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...

Alternatives in regular expressions should be grouped when used with anchors
Bug

Array values should not be replaced unconditionally
Bug

Exceptions should not be created without being thrown
Bug

Array or Countable object count comparisons should make sense
Bug

All branches in a conditional structure should not have exactly the same implementation
Bug

The output of functions that don't return anything should not be used
Bug

Unary prefix operators should not be repeated
Bug

"=+" should not be used instead of "+="
Bug

A "for" loop update clause should move the counter in the right direction
Bug

Return values from functions without side effects should not be ignored
Bug

Values should not be uselessly incremented
Bug

Related "if/else if" statements and "cases" in a "switch" should not have

References used in "foreach" loops should be "unset"

Analyze your code

Bug | Critical | pitfall

When a reference is used in a `foreach` loop instead of using a simple variable, the reference remains assigned and keeps its "value" which is a reference, even after the `foreach` execution. Most of the time, this is not what the developer is expecting and the reference may be used wrongly in the rest of the code. For this reason, it is recommended to always `unset` a reference that is used in a `foreach` to avoid any unexpected side effects.

Noncompliant Code Example

```
$arr = array(1, 2, 3);
foreach ($arr as &$value) { // Noncompliant; $value is still
    $value = $value * 2;
}
$value = 'x';
```

Compliant Solution

```
$arr = array(1, 2, 3);
foreach ($arr as &$value) { // Compliant; there is no risk t
    $value = $value * 2;
}
unset($value);
$value = 'x';
```

See

- PHP Documentation: Foreach
- References and Foreach

Available In:

sonarlint | sonarcloud | sonarqube

the same condition

Bug

Objects should not be created to be dropped immediately without being used

Bug

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

Bug

All code should be reachable

Bug