

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- 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...

Bug

Function arguments should be passed only once

Bug

Iterable unpacking, "for-in" loops and "yield from" should use an Iterable object

Bug

Variables, classes and functions should be defined before being used

Bug

Identity operators should not be used with dissimilar types

Bug

Only strings should be listed in "__all__"

Bug

"__init__" should not return a value

Bug

"yield" and "return" should not be used outside functions

Bug

String formatting should not lead to runtime errors

Bug

Recursion should not be infinite

Bug

Silly equality checks should not be made

Bug

Granting access to S3 buckets to all or authenticated users is security-sensitive

The number and name of arguments passed to a function should match its parameters

Analyze your code

Bug Blocker cwe

Calling a function or a method with fewer or more arguments than expected will raise a `TypeError`. This is usually a bug and should be fixed.

Noncompliant Code Example

```
#####
# Positional Arguments
#####

param_args = [1, 2, 3]
param_kwargs = {'x': 1, 'y': 2}

def func(a, b=1):
    print(a, b)

def positional_unlimited(a, b=1, *args):
    print(a, b, *args)

func(1)
func(1, 42)
func(1, 2, 3) # Noncompliant. Too many positional arguments
func() # Noncompliant. Missing positional argument for 'b'

positional_unlimited(1, 2, 3, 4, 5)

def positional_limited(a, *, b=2):
    print(a, b)

positional_limited(1, 2) # Noncompliant. Too many positional arguments

#####
# Unexpected Keyword argument
#####






def keywords(a=1, b=2, *, c=3):
    print(a, b, c)

keywords(1)
keywords(1, z=42) # Noncompliant. Unexpected keyword argument 'z'

def keywords_unlimited(a=1, b=2, *, c=3, **kwargs):
    print(a, b, kwargs)

keywords_unlimited(a=1, b=2, z=42)

#####
```

 Security Hotspot
Hard-coded credentials are security-sensitive  Security Hotspot
Functions returns should not be invariant  Code Smell
The "exec" statement should not be used  Code Smell
Backticks should not be used  Code Smell

```
# Mandatory Keyword argument only
#####

def mandatory_keyword(a, *, b):
    print(a, b)

mandatory_keyword(1, b=2)
mandatory_keyword(1) # Noncompliant. Missing keyword a
```

See

- [MITRE, CWE-628](#) - Function Call with Incorrectly Specified Arguments

Available In:

 |  | 

© 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](#)