

- 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 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

HTML autoescape mechanism should not be globally disabled

Analyze your code

Vulnerability Blocker

Template engines have an HTML autoescape mechanism that protects web applications against most common cross-site-scripting (XSS) vulnerabilities.

By default, it automatically replaces HTML special characters in any template variables. This secure by design configuration should not be globally disabled.

Escaping HTML from template variables prevents switching into any execution context, like `<script>`. Disabling autoescaping forces developers to manually escape each template variable for the application to be safe. A more pragmatic approach is to escape by default and to manually disable escaping when needed.

A successful exploitation of a cross-site-scripting vulnerability by an attacker allow him to execute malicious JavaScript code in a user's web browser. The most severe XSS attacks involve:

- Forced redirection
- Modify presentation of content
- User accounts takeover after disclosure of sensitive information like session cookies or passwords

This rule supports the following libraries:

- [Django Templates](#)
- [Jinja2](#)

Noncompliant Code Example

```
from jinja2 import Environment






env = Environment() # Noncompliant; New Jinja2 Environm
env = Environment(autoescape=False) # Noncompliant
```

Compliant Solution

```
from jinja2 import Environment
env = Environment(autoescape=True) # Compliant
```

See

- [OWASP Cheat Sheet](#) - XSS Prevention Cheat Sheet
- [OWASP Top 10 2017 Category A7](#) - Cross-Site Scripting (XSS)
- [OWASP Top 10 2017 Category A6](#) - Security Misconfiguration
- [MITRE, CWE-79](#) - Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- [MITRE, CWE-80](#) - Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)

 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

- [MITRE, CWE-81](#) - Improper Neutralization of Script in an Error Message Web Page
- [MITRE, CWE-82](#) - Improper Neutralization of Script in Attributes of IMG Tags in a Web Page
- [MITRE, CWE-83](#) - Improper Neutralization of Script in Attributes in a Web Page
- [MITRE, CWE-84](#) - Improper Neutralization of Encoded URI Schemes in a Web Page
- [MITRE, CWE-85](#) - Doubled Character XSS Manipulations
- [MITRE, CWE-86](#) - Improper Neutralization of Invalid Characters in Identifiers in Web Pages
- [MITRE, CWE-87](#) - Improper Neutralization of Alternate XSS Syntax
- [SANS Top 25](#) - Insecure Interaction Between Components

Deprecated

This rule is deprecated; use {rule:python:S5247} instead.

Available In:

sonarlint  | sonarcloud  | sonarqube 