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...

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

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 implementation
Bug

The output of functions that don't return anything should not be used
Bug

"=+" should not be used instead of "+="

Weak SSL/TLS protocols should not be used

Analyze your code

Vulnerability | Critical | cwe privacy owasp sans-top25

Older versions of SSL/TLS protocol like "SSLv3" have been proven to be insecure.

This rule raises an issue when an SSL/TLS context is created with an insecure protocol version, i.e. when one of the following constants is detected in the code:

- `OpenSSL.SSL.SSLv3_METHOD` (Use instead `OpenSSL.SSL.TLSv1_2_METHOD`)
- `ssl.PROTOCOL_SSLv3` (Use instead `ssl.PROTOCOL_TLSv1_2`)

Protocol versions different from TLSv1.2 and TLSv1.3 are considered insecure.

Noncompliant Code Example

```
from OpenSSL import SSL

SSL.Context(SSL.SSLv3_METHOD)  # Noncompliant
```

```
import ssl

ssl.SSLContext(ssl.PROTOCOL_SSLv3) # Noncompliant
```

Compliant Solution

```
from OpenSSL import SSL

SSL.Context(SSL.TLSv1_2_METHOD)  # Compliant
```

```
import ssl

ssl.SSLContext(ssl.PROTOCOL_TLSv1_2) # Compliant
```

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- OWASP Top 10 2021 Category A7 - Identification and Authentication Failures
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- Mobile AppSec Verification Standard - Network Communication Requirements

Bug

Increment and decrement operators should not be used

Bug

Return values from functions without side effects should not be ignored

Bug

Related "if/else if" statements should not have the same condition

Bug

Identical expressions should not be used on both sides of a binary operator

- OWASP Mobile Top 10 2016 Category M3 - Insecure Communication
- MITRE, CWE-327 - Inadequate Encryption Strength
- MITRE, CWE-326 - Use of a Broken or Risky Cryptographic Algorithm
- SANS Top 25 - Porous Defenses
- SSL and TLS Deployment Best Practices - Use secure protocols

Available In:

sonarlint | sonarcloud | sonarqube