Python static code analysis

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

Functions should not have too many lines of code
Code Smell

Track uses of "NOSONAR" comments
Code Smell

Track comments matching a regular expression
Code Smell

Statements should be on separate lines
Code Smell

Functions should not contain too many return statements
Code Smell

Files should not have too many lines of code
Code Smell

Lines should not be too long
Code Smell

Methods and properties that don't access instance data should be static
Code Smell

New-style classes should be used
Code Smell

Parentheses should not be used after certain keywords
Code Smell

Track "TODO" and "FIXME" comments that do not contain a reference to a person
Code Smell

Module names should comply with a naming convention

Sending emails is security-sensitive

Analyze your code

Security Hotspot Critical

Sending emails is security-sensitive and can expose an application to a large range of vulnerabilities.

Information Exposure

Emails often contain sensitive information which might be exposed to an attacker if he can add an arbitrary address to the recipient list.

Spamming / Phishing

Malicious user can abuse email based feature to send spam or phishing content.

Dangerous Content Injection

Emails can contain HTML and JavaScript code, thus they can be used for XSS attacks.

Email Headers Injection

Email fields such as `subject`, `to`, `cc`, `bcc`, `from` are set in email "headers". Using unvalidated user input to set those fields might allow attackers to inject new line characters in headers to craft malformed SMTP requests. Although modern libraries are filtering new line character by default, user data used in email "headers" should always be validated.

In the past, it has led to the following vulnerabilities:

- CVE-2017-9801
- CVE-2016-4803

Ask Yourself Whether

- Unvalidated user input are used to set email headers.
- Email content contains data provided by users and it is not sanitized.
- Email recipient list or body are based on user inputs.

You are at risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

- Use an email library which sanitizes headers (Flask-Mail or django.core.mail).
- Use html escape functions to sanitize every piece of data used to in the email body.
- Verify application logic to make sure that email base feature can not be abuse to:
  - Send arbitrary email for spamming or fishing
  - Disclose sensitive email content

Sensitive Code Example

smtplib

Code Smell

Comments should not be located at the end of lines of code

Code Smell

Lines should not end with trailing whitespaces

Code Smell

Files should contain an empty newline at the end

Code Smell

Long suffix "L" should be upper case

Code Smell

```
import smtplib

def send(from_email, to_email, msg):
  server = smtplib.SMTP('localhost', 1025)
  server.sendmail(from_email, to_email, msg) # Sensitiv
```

Django

```
from django.core.mail import send_mail

def send(subject, msg, from_email, to_email):
  send_mail(subject, msg, from_email, [to_email]) # Sen
```

Flask-Mail

```
from flask import Flask
from flask_mail import Mail, Message

app = Flask(__name__)

def send(subject, msg, from_email, to_email):
    mail = Mail(app)
    msg = Message(subject, [to_email], body, sender=fro
    mail.send(msg) # Sensitive{code}
```

See

- Email Injection
- OWASP Top 10 2017 Category A1 - Injection
- MITRE, CWE-93 - Improper Neutralization of CRLF Sequences ('CRLF Injection')
- MITRE, CWE-80 - Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)
- SANS Top 25 - Insecure Interaction Between Components

Deprecated

This rule is deprecated, and will eventually be removed.

Available In:

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