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...

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
Security Hotspot

Using unencrypted RDS databases is security-sensitive
Security Hotspot

Using unencrypted EBS volumes is security-sensitive
Security Hotspot

Disabling logging is security-sensitive
Security Hotspot

"Log Groups" should be declared explicitly
Code Smell

Administration services access should be restricted to specific IP addresses
Vulnerability

Disabling versioning of S3 buckets is security-sensitive
Security Hotspot

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

AWS tag keys should comply with a naming convention
Code Smell

CloudFormation parsing failure
Code Smell

Using unencrypted SageMaker notebook instances is security-sensitive

Analyze your code

Security Hotspot | Major | aws cwe owasp

Amazon SageMaker is a managed machine learning service in a hosted production-ready environment. To train machine learning models, SageMaker instances can process potentially sensitive data, such as personal information that should not be stored unencrypted. In the event that adversaries physically access the storage media, they cannot decrypt encrypted data.

Ask Yourself Whether

- The instance contains sensitive data that could cause harm when leaked.
- There are compliance requirements for the service to store data encrypted.

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

Recommended Secure Coding Practices

It's recommended to encrypt SageMaker notebook instances that contain sensitive information. Encryption and decryption are handled transparently by SageMaker, so no further modifications to the application are necessary.

Sensitive Code Example

For AWS::SageMaker::NotebookInstance:

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  Notebook:  # Sensitive, encryption disabled by default
    Type: AWS::SageMaker::NotebookInstance
```

Compliant Solution

For AWS::SageMaker::NotebookInstance:

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  Notebook:
    Type: AWS::SageMaker::NotebookInstance
    Properties:
      KmsKeyId:
        Fn::GetAtt:
          - SomeKey
          - KeyId
```

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- OWASP Top 10 2021 Category A4 - Insecure Design
- OWASP Top 10 2021 Category A5 - Security Misconfiguration
- Protect Data at Rest Using Encryption
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- MITRE, CWE-311 - Missing Encryption of Sensitive Data

Available In:

sonarcloud | sonarqube