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

Single-character alternations in regular expressions should be replaced with character classes

Code Smell

Reluctant quantifiers in regular expressions should be followed by an expression that can't match the empty string

Code Smell

Character classes in regular expressions should not contain the same character twice

Code Smell

Regular expressions should not be too complicated

Code Smell

PHPUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion

Code Smell

Methods should not have identical implementations

Code Smell

Functions should use "return" consistently

Code Smell

Assertion arguments should be passed in the correct order

Code Smell

Ternary operators should not be nested

Code Smell

Reflection should not be used to increase accessibility of classes, methods, or fields

Server-side requests should not be vulnerable to forging attacks

Analyze your code

Vulnerability Major injection cwe sans-top25 owasp

User-supplied data, such as URL parameters, POST data payloads, or cookies, should always be considered untrusted and tainted. Performing requests from user-controlled data could allow attackers to make arbitrary requests on the internal network or to change their original meaning and thus to retrieve or delete sensitive information.

The problem could be mitigated in any of the following ways:

- Validate the user-provided data, such as the URL and headers, used to construct the request.
- Redesign the application to not send requests based on user-provided data.

Noncompliant Code Example

```
$url = $_GET["url"];
$resp = file_get_contents($url); // Noncompliant
// ...
```

Compliant Solution

```
$whitelist = array(
  "www.example.com",
  "example.com"
);

$url         = $_GET["url"];
$parsed_url  = parse_url($url);

if (in_array($parsed_url['host'], $whitelist)) {
  $resp = file_get_contents($url);
  // ...
}
```

See

- OWASP Top 10 2021 Category A10 - Server-Side Request Forgery (SSRF)
- OWASP Attack Category - Server Side Request Forgery
- OWASP Top 10 2017 Category A5 - Broken Access Control
- MITRE, CWE-20 - Improper Input Validation
- MITRE, CWE-641 - Improper Restriction of Names for Files and Other Resources
- MITRE, CWE-918 - Server-Side Request Forgery (SSRF)
- SANS Top 25 - Risky Resource Management

Code Smell

Multiline blocks should be enclosed in curly braces

Code Smell

Parameters should be passed in the correct order

Code Smell

Classes named like "Exception" should extend "Exception" or a subclass

Code Smell

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

Code Smell

Available In:

sonarcloud | sonarqube Developer Edition