



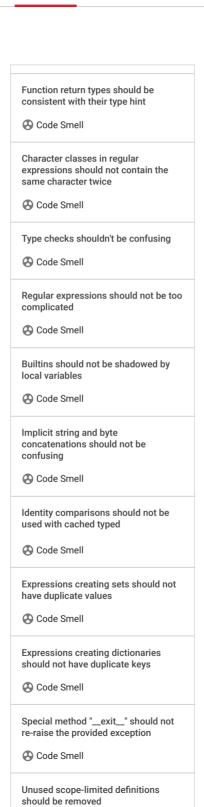
XML



Python static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PYTHON code





Code Smell

Functions and methods should not



Tags

Analyze your code

Search by name...

cwe owasp

Using pseudorandom number generators (PRNGs) is security-sensitive. For example, it has led in the past to the following vulnerabilities:

- CVE-2013-6386
- CVE-2006-3419
- CVE-2008-4102

When software generates predictable values in a context requiring unpredictability, it may be possible for an attacker to guess the next value that will be generated, and use this guess to impersonate another user or access sensitive information.

Ask Yourself Whether

- the code using the generated value requires it to be unpredictable. It is the case for all encryption mechanisms or when a secret value, such as a password, is hashed
- the function you use generates a value which can be predicted (pseudorandom)
- the generated value is used multiple times.
- an attacker can access the generated value.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

- Only use random number generators which are recommended by OWASP or any other trusted organization.
- Use the generated random values only once.
- You should not expose the generated random value. If you have to store it. make sure that the database or file is secure.

Sensitive Code Example

```
import random
random.getrandbits(1) # Sensitive
random.randint(0,9) # Sensitive
random.random() # Sensitive
# the following functions are sadly used to generate salt by
random.sample(['a', 'b'], 1) # Sensitive
random.choice(['a', 'b']) # Sensitive
random.choices(['a', 'b']) # Sensitive
```

- OWASP Top 10 2021 Category A2 Cryptographic Failures
- OWASP Top 10 2017 Category A3 Sensitive Data Exposure
- Mobile AppSec Verification Standard Cryptography Requirements
- OWASP Mobile Top 10 2016 Category M5 Insufficient Cryptography

have identical implementations

Code Smell

Unused private nested classes should be removed

A Code Smell

String formatting should be used correctly

Code Smell

Conditional expressions should not be nested

Code Smell

- <u>MITRE, CWE-338</u> Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
- MITRE, CWE-330 Use of Insufficiently Random Values
- MITRE, CWE-326 Inadequate Encryption Strength
- MITRE, CWE-1241 Use of Predictable Algorithm in Random Number Generator
- Derived from FindSecBugs rule Predictable Pseudo Random Number Generator

Available In:

sonarcloud 쉆 | 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