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...

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

Collection content should not be replaced unconditionally
Bug

Exceptions should not be created without being raised

LDAP connections should be authenticated

Analyze your code

Vulnerability | Critical | cwe owasp

An LDAP client authenticates to an LDAP server with a "bind request" which provides, among other, a simple authentication method.

Simple authentication in LDAP can be used with three different mechanisms:

- *Anonymous Authentication Mechanism* by performing a bind request with a username and password value of zero length.
- *Unauthenticated Authentication Mechanism* by performing a bind request with a password value of zero length.
- *Name/Password Authentication Mechanism* by performing a bind request with a password value of non-zero length.

Anonymous binds and unauthenticated binds allow access to information in the LDAP directory without providing a password, their use is therefore strongly discouraged.

Noncompliant Code Example

```
import ldap

def init_ldap():
   connect = ldap.initialize('ldap://example:1389')

   connect.simple_bind('cn=root') # Noncompliant
   connect.simple_bind_s('cn=root') # Noncompliant
   connect.bind_s('cn=root', None) # Noncompliant
   connect.bind('cn=root', None) # Noncompliant
```

Compliant Solution

```
import ldap
import os

def init_ldap():
   connect = ldap.initialize('ldap://example:1389')

   connect.simple_bind('cn=root', os.environ.get('LDAP_
   connect.simple_bind_s('cn=root', os.environ.get('LDA
   connect.bind_s('cn=root', os.environ.get('LDAP_PASSW
   connect.bind('cn=root', os.environ.get('LDAP_PASSWOR
```

See

- OWASP Top 10 2021 Category A7 - Identification and Authentication Failures
- OWASP Top 10 2017 Category A2 - Broken Authentication
- MITRE, CWE-521 - Weak Password Requirements

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 "+="

- ldapwiki.com- Simple Authentication

Available In:

sonarlint | sonarcloud | sonarqube