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

Disabling server-side encryption of S3 buckets is security-sensitive

Analyze your code

Security Hotspot Minor ? aws cwe owasp

Server-side encryption (SSE) encrypts an object (not the metadata) as it is written to disk (where the S3 bucket resides) and decrypts it as it is read from disk. This doesn’t change the way the objects are accessed, as long as the user has the necessary permissions, objects are retrieved as if they were unencrypted. Thus, SSE only helps in the event of disk thefts, improper disposals of disks and other attacks on the AWS infrastructure itself.

There are three SSE options:

- Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)
  - AWS manages encryption keys and the encryption itself (with AES-256) on its own.
- Server-Side Encryption with Customer Master Keys (CMKs) Stored in AWS Key Management Service (SSE-KMS)
  - AWS manages the encryption (AES-256) of objects and encryption keys provided by the AWS KMS service.
- Server-Side Encryption with Customer-Provided Keys (SSE-C)
  - AWS manages only the encryption (AES-256) of objects with encryption keys provided by the customer. AWS doesn’t store the customer’s encryption keys.

Ask Yourself Whether

- The S3 bucket stores sensitive information.
- The infrastructure needs to comply to some regulations, like HIPAA or PCI DSS, and other standards.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

It’s recommended to use SSE. Choosing the appropriate option depends on the level of control required for the management of encryption keys.

Sensitive Code Example

Server-side encryption is not used:

```
bucket = s3.Bucket(self,"bucket",
    encryption=s3.BucketEncryption.UNENCRYPTED       #
)
```

The default value of `encryption` is `KMS` if `encryptionKey` is set. Otherwise, if both parameters are absent the bucket is unencrypted.

Compliant Solution

Server-side encryption with Amazon S3-Managed Keys is used:

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

```
bucket = s3.Bucket(self,"bucket",
    encryption=s3.BucketEncryption.S3_MANAGED
)

# Alternatively with a KMS key managed by the user.

bucket = s3.Bucket(self,"bucket",
    encryptionKey=access_key
)
```

See

- OWASP Top 10 2021 Category A4 - Insecure Design
- OWASP Top 10 2021 Category A5 - Security Misconfiguration
- MITRE, CWE-311 - Missing Encryption of Sensitive Data
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- AWS documentation - Protecting data using server-side encryption
- AWS CDK version 2 - BucketEncryption

Available In:

sonarcloud sonarqube