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...

Variables should not be self-assigned
Bug

All "except" blocks should be able to catch exceptions
Bug

Constructing arguments of system commands from user input is security-sensitive
Security Hotspot

Disabling auto-escaping in template engines is security-sensitive
Security Hotspot

Setting loose POSIX file permissions is security-sensitive
Security Hotspot

Formatting SQL queries is security-sensitive
Security Hotspot

Character classes in regular expressions should not contain only one character
Code Smell

Superfluous curly brace quantifiers should be avoided
Code Smell

Non-capturing groups without quantifier should not be used
Code Smell

Regular expressions should not contain empty groups
Code Smell

Regular expressions should not contain multiple spaces
Code Smell

Allowing public ACLs or policies on a S3 bucket is security-sensitive

Analyze your code

Security Hotspot | Critical | aws cwe owasp

By default S3 buckets are private, it means that only the bucket owner can access it.

This access control can be relaxed with ACLs or policies.

To prevent permissive policies to be set on a S3 bucket the following booleans settings can be enabled:

- `block_public_acls`: to block or not public ACLs to be set to the S3 bucket.
- `ignore_public_acls`: to consider or not existing public ACLs set to the S3 bucket.
- `block_public_policy`: to block or not public policies to be set to the S3 bucket.
- `restrict_public_buckets`: to restrict or not the access to the S3 endpoints of public policies to the principals within the bucket owner account.

The other attribute `BlockPublicAccess.BLOCK_ACLS` only turns on `block_public_acls` and `ignore_public_acls`. The public policies can still affect the S3 bucket.

However, all of those options can be enabled by setting the `block_public_access` property of the S3 bucket to `BlockPublicAccess.BLOCK_ALL`.

Ask Yourself Whether

- The S3 bucket stores sensitive data.
- The S3 bucket is not used to store static resources of websites (images, css ...).
- Many users have the permission to set ACL or policy to the S3 bucket.
- These settings are not already enforced to true at the account level.

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

Recommended Secure Coding Practices

It's recommended to configure:

- `block_public_acls` to `True` to block new attempts to set public ACLs.
- `ignore_public_acls` to `True` to block existing public ACLs.
- `block_public_policy` to `True` to block new attempts to set public policies.
- `restrict_public_buckets` to `True` to restrict existing public policies.

Sensitive Code Example

Code Smell

Single-character alternations in regular expressions should be replaced with character classes

Code Smell

Reluctant quantifiers in regular expressions should be followed by an expression that can't match the empty string

Code Smell

Values assigned to variables should match their type annotations

Code Smell

Function return types should be consistent with their type hint

By default, when not set, the `block_public_access` is fully deactivated (nothing is blocked):

```
bucket = s3.Bucket(self,
    "bucket"        # Sensitive
)
```

This `block_public_access` allows public ACL to be set:

```
bucket = s3.Bucket(self,
    "bucket",
    block_public_access=s3.BlockPublicAccess(
        block_public_acls=False,       # Sensitive
        ignore_public_acls=True,
        block_public_policy=True,
        restrict_public_buckets=True
    )
)
```

The attribute `BLOCK_ACLS` only blocks and ignores public ACLs:

```
bucket = s3.Bucket(self,
    "bucket",
    block_public_access=s3.BlockPublicAccess.BLOCK_ACLS
)
```

Compliant Solution

This `block_public_access` blocks public ACLs and policies, ignores existing public ACLs and restricts existing public policies:

```
bucket = s3.Bucket(self,
    "bucket",
    block_public_access=s3.BlockPublicAccess.BLOCK_ALL
)
```

A similar configuration to the one above can obtained by setting all parameters of the `block_public_access`

```
bucket = s3.Bucket(self, "bucket",
    block_public_access=s3.BlockPublicAccess(       # C
        block_public_acls=True,
        ignore_public_acls=True,
        block_public_policy=True,
        restrict_public_buckets=True
    )
)
```

See

- OWASP Top 10 2021 Category A1 - Broken Access Control
- OWASP Top 10 2021 Category A5 - Security Misconfiguration
- AWS Documentation - Blocking public access to your Amazon S3 storage
- MITRE, CWE-284 - Improper Access Control
- OWASP Top 10 2017 Category A5 - Broken Access Control
- AWS CDK version 2 - Bucket

Available In:

sonarcloud | sonarqube