sonar RULES

Products

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...

Code Smell

Bare "raise" statements should not be used in "finally" blocks
Code Smell

Arguments given to functions should be of an expected type
Code Smell

`str.replace` should be preferred to `re.sub`
Code Smell

Unread "private" attributes should be removed
Code Smell

Cognitive Complexity of functions should not be too high
Code Smell

The first argument to class methods should follow the naming convention
Code Smell

Method overrides should not change contracts
Code Smell

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

JWT should be signed and verified

Analyze your code

Vulnerability Critical cwe privacy owasp

If a JSON Web Token (JWT) is not signed with a strong cipher algorithm (or not signed at all) an attacker can forge it and impersonate user identities.

- Don't use `none` algorithm to sign or verify the validity of a token.
- Don't use a token without verifying its signature before.

Noncompliant Code Example

For pyjwt module:

```
jwt.decode(token, verify = False)  # Noncompliant
jwt.decode(token, key, options={"verify_signature": Fal
```

For python_jwt module:

```
jwt.process_jwt(token)  # Noncompliant
```

Compliant Solution

For pyjwt module:

```
jwt.decode(token, key, algo)
```

For python_jwt module:

```
jwt.process_jwt(token)  #  Compliant because followed b
jwt.verify_jwt(token, key, algo)
```

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- MITRE, CWE-347 - Improper Verification of Cryptographic Signature

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

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