




 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



### Granting highly privileged GCP resource rights is security-sensitive

 Security Hotspot  Major  azure cwe-284

Granting highly privileged resource rights to users or groups can reduce an organization's ability to protect against account or service theft. It prevents proper segregation of duties and creates potentially critical attack vectors on affected resources.

If elevated access rights are abused or compromised, both the data that the affected resources work with and their access tracking are at risk.

#### Ask Yourself Whether

- This GCP resource is essential to the information system infrastructure.
- This GCP resource is essential to mission-critical functions.
- Compliance policies require that administrative privileges for this resource be limited to a small group of individuals.

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

#### Recommended Secure Coding Practices

Grant IAM policies or members a less permissive role: In most cases, granting them read-only privileges is sufficient.

Separate tasks by creating multiple roles that do not use a full access role for day-to-day work.

If the predefined GCP roles do not include the specific permissions you need, create [custom IAM roles](#).

#### Sensitive Code Example

For an IAM policy setup:






```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/run.admin" # Sensitive
    members = [
      "user:name@example.com",
    ]
  }
}

resource "google_cloud_run_service_iam_policy" "policy"
  location = google_cloud_run_service.default.location
  project = google_cloud_run_service.default.project
  service = google_cloud_run_service.default.name
  policy_data = data.google_iam_policy.admin.policy_data
}
```

For an IAM policy binding:

https://rules.sonarsource.com/terraform/RSPEC-6400

1/3

 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

```
resource "google_cloud_run_service_iam_binding" "example" {
  location = google_cloud_run_service.default.location
  project  = google_cloud_run_service.default.project
  service  = google_cloud_run_service.default.name
  role     = "roles/run.admin" # Sensitive
  members = [
    "user:name@example.com",
  ]
}
```

For adding a member to a policy:

```
resource "google_cloud_run_service_iam_member" "example" {
  location = google_cloud_run_service.default.location
  project  = google_cloud_run_service.default.project
  service  = google_cloud_run_service.default.name
  role     = "roles/run.admin" # Sensitive
  member   = "user:name@example.com"
}
```

Compliant Solution

For an IAM policy setup:

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:name@example.com",
    ]
  }
}

resource "google_cloud_run_service_iam_policy" "example" {
  location = google_cloud_run_service.default.location
  project  = google_cloud_run_service.default.project
  service  = google_cloud_run_service.default.name
  policy_data = data.google_iam_policy.admin.policy_data
}
```

For an IAM policy binding:

```
resource "google_cloud_run_service_iam_binding" "example" {
  location = google_cloud_run_service.default.location
  project  = google_cloud_run_service.default.project
  service  = google_cloud_run_service.default.name
  role     = "roles/viewer"
  members = [
    "user:name@example.com",
  ]
}
```

For adding a member to a policy:

```
resource "google_cloud_run_service_iam_member" "example" {
  location = google_cloud_run_service.default.location
  project  = google_cloud_run_service.default.project
  service  = google_cloud_run_service.default.name
  role     = "roles/viewer"
  member   = "user:name@example.com"
}
```

See

- OWASP Top 10 2021 Category A1 - Broken Access Control
- OWASP Top 10 2017 Category A5 - Broken Access Control
- MITRE, CWE-284 - Improper Access Control

Available In:



---

© 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](#)