
































-  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... 

Dynamically executing code is security-sensitive	Security Hotspot
`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

Server hostnames should be verified during SSL/TLS connections

Analyze your code

Vulnerability Critical cwe privacy owasp ssl

To establish a SSL/TLS connection not vulnerable to man-in-the-middle attacks, it's essential to make sure the server presents the right certificate.

The certificate's hostname-specific data should match the server hostname.

It's not recommended to re-invent the wheel by implementing custom hostname verification.

TLS/SSL libraries provide built-in hostname verification functions that should be used.

Noncompliant Code Example

```
curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, FALSE); // Noncompliant
curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0); // Noncompliant
```

Compliant Solution

```
curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 2); // Compliant;
curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, TRUE); // Compliant
```

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-297](#) - Improper Validation of Certificate with Host Mismatch


Available In:

sonarlint | sonarcloud | sonarqube


Methods should not be empty

 Code Smell


Constant names should comply with a naming convention

 Code Smell

Secret keys and salt values should be robust

 Vulnerability

Authorizations should be based on strong decisions

 Vulnerability