Pratik Agrawal

500123601

Devops B2

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

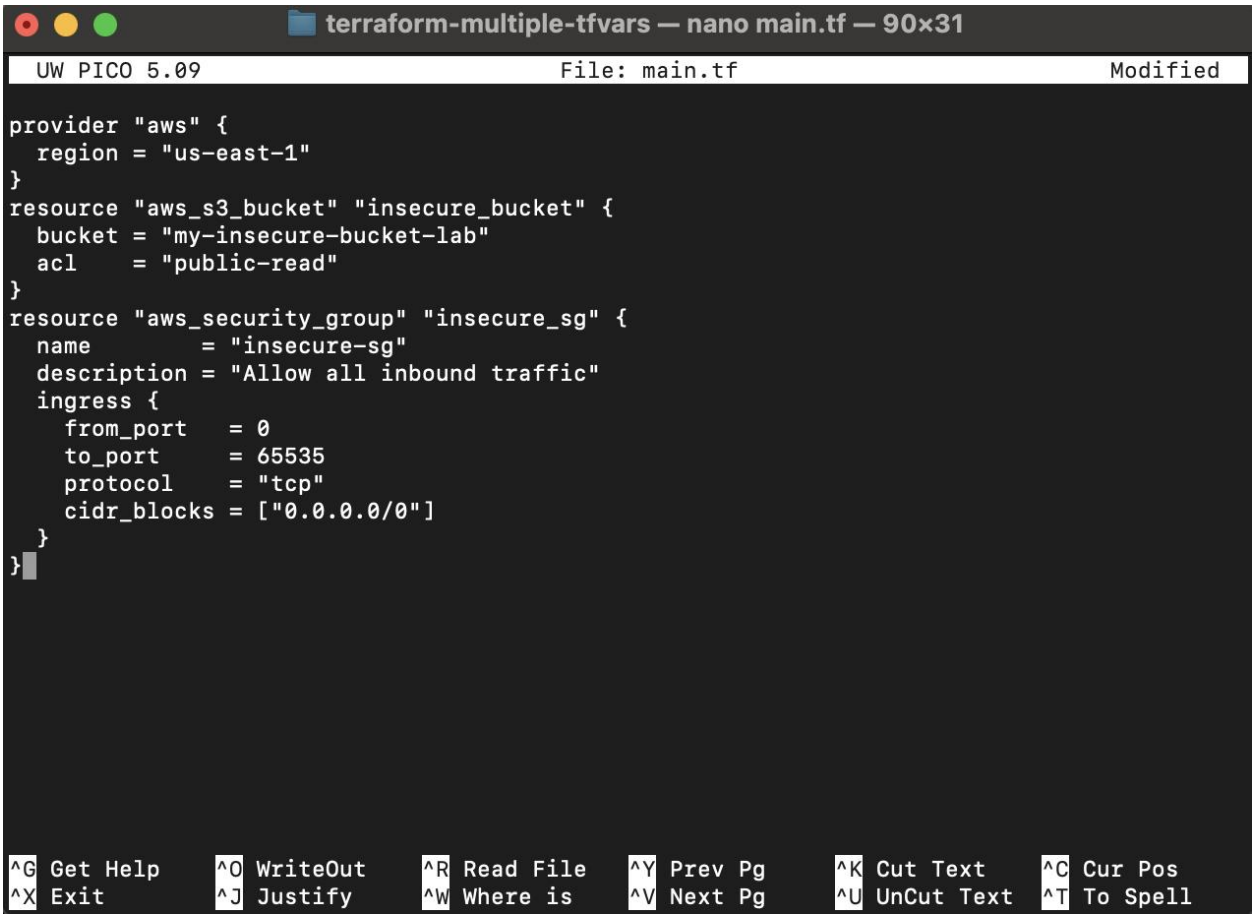
- A Linux/Windows/Mac machine with:
    - Terraform installed (for sample IaC)
    - **Checkov** (pip install checkov) or **tfsec** (brew install tfsec or binary download)

