

Lab Exercise 17 – Scanning IaC Templates for Vulnerabilities

Name:-Vansh Bhatt

SapId:- 500125395

R.No:- R2142231689

Batch:- DevOps B1

To:- Hitesh Kumar Sharma Sir

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"]  
}  
}  
}
```

```
provider "aws" {  
    region = "ap-south-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:

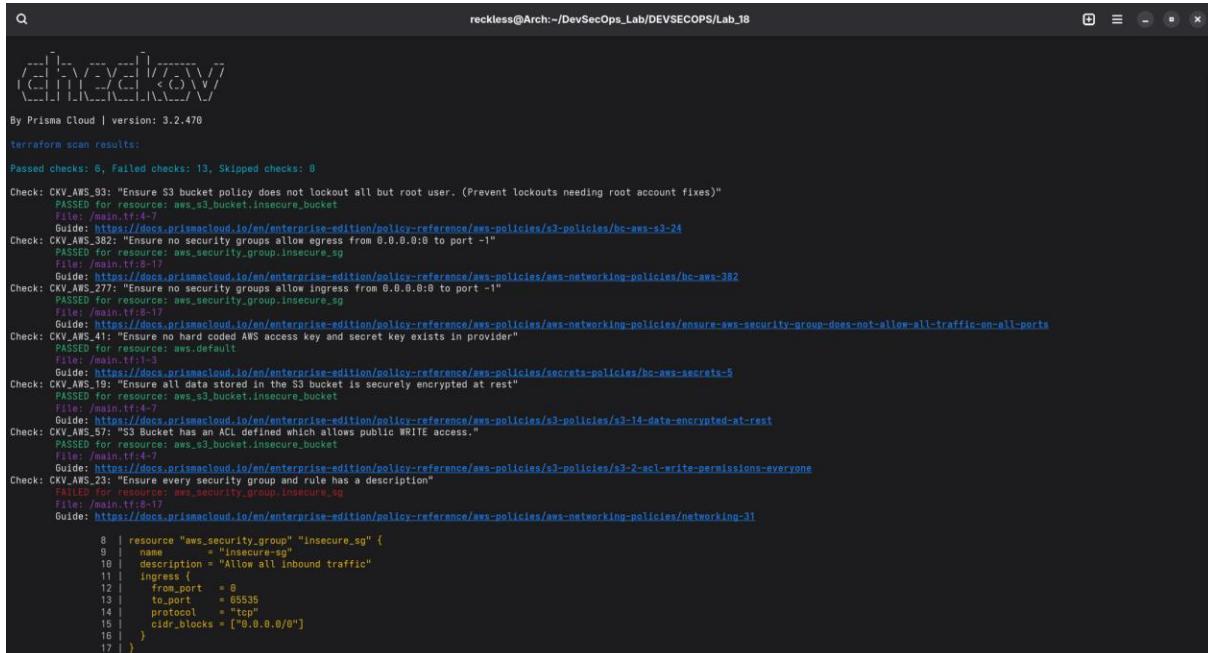
```
checkov -d .
```

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



```
reckless@Arch:~/DevSecOps_Lab/DEVSECOPS/Lab_18
[...]
By Prisma Cloud | version: 3.2.470
terraform scan results:
Passed checks: 6, Failed checks: 13, Skipped checks: 8
Check: CKV_AWS_03: "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:4-7
    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:8-17
    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:8-17
    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_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_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:8-17
    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:4-7
    Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/s3-2-s3-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:8-17
    Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-policies/networking-31
8 | resource "aws_security_group" "insecure_sg" {
9 |   name        = "insecure_sg"
10|   description = "Allow all inbound traffic"
11|   ingress {
12|     from_port  = 8
13|     to_port    = 65535
14|     protocol   = "tcp"
15|     cidr_blocks = ["0.0.0.0/0"]
16|   }
17| }
```

```

reckless@Arch:~/DevSecOps_Lab/DEVSECOPS/Lab_18
File: /tmp/min_tf4-7
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-logging-policies/bc-aws-2-62
  4 | resource "aws_s3_bucket" "insecure_bucket" {
  5 |   bucket = "my-insecure-bucket-lab"
  6 |   acl    = "public-read"
  7 | }

Check: CKV2_AWS_61: "Ensure that an S3 bucket has a lifecycle configuration"
FAILED for resource: aws_s3_bucket.insecure_bucket
File: /tmp/min_tf4-7
