- 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...

"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

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

Methods and field names should not differ only by capitalization

Code Smell

JWT should be signed and verified

Vulnerability

Calls should not be made to non-callable values

Analyze your code

Bug Blocker

In order to be callable, a python class should implement the `__call__` method.

This rule raises an issue when a non-callable object is called.

Noncompliant Code Example

```
class MyClass:
    pass

myvar = MyClass()
myvar()  # Noncompliant

none_var = None
none_var()  # Noncompliant
```

Compliant Solution

```
class MyClass:
    def __call__(self):
        print("called")

myvar = MyClass()
myvar()
```

See

- Python documentation - __call__ method

Available In:

sonarlint | sonarcloud | sonarqube

Cipher algorithms should be robust

Vulnerability

Encryption algorithms should be used with secure mode and padding scheme

Vulnerability

Server hostnames should be verified during SSL/TLS connections

Vulnerability

Insecure temporary file creation methods should not be used

Vulnerability

Server certificates should be verified during SSL/TLS connections