- 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"]
  }
}
```



OUTPUT :-

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:

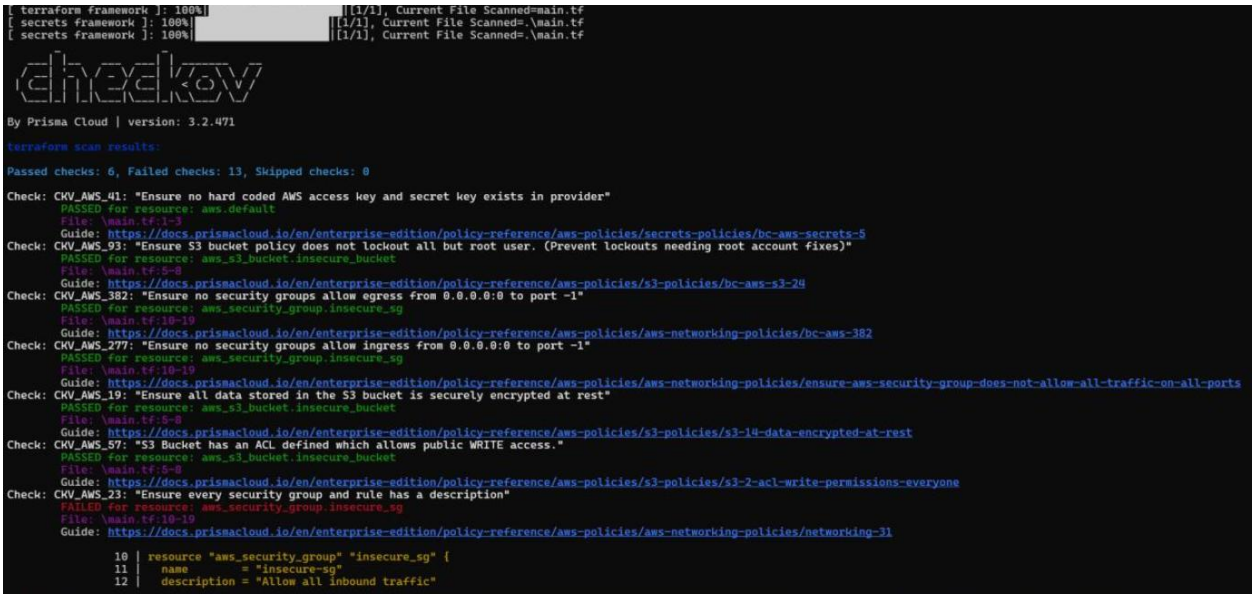
- Warns about S3 bucket without encryption
- Flags open Security Group rules

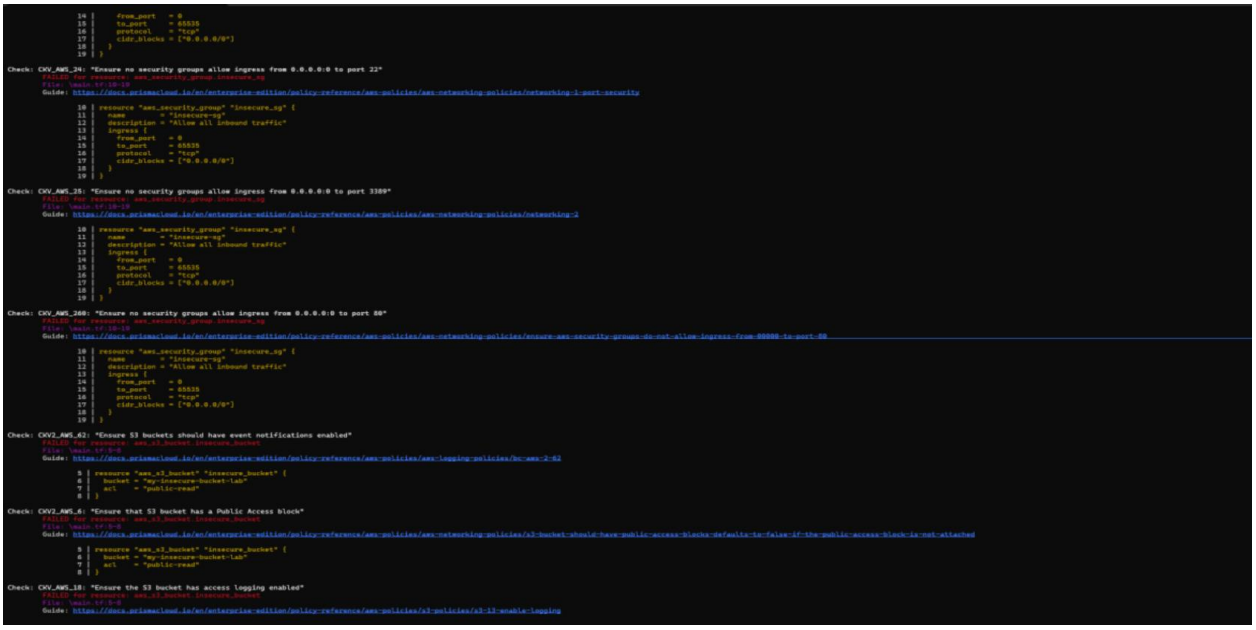
Step 4: Review the Report

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
```

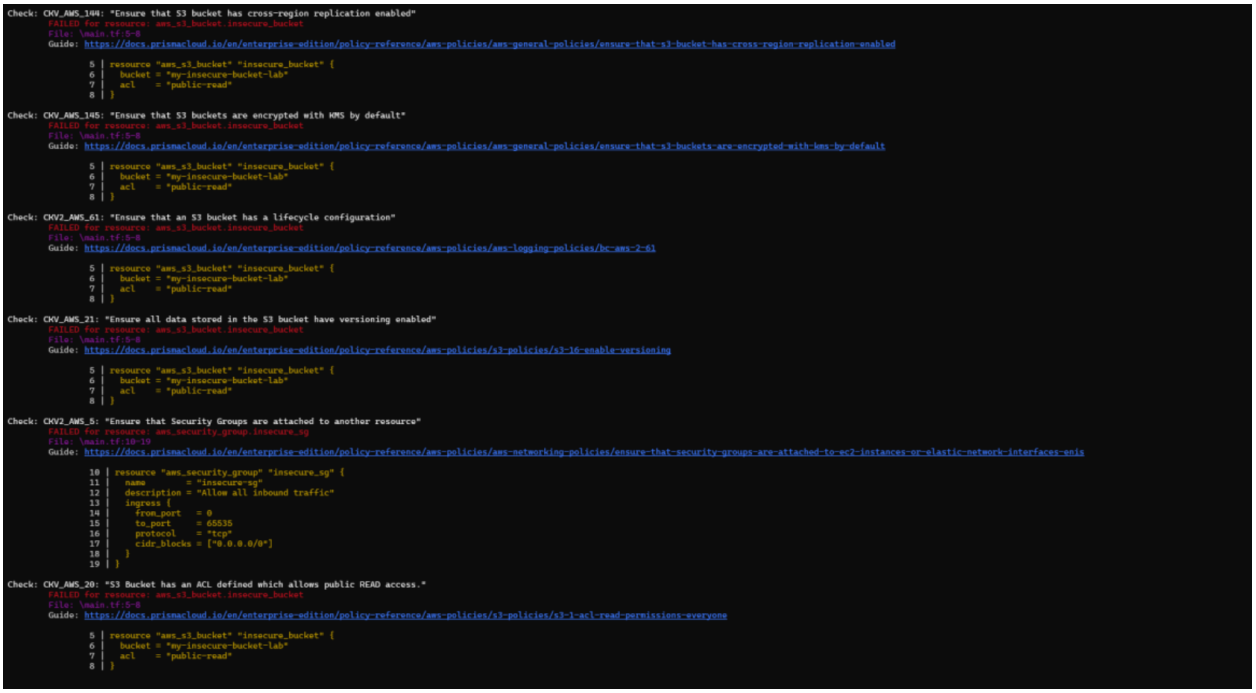
OUTPUT :-


[terraform framework]: 100%|[1/1], Current File Scanned=main.tf
[secrets framework]: 100%|[1/1], Current File Scanned=.\main.tf
[secrets framework]: 100%|[1/1], Current File Scanned=.\main.tf
By Prisma Cloud | version: 3.2.471
terraform scan results:
Passed checks: 6, Failed checks: 13, Skipped checks: 0
Check: CKV_AWS_41: "Ensure no hard coded AWS access key and secret key exists in provider"
PASSED for resource: aws.default
File: \main.tf:1-3
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/secrets-policies/bc-aws-secrets-5
Check: CKV_AWS_93: "Ensure S3 bucket policy does not lockout all but root user. (Prevent lockouts needing root account fixes)"
PASSED for resource: aws_s3_bucket.insecure_bucket
File: \main.tf:5-8
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-24
Check: CKV_AWS_382: "Ensure no security groups allow egress from 0.0.0.0:0 to port -1"
PASSED for resource: aws_security_group.insecure_sg
File: \main.tf:10-19
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-policies/bc-aws-382
Check: CKV_AWS_277: "Ensure no security groups allow ingress from 0.0.0.0:0 to port -1"