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-logging-policies/bc-aws-2-61
  4 | resource "aws_s3_bucket" "insecure_bucket" {
  5 |   bucket = "my-insecure-bucket-lab"
  6 |   acl    = "public-read"
  7 | }

dockerfile scan results:
Passed checks: 0, Failed checks: 2, Skipped checks: 0

Check: CKV_DOCKER_2: "Ensure that HEALTHCHECK instructions have been added to container images"
FAILED for resource: /venv/lib/python3.10/site-packages/checkov/common/util/dockerfile.py
File: /venv/lib/python3.10/site-packages/checkov/common/util/dockerfile.py;1-10
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/docker-policies/docker-policy-index/ensure-that-healthcheck-instructions-have-been-added-to-container-images
  1 | import re
  2 |
  3 | DOCKERFILE_MASK = re.compile(r"(?:[^\\.]?)?([Dd])ockerfile(?:\\..)?$(?:<![Dd])ockerignore")
  4 |
  5 |
  6 | def is_dockerfile(file: str) -> bool:
  7 |     if "ockerfile" not in file:
  8 |         # no need to check the full regex, if 'ockerfile' couldn't be found
  9 |         return False
 10 |     return re.fullmatch(DOCKERFILE_MASK, file) is not None

Check: CKV_DOCKER_3: "Ensure that a user for the container has been created"
FAILED for resource: /venv/lib/python3.10/site-packages/checkov/common/util/dockerfile.py
File: /venv/lib/python3.10/site-packages/checkov/common/util/dockerfile.py;1-10
Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/docker-policies/docker-policy-index/ensure-that-a-user-for-the-container-has-been-created
  1 | import re
  2 |
  3 | DOCKERFILE_MASK = re.compile(r"(?:[^\\.]?)?([Dd])ockerfile(?:\\..)?$(?:<![Dd])ockerignore")
  4 |
  5 |
  6 | def is_dockerfile(file: str) -> bool:
  7 |     if "ockerfile" not in file:
  8 |         # no need to check the full regex, if 'ockerfile' couldn't be found
  9 |         return False
 10 |     return re.fullmatch(DOCKERFILE_MASK, file) is not None

```

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

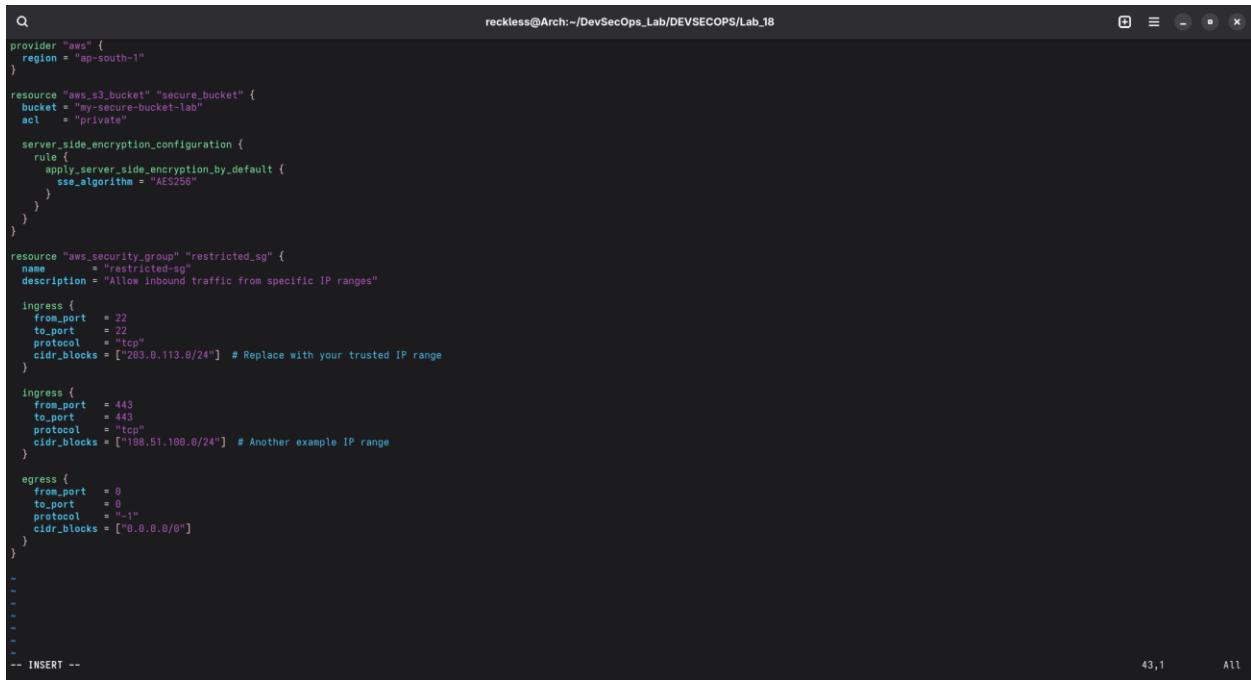
Check: CKV_AWS_260: "Security group allows ingress from 0.0.0.0/0"

