

- 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...



Raised Exceptions must derive from Throwable

Bug

"\$this" should not be used in a static context

Bug

Hard-coded credentials are security-sensitive

Security Hotspot

Test class names should end with "Test"

Code Smell

Tests should include assertions

Code Smell

TestCases should contain tests

Code Smell

Variable variables should not be used

Code Smell

A new session should be created during user authentication

Vulnerability

Cipher algorithms should be robust

Vulnerability

Encryption algorithms should be used with secure mode and padding scheme

Vulnerability

Server hostnames should be verified during SSL/TLS connections

Vulnerability

Server certificates should be verified during SSL/TLS connections

Vulnerability

A secure password should be used when connecting to a database

Analyze your code

Vulnerability Blocker cwe owasp

When relying on the password authentication mode for the database connection, a secure password should be chosen.

This rule raises an issue when an empty password is used.

Noncompliant Code Example

```
// example of an empty password when connecting to a mysql d
$conn = new mysqli($servername, $username, "");
```

Compliant Solution

```
// generate a secure password, set it to the username databa
$password = getenv('MYSQL_SECURE_PASSWORD');
// then connect to the database
$conn = new mysqli($servername, $username, $password);
```


See

- [OWASP Top 10 2021 Category A7](#) - Identification and Authentication Failures
- [OWASP Top 10 2017 Category A2](#) - Broken Authentication
- [OWASP Top 10 2017 Category A3](#) - Sensitive Data Exposure
- [MITRE, CWE-521](#) - Weak Password Requirements


Available In:

sonarlint | sonarcloud | sonarqube


LDAP connections should be authenticated

 Vulnerability


Cryptographic keys should be robust

 Vulnerability

Weak SSL/TLS protocols should not be used

 Vulnerability

Regular expressions should not be vulnerable to Denial of Service attacks

 Vulnerability