

- 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

### "exit(...)" and "die(...)" statements should not be used

Analyze your code

Bug Blocker

The `exit(...)` and `die(...)` statements should absolutely not be used in Web PHP pages as this might lead to a very bad user experience. In such case, the end user might have the feeling that the web site is down or has encountered a fatal error.

But of course PHP can also be used to develop command line application and in such case use of `exit(...)` or `die(...)` statement can be justified but must remain limited and not spread all over the application. We expect exceptions to be used to handle errors and those exceptions should be caught just before leaving the application to specify the exit code with help of `exit(...)` or `die(...)` statements.

#### Noncompliant Code Example

```
class Foo {
    public function bar($param) {
        if ($param === 42) {
            exit(23);
        }
    }
}
```

#### Compliant Solution

```
class Foo {
    public function bar($param) {
        if ($param === 42) {
            throw new Exception('Value 42 is not expected');
        }
    }
}
```

Available In:

sonarlint

| sonarcloud

| sonarqube

 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