PASSED for resource: aws_security_group.insecure_sg
File: \main.tf:10-19
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-policies/ensure-aws-security-group-does-not-allow-all-traffic-on-all-ports
Check: CKV_AWS_19: "Ensure all data stored in the S3 bucket is securely encrypted at rest"
PASSED for resource: aws_s3_bucket.insecure_bucket
File: \main.tf:5-8
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/s3-14-data-encrypted-at-rest
Check: CKV_AWS_57: "S3 Bucket has an ACL defined which allows public WRITE access."
PASSED for resource: aws_s3_bucket.insecure_bucket
File: \main.tf:5-8
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/s3-2-acl-write-permissions-everyone
Check: CKV_AWS_23: "Ensure every security group and rule has a description"
FAILED for resource: aws_security_group.insecure_sg
File: \main.tf:10-19
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-policies/networking-31
10 | resource "aws_security_group" "insecure_sg" {
11 | name = "insecure-sg"
12 | description = "Allow all inbound traffic"



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

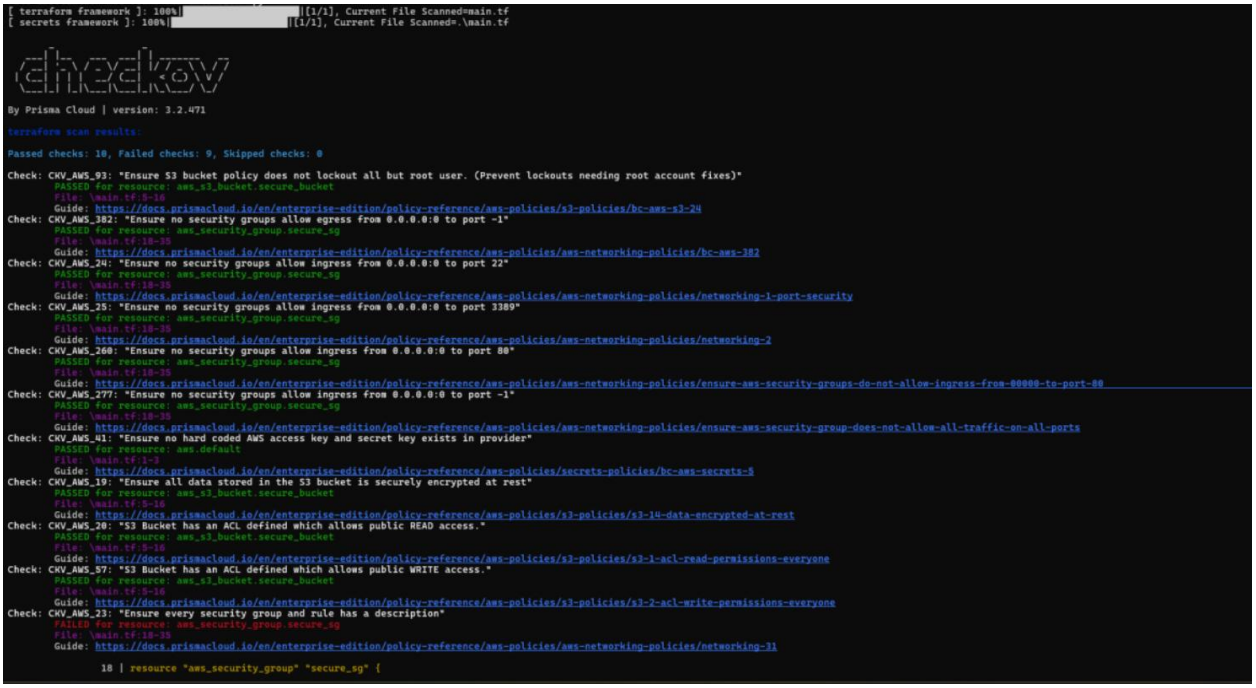
Step 6: Rescan the Template

Run the scan again:

