




 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	
	Security Hotspot
Administration services access should be restricted to specific IP addresses	
	Vulnerability
Unversioned Google Cloud Storage buckets are security-sensitive	
	Security Hotspot
Disabling S3 bucket MFA delete is security-sensitive	
	Security Hotspot

Enabling project-wide SSH keys to access VM instances is security-sensitive

Analyze your code

 Security Hotspot

 Major ?

 gcp

SSH keys stored and managed in a project's metadata can be used to access GCP VM instances. By default, GCP automatically deploys project-level SSH keys to VM instances.

Project-level SSH keys can lead to unauthorized access because:

- Their use prevents fine-grained VM-level access control and makes it difficult to follow [the principle of least privilege](#).
- Unlike managed access control with [OS Login](#), manual cryptographic key management is error-prone and requires careful attention. For example, if a user leaves a project, their SSH keys should be removed from the metadata to prevent unwanted access.
- If a project-level SSH key is compromised, all VM instances may be compromised.

- Ask Yourself Whether
- VM instances in a project have different security requirements.
 - Many users with different profiles need access to the VM instances in that project.

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

- Recommended Secure Coding Practices
- Block project-level SSH keys by setting the `metadata.block-project-ssh-keys` argument to `true`
 - Use [OSLogin](#) to benefit from managed access control.

Sensitive Code Example

```
resource "google_compute_instance" "example" { # Sensitive,
  name           = "example"
  machine_type   = "e2-micro"
  zone           = "us-central1-a"


  network_interface {
    network = "default"

    access_config {
    }
  }
}
```


Compliant Solution

```
resource "google_compute_instance" "example" {
  name           = "example"
  machine_type   = "e2-micro"
  zone           = "us-central1-a"
}
```


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

```
metadata = {
  block-project-ssh-keys = true
}

network_interface {
  network = "default"

  access_config {
  }
}
```

See

- [OWASP Top 10 2021 Category A7](#) - Identification and Authentication Failures
- [OWASP Top 10 2017 Category A2](#) - Broken Authentication
- [MITRE, CWE-266](#) - Incorrect Privilege Assignment
- [MITRE, CWE-269](#) - Improper Privilege Management
- [MITRE, CWE-272](#) - Least Privilege Violation
- [GCP Documentation](#) - Restrict SSH keys from VMs
- [GCP Documentation](#) - Risks of manual key management

Available In:

