

- 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...

Functions should not have too many lines of code

Code Smell

Track uses of "NOSONAR" comments

Code Smell

Track comments matching a regular expression

Code Smell

Statements should be on separate lines

Code Smell

Functions should not contain too many return statements

Code Smell

Files should not have too many lines of code

Code Smell

Lines should not be too long

Code Smell

Methods and properties that don't access instance data should be static

Code Smell

New-style classes should be used

Code Smell

Parentheses should not be used after certain keywords

Code Smell

Track "TODO" and "FIXME" comments that do not contain a reference to a person

Code Smell

Module names should comply with a naming convention

Code Smell

Reading the Standard Input is security-sensitive

Analyze your code

Security Hotspot Critical

Reading Standard Input is security-sensitive. It has led in the past to the following vulnerabilities:

- CVE-2005-2337
- CVE-2017-11449

It is common for attackers to craft inputs enabling them to exploit software vulnerabilities. Thus any data read from the standard input (stdin) can be dangerous and should be validated.

This rule flags code that reads from the standard input.

Ask Yourself Whether

- data read from the standard input is not sanitized before being used.

You are at risk if you answered yes to this question.

Recommended Secure Coding Practices

Sanitize all data read from the standard input before using it.

Sensitive Code Example

Python 2 and Python 3

```
import sys
from sys import stdin, __stdin__

# Any reference to sys.stdin or sys.__stdin__ without a method
sys.stdin # Sensitive

for line in sys.stdin: # Sensitive
    print(line)

it = iter(sys.stdin) # Sensitive
line = next(it)

# Calling the following methods on stdin or __stdin__ is sensitive
sys.stdin.read() # Sensitive
sys.stdin.readline() # Sensitive
sys.stdin.readlines() # Sensitive


# Calling other methods on stdin or __stdin__ does not require
sys.stdin.seekable() # Ok
# ...
```

Python 2 only


```
raw_input('What is your password?') # Sensitive
```

Python 3 only

Comments should not be located at the end of lines of code

 Code Smell

Lines should not end with trailing whitespaces

 Code Smell

Files should contain an empty newline at the end

 Code Smell

Long suffix "L" should be upper case

 Code Smell

```
input('What is your password?') # Sensitive
```

Function `fileinput.input` and class `fileinput.FileInput` read the standard input when the list of files is empty.

```
for line in fileinput.input(): # Sensitive
    print(line)

for line in fileinput.FileInput(): # Sensitive
    print(line)

for line in fileinput.input(['setup.py']): # Ok
    print(line)

for line in fileinput.FileInput(['setup.py']): # Ok
    print(line)
```

See

- [MITRE, CWE-20](#) - Improper Input Validation

Deprecated

This rule is deprecated, and will eventually be removed.

Available In:

sonarcloud  | **sonarqube** 