

# 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)

```
PS C:\Users\ASUS> pip install checkov
Collecting checkov
  Downloading checkov-3.2.471-py3-none-any.whl.metadata (26 kB)
Collecting bc-python-hcl2==0.4.3 (from checkov)
  Downloading bc_python_hcl2-0.4.3-py3-none-any.whl.metadata (4.2 kB)
Collecting bc-detect-secrets==1.5.45 (from checkov)
  Downloading bc_detect_secrets-1.5.45-py3-none-any.whl.metadata (23 kB)
```

## 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

```
[notice] A new release of pip is available: 25.0.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\ASUS> cd C:\Users\dimpl\OneDrive\Desktop\DevSecOps
PS C:\Users\ASUS\OneDrive\Desktop\DevSecOps> chcp 65001
Active code page: 65001
PS C:\Users\ASUS\OneDrive\Desktop\DevSecOps> $env:PYTHONUTF8=1
PS C:\Users\ASUS\OneDrive\Desktop\DevSecOps> checkov -d .
File association not found for extension .py
[ terraform framework ]: 100%|██████████| [1/1], Current
[ secrets framework ]: 100%|██████████| [1/1], Current
```

```

  _ _ _ _ _
 / _ _ _ \
| ( _ _ ) |
 \ _ _ _ /
  _ _ _ _ _

```

By Prisma Cloud | version: 3.2.471

terraform scan results:

Passed checks: 6, Failed checks: 13, Skipped checks: 0

---

## 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:

```
PS C:\Users\ASUS\OneDrive\Desktop\DevSecOps> checkov -d .
File association not found for extension .py
[ terraform framework ]: 100%|██████████|[[1/1], Curre
[ secrets framework ]: 100%|██████████|[[1/1], Current
```

The logo for checkov, featuring the word "checkov" in a stylized, dashed font.

By Prisma Cloud | version: 3.2.471

terraform scan results:

Passed checks: 9, Failed checks: 9, Skipped checks: 0

Check: CKV\_AWS\_93: "Ensure S3 bucket policy does not lockout all  
but root user. (Prevent lockouts needing root account fixes)"