Secrets
ABAP
Apex
C
C++
CloudFormation
COBOL
C#
CSS
Flex
Go
HTML
Java
JavaScript
Kotlin
Kubernetes
Objective C
PHP
PL/I
PL/SQL
Python
RPG
Ruby
Scala
Swift
Terraform
Text
TypeScript
T-SQL
VB.NET
VB6
XML

CloudFormation static code analysis

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

All rules 27 | Vulnerability 3 | Security Hotspot 20 | Code Smell 4

Tags | Search by name...

Granting access to S3 buckets to all or authenticated users is security-sensitive
Security Hotspot

AWS IAM policies should not allow privilege escalation
Vulnerability

Weak SSL/TLS protocols should not be used
Vulnerability

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

Authorizing HTTP communications with S3 buckets is security-sensitive
Security Hotspot

Using clear-text protocols is security-sensitive
Security Hotspot

"Log Groups" should be configured with a retention policy
Code Smell

Defining a short backup retention duration is security-sensitive
Security Hotspot

Using unencrypted EFS file systems is security-sensitive
Security Hotspot

Using unencrypted SQS queues is security-sensitive
Security Hotspot

Using unencrypted SNS topics is security-sensitive
Security Hotspot

Using unencrypted SageMaker notebook instances is security-sensitive
Security Hotspot

Using unencrypted Elasticsearch domains is security-sensitive

Granting access to S3 buckets to all or authenticated users is security-sensitive

Analyze your code

Security Hotspot | Blocker | aws cwe owasp

Predefined permissions, also known as canned ACLs, are an easy way to grant large privileges to predefined groups or users.

The following canned ACLs are security-sensitive:

- `PublicRead`, `PublicReadWrite` grant respectively "read" and "read and write" privileges to everyone in the world (`AllUsers` group).
- `AuthenticatedRead` grants "read" privilege to all authenticated users (`AuthenticatedUsers` group).

Ask Yourself Whether

- The S3 bucket stores sensitive data.
- The S3 bucket is not used to store static resources of websites (images, css ...).

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

Recommended Secure Coding Practices

It's recommended to implement the least privilege policy, ie to grant necessary permissions only to users for their required tasks. In the context of canned ACL, set it to `private` (the default one) and if needed more granularity then use an appropriate S3 policy.

Sensitive Code Example

All users (ie: anyone in the world authenticated or not) have read and write permissions with the `PublicReadWrite` access control:

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  S3Bucket:
    Type: 'AWS::S3::Bucket' # Sensitive
    Properties:
      BucketName: "mynoncompliantbucket"
      AccessControl: "PublicReadWrite"
```

Compliant Solution

With the `private` access control (default), only the bucket owner has the read/write permissions on the buckets and its ACL.

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  S3Bucket:
    Type: 'AWS::S3::Bucket' # Compliant
    Properties:
      BucketName: "mycompliantbucket"
      AccessControl: "Private"
```

See

- OWASP Top 10 2021 Category A1 - Broken Access Control
- AWS Documentation - Access control list (ACL) overview (canned ACLs)
- AWS Documentation - Controlling access to a bucket with user policies
- MITRE, CWE-732 - Incorrect Permission Assignment for Critical Resource
- MITRE, CWE-284 - Improper Access Control
- OWASP Top 10 2017 Category A5 - Broken Access Control

Available In:

sonarcloud | sonarqube