



Secrets



Apex

C С

CloudFormation

COBOL

C#

3 CSS

 $\bowtie$ Flex

-GO Go

5 HTML

Java

JavaScript JS

Kotlin

Objective C

PHP

PL/I

PL/SQL

**Python** 

RPG

Ruby

Scala

J. Swift

Terraform

Text 月

тѕ 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)

6 Vulnerability (29)



Security Hotspot 31



Tags

Search by name...

Exceptions' "\_\_cause\_\_" should be either an Analyze your code

(see PEP-409).



**Exception or None** 

Exception chaining enables users to see if an exception was triggered by another exception (see PEP-3134). Exceptions are chained using either of the following syntax:

- raise NewException() from chained\_exception
- new\_exception.\_\_cause\_\_ = chained\_exception

It is also possible to erase a chaining by setting new\_exception.\_\_cause\_\_ = None or using except ... from None

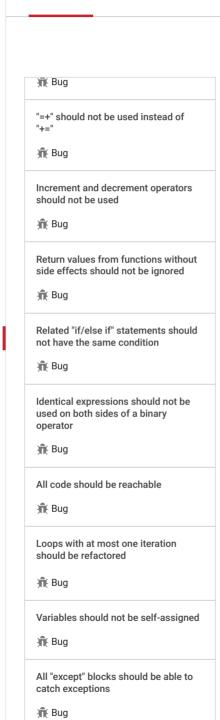
Chaining will fail and raise a TypeError if something else than None or a valid exception, i.e. an instance of BaseException or of a subclass, is provided

## **Noncompliant Code Example**

```
class A:
   pass
   raise ValueError("orig")
except ValueError as e:
   new_exc = TypeError("new")
   new_exc.__cause__ = A() # Noncompliant
    raise new exc
   raise ValueError("orig")
except ValueError as e:
    raise TypeError("new") from "test" # Noncompliant
```

### **Compliant Solution**

```
try:
   raise ValueError("orig")
except ValueError as e:
   new_exc = TypeError("new")
   new_exc.__cause__ = None # Ok
   raise new_exc
try:
   raise ValueError("orig")
except ValueError as e:
   new_exc = TypeError("new")
    new_exc.__cause__ = e \# Ok
   raise new exc
```



Constructing arguments of system

Disabling auto-escaping in template

engines is security-sensitive

commands from user input is

security-sensitive

Security Hotspot

Security Hotspot Setting loose POSIX file permissions is security-sensitive Security Hotspot Formatting SQL queries is securitysensitive

Security Hotspot

Character classes in regular expressions should not contain only one character

Code Smell

Superfluous curly brace quantifiers should be avoided

Code Smell

```
try:
   raise ValueError("orig")
except ValueError as e:
   raise TypeError("new") from None # Ok
   raise ValueError("orig")
except ValueError as e:
   raise TypeError("new") from e # Ok
```

- PEP 3134 Exception Chaining and Embedded Tracebacks
- PEP 409 Suppressing exception context
- PEP 352 Required Superclass for Exceptions
- Python documentation Built-in Exceptions

#### Available In:

sonarlint ⊕ | sonarcloud 👌 | sonarqube

© 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