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...

Security Hotspot

Creating cookies without the "secure" flag is security-sensitive

Security Hotspot

Using hardcoded IP addresses is security-sensitive

Security Hotspot

Regular expression quantifiers and character classes should be used concisely

Code Smell

Character classes should be preferred over reluctant quantifiers in regular expressions

Code Smell

A subclass should not be in the same "except" statement as a parent class

Code Smell

Walrus operator should not make code confusing

Code Smell

Jump statements should not be redundant

Code Smell

"pass" should not be used needlessly

Code Smell

"except" clauses should do more than raise the same issue

Code Smell

Boolean checks should not be inverted

Code Smell

Unused local variables should be removed

Unicode Grapheme Clusters should be avoided inside regex character classes

Analyze your code

Bug Major regex

When placing Unicode Grapheme Clusters (characters which require to be encoded in multiple Code Points) inside a character class of a regular expression, this will likely lead to unintended behavior.

For instance, the grapheme cluster c̈ requires two code points: one for `'c'`, followed by one for the *umlaut* modifier `'\u{0308}'`. If placed within a character class, such as `[c̈]`, the regex will consider the character class being the enumeration `[c\u{0308}]` instead. It will, therefore, match every `'c'` and every *umlaut* that isn't expressed as a single codepoint, which is extremely unlikely to be the intended behavior.

This rule raises an issue every time Unicode Grapheme Clusters are used within a character class of a regular expression.

Noncompliant Code Example

```
re.sub(r"[c̈d̈]", "X", "cc̈d̈d") # Noncompliant, print "XXX
```

Compliant Solution

```
re.sub(r"c̈|d̈", "X", "cc̈d̈d") # print "cXXd"
```

Available In:

sonarlint | sonarcloud | sonarqube

Code Smell

Local variable and function parameter names should comply with a naming convention

Code Smell

Field names should comply with a naming convention

Code Smell

Class names should comply with a naming convention

Code Smell

Method names should comply with a naming convention

Code Smell