- 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...

Octal values should not be used

Code Smell

Switch cases should end with an unconditional "break" statement

Code Smell

Session-management cookies should not be persistent

Vulnerability

Cryptographic RSA algorithms should always incorporate OAEP (Optimal Asymmetric Encryption Padding)

Vulnerability

SHA-1 and Message-Digest hash algorithms should not be used in secure contexts

Vulnerability

Assertions should not be made at the end of blocks expecting an exception

Bug

Regular expressions should be syntactically valid

Bug

Only one method invocation is expected when testing exceptions

Bug

Reading the Standard Input is security-sensitive

Security Hotspot

Using command line arguments is security-sensitive

Security Hotspot

Using Sockets is security-sensitive

Security Hotspot

Character classes in regular expressions should not contain only one character

Analyze your code

Code Smell Major regex

Character classes in regular expressions are a convenient way to match one of several possible characters by listing the allowed characters or ranges of characters. If a character class contains only one character, the effect is the same as just writing the character without a character class.

Thus, having only one character in a character class is usually a simple oversight that remained after removing other characters of the class.

Noncompliant Code Example

```
/a[b]c/
```

Compliant Solution

```
/abc/
```

Exceptions

This rule does not raise when the character inside the class is a metacharacter. This notation is sometimes used to avoid escaping (e.g., `[.]{3}` to match three dots).

Available In:

sonarlint sonarcloud sonarqube

Encrypting data is security-sensitive

Security Hotspot

Using regular expressions is security-sensitive

Security Hotspot

Deserializing objects from an untrusted source is security-sensitive

Security Hotspot

Literal boolean values and nulls should not be used in equality assertions

Code Smell

"global" should not be used