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



```
reckless@Arch:~/DevSecOps_Lab/DEVSECOPS/Lab_18
```

```
provider "aws" {
  region = "ap-south-1"
}

resource "aws_s3_bucket" "secure_bucket" {
  bucket = "my-secure-bucket-lab"
  acl    = "private"

  server_side_encryption_configuration {
    rule {
      apply_server_side_encryption_by_default {
        sse_algorithm = "AES256"
      }
    }
  }
}

resource "aws_security_group" "restricted_sg" {
  name     = "restricted-sg"
  description = "Allow inbound traffic from specific IP ranges"

  ingress {
    from_port  = 22
    to_port    = 22
    protocol   = "tcp"
    cidr_blocks = ["263.0.113.0/24"] # Replace with your trusted IP range
  }

  ingress {
    from_port  = 443
    to_port    = 443
    protocol   = "tcp"
    cidr_blocks = ["198.51.100.0/24"] # Another example IP range
  }

  egress {
    from_port  = 0
    to_port    = 0
    protocol   = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

-- INSERT --
```

Step 6: Rescan the Template

Run the scan again:

```
checkov -d .
```

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

```

reckless@Arch:~/DevSecOps_Lab/DEVSECOPS/Lab_18
2025-09-18 11:17:38,465 [MainThread] [WARNING] An unsupported instruction RETURN was used in /venv/lib/python3.13/site-packages/checkov/common/util/dockerfile.py
2025-09-18 11:17:38,465 [MainThread] [WARNING] An unsupported instruction RETURN was used in /venv/lib/python3.13/site-packages/checkov/common/util/dockerfile.py
[ dockerfile framework ]: 100% |[[1/1], Current File Scanned=main.tf]
[ secrets framework ]: 100% |[[1/1], Current File Scanned=main.tf]
[ terraform framework ]: 100% |[[1/1], Current File Scanned=main.tf]
[ kubernetes framework ]: 100% |[[1358/1358], Current File Scanned=venv/lib/python3.13/site-packages/boto3/data/s3/2008-03-01/resources-1.json

By Prisma Cloud | version: 3.2.470

terraform scan results:

Passed checks: 9, Failed checks: 18, 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 lockdown all but root user. (Prevent lockouts needing root account fixes)"
    PASSED for resource: aws_s3_bucket.secure_bucket
        File: ./main.tf:5-18
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-s3-24
Check: CKV_AWS_24: "Ensure no security groups allow ingress from 0.0.0.0:0 to port 22"
    PASSED for resource: aws_security_group.restricted_sg
        File: ./main.tf:18-42
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-policies/networking-1-port-security
Check: CKV_AWS_25: "Ensure no security groups allow ingress from 0.0.0.0:0 to port 3389"
    PASSED for resource: aws_security_group.restricted_sg
        File: ./main.tf:42-43
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-policies/networking-2
Check: CKV_AWS_260: "Ensure no security groups allow ingress from 0.0.0.0:0 to port 80"
    PASSED for resource: aws_security_group.restricted_sg
        File: ./main.tf:18-42
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-policies/ensure-aws-security-groups-do-not-allow-ingress-from-00000-to-port-80
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.restricted_sg
        File: ./main.tf:43-44
        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_26: "S3 Bucket has an ACL defined which allows public READ access."
    PASSED for resource: aws_s3_bucket.secure_bucket
        File: ./main.tf:5-16
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/s3-1-acl-read-permissions-everyone
Check: CKV_AWS_19: "Ensure all data stored in the S3 bucket is securely encrypted at rest"
    PASSED for resource: aws_s3_bucket.secure_bucket
        File: ./main.tf:5-16
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/s3-14-data-encrypted-at-rest

```

Step 7: Document Findings

Create a simple findings log:

DevSecOps Audit Log – Terraform Hardening

Date: 10 September 2025 **Engineer:** Sreyas **Environment:** Lab_18 (Arch Linux) **Tooling:** Terraform + Checkov v3.2.470

🔧 Changes Made to IaC Template

| Component | Action Taken | Status |
|---------------|-------------------------------------|-----------|
| S3 Bucket ACL | Changed from public-read to private | ✓ Secured |
| S3 Encryption | Enabled AES256 encryption | ✓ Secured |
| S3 Versioning | Not enabled initially | ✗ Failed |

| Component | Action Taken | Status |
|-------------------|--|-----------|
| S3 KMS Encryption | Not configured (AES256 used) | ✗ Failed |
| S3 Replication | Not configured | ✗ Failed |
| Security Group | Ingress restricted to specific IP ranges | ✓ Secured |
| SG Descriptions | Missing on ingress/egress rules | ✗ Failed |
| SG Egress | Allowed all outbound traffic | ✗ Failed |
| SG Attachment | Not attached to any EC2 or ENI | ✗ Failed |