




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... 🔍

Assigning high privileges Azure Active Directory built-in roles is security-sensitive

Security Hotspot

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

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

Administration services access should be restricted to specific IP addresses

Vulnerability

Excessive granting of GCP IAM permissions is security-sensitive

Analyze your code

Security Hotspot

Major ?

gcp cwe-284

Excessive granting of GCP IAM permissions can allow attackers to exploit an organization's cloud resources with malicious intent.

To prevent improper creation or deletion of resources after an account is compromised, proactive measures include both following GCP Security Insights and ensuring custom roles contain as few privileges as possible.

After gaining a foothold in the target infrastructure, sophisticated attacks typically consist of two major parts.

First, attackers must deploy new resources to carry out their malicious intent. To guard against this, operations teams must control what unexpectedly appears in the infrastructure, such as what is:

- added
- written down
- updated
- started
- appended
- applied
- accessed.

Once the malicious intent is executed, attackers must avoid detection at all costs.

To counter attackers' attempts to remove their fingerprints, operations teams must control what unexpectedly disappears from the infrastructure, such as what is:

- stopped
- disabled
- canceled
- deleted
- destroyed
- detached
- disconnected
- suspended
- rejected
- removed.

For operations teams to be resilient in this scenario, their organization must apply both:

- Detection security: log these actions to better detect malicious actions.
- Preventive security: review and limit granted permissions.





This rule raises an issue when a custom role grants a number of sensitive permissions (read-write or destructive permission) that is greater than a given parameter.

Ask Yourself Whether

- This custom role will be mostly used for read-only purposes.
- Compliance policies require read-only access.

https://rules.sonarsource.com/terraform/RSPEC-6406

1/2

Unversioned Google Cloud Storage buckets are security-sensitive
 Security Hotspot
Disabling S3 bucket MFA delete is security-sensitive
 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

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

Recommended Secure Coding Practices

To reduce the risks associated with this role after a compromise:

- Reduce the list of permissions to grant only those that are actually needed.
- Favor read-only over read-write.

Sensitive Code Example

This custom role grants more than 5 sensitive permissions:

```
resource "google_project_iam_custom_role" "example" {
  permissions = [ # Sensitive
    "resourcemanager.projects.create", # Sensitive perm
    "resourcemanager.projects.delete", # Sensitive perm
    "resourcemanager.projects.get",
    "resourcemanager.projects.list",
    "run.services.create", # Sensitive permission
    "run.services.delete", # Sensitive permission
    "run.services.get",
    "run.services.getIamPolicy",
    "run.services.setIamPolicy", # Sensitive permisso
    "run.services.list",
    "run.services.update", # Sensitive permission
  ]
}
```

Compliant Solution

This custom role grants less than 5 sensitive permissions:

```
resource "google_project_iam_custom_role" "example" {
  permissions = [
    "resourcemanager.projects.get",
    "resourcemanager.projects.list",
    "run.services.create",
    "run.services.delete",
    "run.services.get",
    "run.services.getIamPolicy",
    "run.services.list",
    "run.services.update",
  ]
}
```

See

- [GCP Docs](#) - Enforce least privilege with role recommendations
- [GCP Docs](#) - Security Insights
- [OWASP Top 10 2021 Category A1](#) - Broken Access Control
- [OWASP Top 10 2017 Category A5](#) - Broken Access Control
- [MITRE, CWE-668](#) - Exposure of Resource to Wrong Sphere

Available In:

