

- 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...



Method names should comply with a naming convention

Code Smell

Track uses of "TODO" tags

Code Smell

HTML autoescape mechanism should not be globally disabled

Vulnerability

Variables, classes and functions should be either defined or imported

Bug

"__exit__" should accept type, value, and traceback arguments

Bug

"return" and "yield" should not be used in the same function

Bug

Track lack of copyright and license headers

Code Smell

HTTP response headers should not be vulnerable to injection attacks

Vulnerability

Regular expressions should be syntactically valid

Bug

Sending emails is security-sensitive

Security Hotspot

Reading the Standard Input is security-sensitive

Security Hotspot

Using command line arguments is security-sensitive

All branches in a conditional structure should not have exactly the same implementation

Analyze your code

Bug Major

Having the same implementation in every branch of an `if` chain is an error. Either a copy-paste error was made and something different should be executed, or there shouldn't be an `if` chain at all.

Noncompliant Code Example

```
if b == 0: # Noncompliant
    doOneMoreThing()
elif b == 1:
    doOneMoreThing()
else:
    doOneMoreThing()

b = 4 if a > 12 else 4 # Noncompliant
```

Exceptions


This rule does not apply to `if` chains without `else`-s.

```
if b == 0: # no issue, this could have been done on pu
    doOneMoreThing()
elif b == 1:
    doOneMoreThing()
```

Available In:

sonarlint | **sonarcloud** | **sonarqube**


© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved. [Privacy Policy](#)

 Security Hotspot


Encrypting data is security-sensitive

 Security Hotspot


Using regular expressions is security-sensitive

 Security Hotspot

Dynamically executing code is security-sensitive

 Security Hotspot

Cyclomatic Complexity of functions should not be too high

 Code Smell