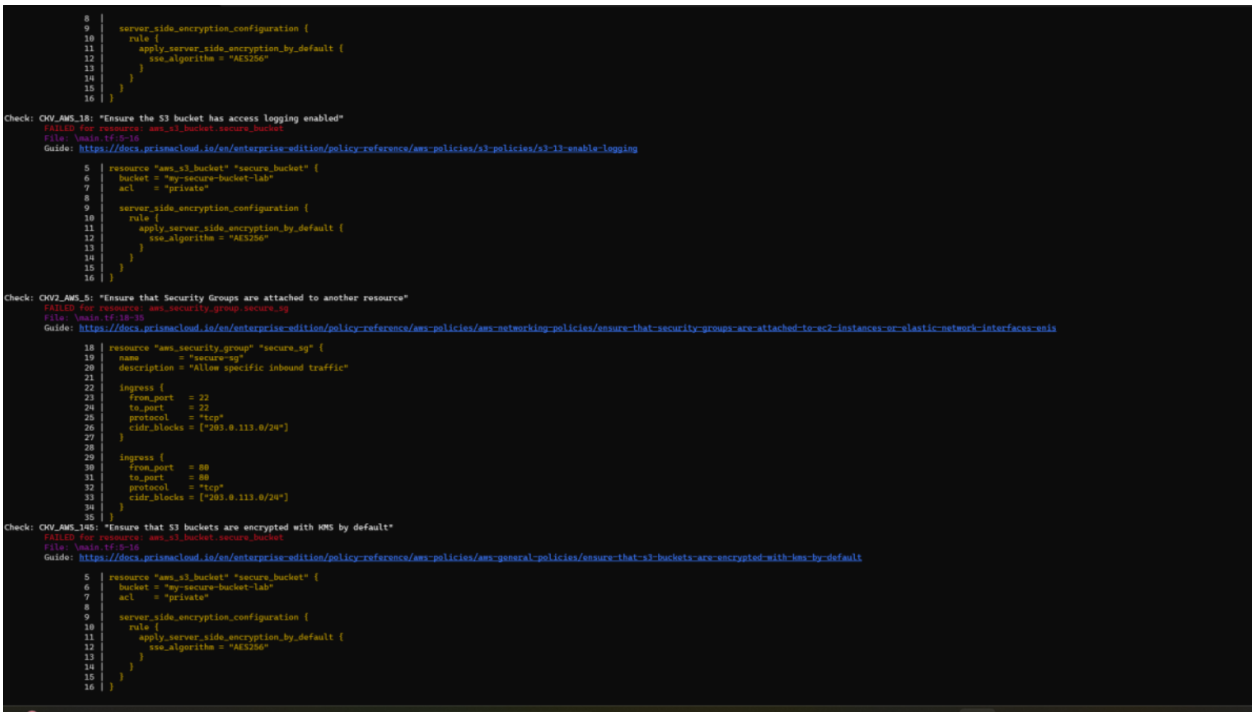
```
checkov -d .
```

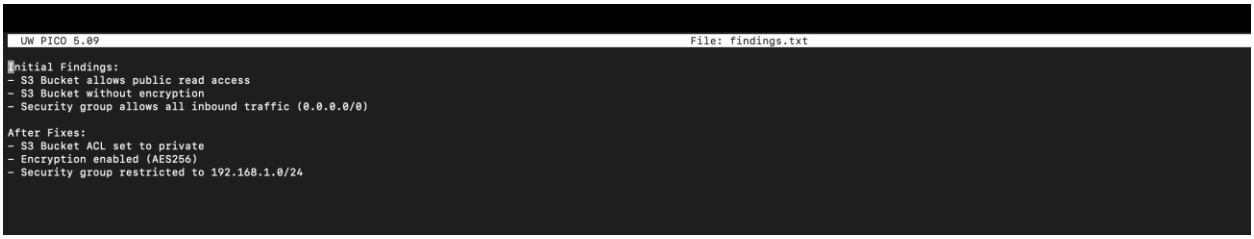
Now the findings should be **resolved or reduced**.

OUTPUT :-



Step 7: Document Findings





Create a simple findings log: