**Lab Exercise 18- Scanning IaC Templates for Vulnerabilities**

**Objective**

* Learn how to scan Infrastructure as Code (IaC) templates for security vulnerabilities.
* Use open-source IaC security tools to detect misconfigurations.
* Understand common risks such as public access, unencrypted resources, and insecure network rules.

**Prerequisites**

* A Linux/Windows/Mac machine with:
  + Terraform installed (for sample IaC)
  + **Checkov** (pip install checkov) or **tfsec** (brew install tfsec or binary download)
* Git installed (optional, for version control of IaC templates)

**Step 1: Create an Insecure IaC Template**

Create a file named main.tf with the following Terraform code:

provider "aws" {

region = "us-east-1"

}

resource "aws\_s3\_bucket" "insecure\_bucket" {

bucket = "my-insecure-bucket-lab"

acl = "public-read"

}

resource "aws\_security\_group" "insecure\_sg" {

name = "insecure-sg"

description = "Allow all inbound traffic"

ingress {

from\_port = 0

to\_port = 65535

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

}

**Step 2: Scan the Template with Checkov**

Run Checkov on the current directory:

checkov -d .

**Expected Findings:**

* Public S3 bucket access (public-read)
* Security group open to all inbound traffic

**Expected Findings:**

* Warns about S3 bucket without encryption
* Flags open Security Group rules

**Step 4: Review the Report**

Example output (Checkov):

Check: CKV\_AWS\_20: "S3 Bucket allows public read access"

FAILED for resource: aws\_s3\_bucket.insecure\_bucket

Check: CKV\_AWS\_260: "Security group allows ingress from 0.0.0.0/0"

FAILED for resource: aws\_security\_group.insecure\_sg

**Step 5: Apply Fixes (Optional)**

Modify the IaC template to:

* Set S3 bucket ACL to private
* Enable encryption (AES256)
* Restrict Security Group to specific IP ranges

**Step 6: Rescan the Template**

Run the scan again:

checkov -d .

Now the findings should be **resolved or reduced**.

**Step 7: Document Findings**

Create a simple findings log: