




 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


 Security Hotspot


Using unencrypted SQS queues is security-sensitive

 Security Hotspot


 Security Hotspot


Using unencrypted SNS topics is security-sensitive

 Security Hotspot


 Security Hotspot


Using unencrypted SageMaker notebook instances is security-sensitive

 Security Hotspot


 Security Hotspot


Using unencrypted Elasticsearch domains is security-sensitive

 Security Hotspot


 Security Hotspot


Using unencrypted RDS databases is security-sensitive

 Security Hotspot


 Security Hotspot


Using unencrypted EBS volumes is security-sensitive

 Security Hotspot


 Security Hotspot

Disabling logging is security-sensitive


 Security Hotspot

 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

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

 Security Hotspot  Major  azure

Azure Active Directory offers built-in roles that can be assigned to users, groups, or service principals. Some of these roles should be carefully assigned as they grant sensitive permissions like the ability to reset passwords for all users.

An Azure account that fails to limit the use of such roles has a higher risk of being breached by a compromised owner.

This rule raises an issue when one of the following roles is assigned:

- Application Administrator
- Authentication Administrator
- Cloud Application Administrator
- Global Administrator
- Groups Administrator
- Helpdesk Administrator
- Password Administrator
- Privileged Authentication Administrator
- Privileged Role Administrator
- User Administrator

Ask Yourself Whether

- The user, group, or service principal doesn't use the entirety of this extensive set of permissions to operate on a day-to-day basis.
- It is possible to follow the Separation of Duties principle and split permissions between multiple users, but it's not enforced.

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

Recommended Secure Coding Practices

- Limit the assignment of Global Administrator roles to less than five people or service principals.
- Apply the least privilege principle by choosing a role with a limited set of permissions.
- If no built-in role meets your needs, create a custom role with as few permissions as possible.






Sensitive Code Example

```
resource "azuread_directory_role" "example" {
  display_name = "Privileged Role Administrator" # Sensitive
}

resource "azuread_directory_role_member" "example" {
  role_object_id = azuread_directory_role.example.object_id
  member_object_id = data.azuread_user.example.object_id
}
```

https://rules.sonarsource.com/terraform/RSPEC-6375

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

Compliant Solution

```
resource "azuread_directory_role" "example" {
  display_name = "Usage Summary Reports Reader"
}

resource "azuread_directory_role_member" "example" {
  role_object_id   = azuread_directory_role.example.object_id
  member_object_id = data.azuread_user.example.object_id
}
```

See

- [OWASP Top 10 2021 Category A1](#) - Broken Access Control
- [OWASP Top 10 2021 Category A4](#) - Insecure Design
- [OWASP Top 10 2017 Category A5](#) - Broken Access Control
- [MITRE, CWE-266](#) - Incorrect Privilege Assignment
- [Azure AD Documentation](#) - Azure AD built-in roles
- [Azure AD Documentation](#) - Best practices for Azure AD roles

Available In:

