_basedir` contains root, or the current directory (`.`) symbol.

Noncompliant Code Example

```
; php.ini try 1
; open_basedir="${USER}/scripts/data"  Noncompliant; co

; php.ini try 2
open_basedir="/:${USER}/scripts/data"  ; Noncompliant;
```

Compliant Solution

```
; php.ini try 1
open_basedir="${USER}/scripts/data"
```

See

- OWASP Top 10 2021 Category A1 - Broken Access Control
- OWASP Top 10 2021 Category A5 - Security Misconfiguration
- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- MITRE, CWE-23 - Relative Path Traversal
- MITRE, CWE-36 - Absolute Path Traversal

Available In:

sonarlint sonarcloud sonarqube

Privacy Policy

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