
































-  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... 

Identical expressions should not be used on both sides of a binary operator

All code should be reachable

Loops with at most one iteration should be refactored

Short-circuit logic should be used to prevent null pointer dereferences in conditionals

Variables should not be self-assigned

Useless "if(true){...}" and "if(false){...}" blocks should be removed

All "catch" blocks should be able to catch exceptions

Constructing arguments of system commands from user input is security-sensitive

Allowing unfiltered HTML content in WordPress is security-sensitive

Allowing unauthenticated database repair in WordPress is security-sensitive

Allowing all external requests from a WordPress server is security-sensitive

"default" clauses should be first or last

Analyze your code

 Code Smell

 Critical



switch can contain a default clause for various reasons: to handle unexpected values, to show that all the cases were properly considered.

For readability purpose, to help a developer to quickly find the default behavior of a switch statement, it is recommended to put the default clause at the end of the switch statement. This rule raises an issue if the default clause is not the first or the last one of the switch's cases.

Noncompliant Code Example

```
switch ($param) {
    case 0:
        doSomething();
        break;
    default: // default clause should be the first or last
        error();
        break;
    case 1:
        doSomethingElse();
        break;
}
```

Compliant Solution

```
switch ($param) {
    case 0:
        doSomething();
        break;
    case 1:
        doSomethingElse();
        break;
    default:
        error();
        break;
}
```

Available In:


 sonarlint

 sonarcloud


 sonarqube

 Security Hotspot


Disabling automatic updates is security-sensitive

 Security Hotspot


WordPress theme and plugin editors are security-sensitive

 Security Hotspot

Allowing requests with excessive content length is security-sensitive

 Security Hotspot

Manual generation of session ID is security-sensitive

 Security Hotspot