

- 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...

Wildcard imports should not be used	Code Smell
String literals should not be duplicated	Code Smell
Functions and methods should not be empty	Code Smell
Server-side requests should not be vulnerable to forging attacks	Vulnerability
Non-empty statements should change control flow or have at least one side-effect	Bug
Replacement strings should reference existing regular expression groups	Bug
Alternation in regular expressions should not contain empty alternatives	Bug
Unicode Grapheme Clusters should be avoided inside regex character classes	Bug
Regex alternatives should not be redundant	Bug
Alternatives in regular expressions should be grouped when used with anchors	Bug
New objects should not be created only to check their identity	Bug

Insecure temporary file creation methods should not be used

Analyze your code

Vulnerability

Critical

cwe owasp

Creating temporary files using insecure methods exposes the application to race conditions on filenames: a malicious user can try to create a file with a predictable name before the application does. A successful attack can result in other files being accessed, modified, corrupted or deleted. This risk is even higher if the application run with elevated permissions.

In the past, it has led to the following vulnerabilities:

- [CVE-2014-1858](#)
- [CVE-2014-1932](#)

Noncompliant Code Example

```
import tempfile

filename = tempfile.mktemp() # Noncompliant
tmp_file = open(filename, "w+")
```

Compliant Solution

```
import tempfile

tmp_file1 = tempfile.NamedTemporaryFile(delete=False) # Compliant
tmp_file2 = tempfile.NamedTemporaryFile() # Compliant; Create and delete
```

See





- [OWASP Top 10 2021 Category A1](#) - Broken Access Control
- [OWASP Top 10 2017 Category A9](#) - Using Components with Known Vulnerabilities
- [MITRE, CWE-377](#) - Insecure Temporary File
- [MITRE, CWE-379](#) - Creation of Temporary File in Directory with Incorrect Permissions
- [OWASP, Insecure Temporary File](#)
- [Python tempfile module](#)
- [Python 2.7 os module](#)

Available In:

sonarlint

sonarcloud

sonarqube

 Bug
Collection content should not be replaced unconditionally  Bug
Exceptions should not be created without being raised  Bug
Collection sizes and array length comparisons should make sense  Bug
All branches in a conditional structure should not have exactly the same