

- 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



Python static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PYTHON code

All rules 216

Vulnerability 29

Bug 55

Security Hotspot 31

Code Smell 101

Tags ▾

Search by name...

	Security Hotspot
Creating cookies without the "HttpOnly" flag is security-sensitive	Security Hotspot
Creating cookies without the "secure" flag is security-sensitive	Security Hotspot
Using hardcoded IP addresses is security-sensitive	Security Hotspot
Regular expression quantifiers and character classes should be used concisely	Code Smell
Character classes should be preferred over reluctant quantifiers in regular expressions	Code Smell
A subclass should not be in the same "except" statement as a parent class	Code Smell
Walrus operator should not make code confusing	Code Smell
Jump statements should not be redundant	Code Smell
"pass" should not be used needlessly	Code Smell
"except" clauses should do more than raise the same issue	Code Smell
Boolean checks should not be inverted	Code Smell

Alternation in regular expressions should not contain empty alternatives

Analyze your code

Bug

Major

regex

Alternation is used to match a single regular expression out of several possible regular expressions. If one of the alternatives is empty it would match any input, which is most probably a mistake.

Noncompliant Code Example

```
re.search(r"Jack|Peter|", "John") # Noncompliant - will match
re.search(r"Jack||Peter", "John") # Noncompliant - will match
```

Compliant Solution

```
re.search(r"Jack|Peter", "John") # returns false
```

Exceptions

One could use an empty alternation to make a regular expression group optional. Rule will not report on such cases.

```
re.search(r"mandatory(-optional|)", "mandatory")
re.search(r"mandatory(-optional|)", "mandatory-optional")
```

However, if there is a quantifier after the group the issue will be reported as using both | and quantifier is redundant.

```
re.search(r"mandatory(-optional|)?", "mandatory-optional") #
```

Available In:

sonarlint

sonarcloud

sonarqube

Unused local variables should be removed

 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

Class names should comply with a naming convention