

- 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...

Code Smell
Logging should not be vulnerable to injection attacks Vulnerability
Repeated patterns in regular expressions should not match the empty string Bug
Function parameters initial values should not be ignored Bug
Disabling versioning of S3 buckets is security-sensitive Security Hotspot
Disabling server-side encryption of S3 buckets is security-sensitive Security Hotspot
Having a permissive Cross-Origin Resource Sharing policy is security-sensitive Security Hotspot
Delivering code in production with debug features activated is security-sensitive Security Hotspot
Allowing both safe and unsafe HTTP methods is security-sensitive 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

String literals should not be duplicated

Analyze your code

Code Smell

Critical

design

Duplicated string literals make the process of refactoring error-prone, since you must be sure to update all occurrences.

On the other hand, constants can be referenced from many places, but only need to be updated in a single place.

Noncompliant Code Example

With the default threshold of 3:

```
def run():
    prepare("this is a duplicate") # Noncompliant - "this
    execute("this is a duplicate")
    release("this is a duplicate")
```

Compliant Solution

```
ACTION_1 = "action1"

def run():
    prepare(ACTION_1)
    execute(ACTION_1)
    release(ACTION_1)
```

Exceptions

No issue will be raised on:

- duplicated string in decorators
- strings with less than 5 characters
- strings with only letters, numbers and underscores

```
@app.route("/api/users/", methods=['GET', 'POST', 'PUT'])
def users():
    pass

@app.route("/api/projects/", methods=['GET', 'POST', 'PUT'])
def projects():
    pass
```


Available In:

sonarlint

sonarcloud

sonarqube


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