sonar RULES

Products

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

Logging should not be vulnerable to injection attacks

Analyze your code

Vulnerability · Minor · injection cwe owasp sans-top25

User-provided data, such as URL parameters, POST data payloads or cookies, should always be considered untrusted and tainted. Applications logging tainted data could enable an attacker to inject characters that would break the log file pattern. This could be used to block monitors and SIEM (Security Information and Event Management) systems from detecting other malicious events.

This problem could be mitigated by sanitizing the user-provided data before logging it.

Noncompliant Code Example

```
$data = $_GET["data"];
error_log($data); // Noncompliant
```

Compliant Solution

```
$data = $_GET["data"];
$badchars = array("\n", "\r", "\t");
$safedata = str_replace($badchars, "", $data);
error_log($safedata);
```

See

- OWASP Top 10 2021 Category A9 - Security Logging and Monitoring Failures
- OWASP Cheat Sheet - Logging
- OWASP Attack Category - Log Injection
- OWASP Top 10 2017 Category A1 - Injection
- MITRE, CWE-20 - Improper Input Validation
- MITRE, CWE-117 - Improper Output Neutralization for Logs
- SANS Top 25 - Insecure Interaction Between Components

Available In:
sonarcloud sonarqube Developer Edition

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