




 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
	Using unencrypted SQS queues is security-sensitive
	Using unencrypted SNS topics is security-sensitive
	Using unencrypted SageMaker notebook instances is security-sensitive
	Using unencrypted Elasticsearch domains is security-sensitive
	Using unencrypted RDS databases is security-sensitive
	Using unencrypted EBS volumes is security-sensitive
	Disabling logging is security-sensitive
	Administration services access should be restricted to specific IP addresses
	Unversioned Google Cloud Storage buckets are security-sensitive
	Disabling S3 bucket MFA delete is security-sensitive

AWS tag keys should comply with a naming convention

Analyze your code

 Code Smell

 Minor ?

 aws convention

Shared conventions allow teams to collaborate effectively. This rule allows to check that all tag keys match a provided regular expression.

Noncompliant Code Example

With default provided regular expression `^([A-Z])([A-Z][A-Za-z]*)$`:

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

  tags = {
    "anycompany:cost-center" = "Accounting" # Noncompliant
  }
}
```

Compliant Solution

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

  tags = {
    "AnyCompany:CostCenter" = "Accounting"
  }
}
```

See

- [AWS Documentation](#): Adopt a Standardized Approach for Tag Names

Available In:

sonarcloud  **sonarqube** 

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved. [Privacy Policy](#)

<div><div></div><div>Security Hotspot</div></div>
<div><div><div>Disabling versioning of S3 buckets is security-sensitive</div><div><div></div><div>Security Hotspot</div></div></div></div>
<div><div><div>Disabling server-side encryption of S3 buckets is security-sensitive</div><div><div></div><div>Security Hotspot</div></div></div></div>
<div><div><div>AWS tag keys should comply with a naming convention</div><div><div></div><div>Code Smell</div></div></div></div>
<div><div><div>Terraform parsing failure</div><div><div></div><div>Code Smell</div></div></div></div>