




 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





Terraform static code analysis


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


All rules 50


 Vulnerability 5

 Security Hotspot 43


 Code Smell 2


Tags 

Search by name... 


 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




 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




 Vulnerability

Administration services access should be restricted to specific IP addresses


 Security Hotspot



Unversioned Google Cloud Storage buckets are security-sensitive


 Security Hotspot

Disabling S3 bucket MFA delete is security-sensitive

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

 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:

```
resource "aws_s3_bucket" "mynoncompliantbucket" {
  bucket = "mynoncompliantbucket"
}
```

Compliant Solution

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

```
resource "aws_s3_bucket" "mycompliantbucket" {
  bucket = "mycompliantbucket"
}
```

https://rules.sonarsource.com/terraform/RSPEC-6245

1/2

Security Hotspot
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
Terraform parsing failure
Code Smell

```
server_side_encryption_configuration {
  rule {
    apply_server_side_encryption_by_default {
      sse_algorithm = "AES256"
    }
  }
}
```

See

- [OWASP Top 10 2021 Category A4](#) - Insecure Design
- [OWASP Top 10 2021 Category A5](#) - Security Misconfiguration
- [AWS documentation](#) - Protecting data using server-side encryption
- [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

Available In:

