

- 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... 🔍

Signalling processes is security-sensitive	Security Hotspot
Configuring loggers is security-sensitive	Security Hotspot
Using weak hashing algorithms is security-sensitive	Security Hotspot
Disabling CSRF protections is security-sensitive	Security Hotspot
Using non-standard cryptographic algorithms is security-sensitive	Security Hotspot
Using pseudorandom number generators (PRNGs) is security-sensitive	Security Hotspot
Constants should not be used as conditions	Code Smell
"SystemExit" should be re-raised	Code Smell
Bare "raise" statements should only be used in "except" blocks	Code Smell
Comparison to None should not be constant	Code Smell
"self" should be the first argument to instance methods	Code Smell

Recursion should not be infinite

Analyze your code

Bug

Blocker

suspicious

Recursion happens when control enters a loop that has no exit. This can happen when a method invokes itself or when a pair of methods invoke each other. It can be a useful tool, but unless the method includes a provision to break out of the recursion and return, the recursion will continue until the stack overflows and the program crashes.

Noncompliant Code Example

```
def my_pow(num, exponent): # Noncompliant
    num = num * my_pow(num, exponent - 1)
    return num # this is never reached
```

Compliant Solution

```
def my_pow(num, exponent): # Compliant
    if exponent > 1:
        num = num * my_pow(num, exponent - 1)
    return num
```

Available In:
sonarcloud

Function parameters' default values should not be modified or assigned

 Code Smell

Some special methods should return "NotImplemented" instead of raising "NotImplementedError"

 Code Smell

Custom Exception classes should inherit from "Exception" or one of its subclasses

 Code Smell

Bare "raise" statements should not be used in "finally" blocks

 Code Smell