

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- 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... 🔍

``str_replace`` should be preferred to ``preg_replace``

Code Smell

"default" clauses should be first or last

Code Smell

A conditionally executed single line should be denoted by indentation

Code Smell

Conditionals should start on new lines

Code Smell

Cognitive Complexity of functions should not be too high

Code Smell

Parentheses should not be used for calls to "echo"

Code Smell

Functions should not be nested too deeply

Code Smell

References should not be passed to function calls

Code Smell

"switch" statements should have "default" clauses

Code Smell

Control structures should use curly braces

Code Smell

String literals should not be duplicated

Code Smell

Methods should not be empty

## Server certificates should be verified during SSL/TLS connections

Analyze your code

Vulnerability Critical cwe privacy owasp ssl

Validation of X.509 certificates is essential to create secure SSL/TLS sessions not vulnerable to man-in-the-middle attacks.

The certificate chain validation includes these steps:

- The certificate is issued by its parent Certificate Authority or the root CA trusted by the system.
- Each CA is allowed to issue certificates.
- Each certificate in the chain is not expired.

It's not recommended to reinvent the wheel by implementing custom certificate chain validation.

TLS libraries provide built-in certificate validation functions that should be used.

### Noncompliant Code Example

```
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE); // N
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0); // Nonc
```

### Compliant Solution






```
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, TRUE); // Co
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 1); // Comp
```

### See

- [OWASP Top 10 2021 Category A2](#) - Cryptographic Failures
- [OWASP Top 10 2021 Category A5](#) - Security Misconfiguration
- [OWASP Top 10 2021 Category A7](#) - Identification and Authentication Failures
- [OWASP Top 10 2017 Category A3](#) - Sensitive Data Exposure
- [OWASP Top 10 2017 Category A6](#) - Security Misconfiguration
- [Mobile AppSec Verification Standard](#) - Network Communication Requirements
- [OWASP Mobile Top 10 2016 Category M3](#) - Insecure Communication
- [MITRE, CWE-295](#) - Improper Certificate Validation

Available In:

sonarlint | sonarcloud | sonarqube

 Code Smell
<b>Constant names should comply with a naming convention</b>  Code Smell
<b>Secret keys and salt values should be robust</b>  Vulnerability
<b>Authorizations should be based on strong decisions</b>  Vulnerability
<b>Server-side requests should not be vulnerable to forging attacks</b>  Vulnerability

trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.  
[Privacy Policy](#)