# Pulkit
# 500125835
# 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
```

```
[ terraform framework ]: 100%|                    |[3/3], Current File Scanned=variable.tf
[ secrets framework ]: 100%|                      |[3/3], Current File Scanned=.\variable.tf


      _               _
   ___| |__   ___  ___| | _______   __
  / __| '_ \ / _ \/ __| |/ / _ \ \ / /
 | (__| | | |  __/ (__|   < (_) \ V /
  \___|_| |_|\___|\___|_|\_\___/ \_/

By Prisma Cloud | version: 3.2.473

terraform scan results:

Passed checks: 8, Failed checks: 18, Skipped checks: 0

Check: CKV_AWS_46: "Ensure no hard-coded secrets exist in EC2 user data"
        PASSED for resource: aws_instance.web1
        File: \instance.tf:1-8
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/secrets-policies/bc-aws-secrets-1
Check: CKV_AWS_88: "EC2 instance should not have public IP."
        PASSED for resource: aws_instance.web1
        File: \instance.tf:1-8
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/public-policies/public-12
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:15-18
        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:19-28
        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:19-28
        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
```

```
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/s3-2-acl-write-permissions-everyone
Check: CKV_AWS_126: "Ensure that detailed monitoring is enabled for EC2 instances"
        FAILED for resource: aws_instance.web1
        File: \instance.tf:1-8
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-logging-policies/ensure-that-detailed-monitoring-is-enabled-for-ec2-instances

                1 | resource "aws_instance" "web1" {
                2 |   ami           = var.my-ami
                3 |   instance_type = var.my-instance-type
                4 |
                5 |   tags = {
                6 |     Name = "EC2-INSTANCE-variable"
                7 |   }
                8 | }

Check: CKV_AWS_135: "Ensure that EC2 is EBS optimized"
        FAILED for resource: aws_instance.web1
        File: \instance.tf:1-8
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-general-policies/ensure-that-ec2-is-ebs-optimized

                1 | resource "aws_instance" "web1" {
                2 |   ami           = var.my-ami
                3 |   instance_type = var.my-instance-type
                4 |
                5 |   tags = {
                6 |     Name = "EC2-INSTANCE-variable"
                7 |   }
                8 | }

Check: CKV_AWS_79: "Ensure Instance Metadata Service Version 1 is not enabled"
        FAILED for resource: aws_instance.web1
        File: \instance.tf:1-8
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-general-policies/bc-aws-general-31
```

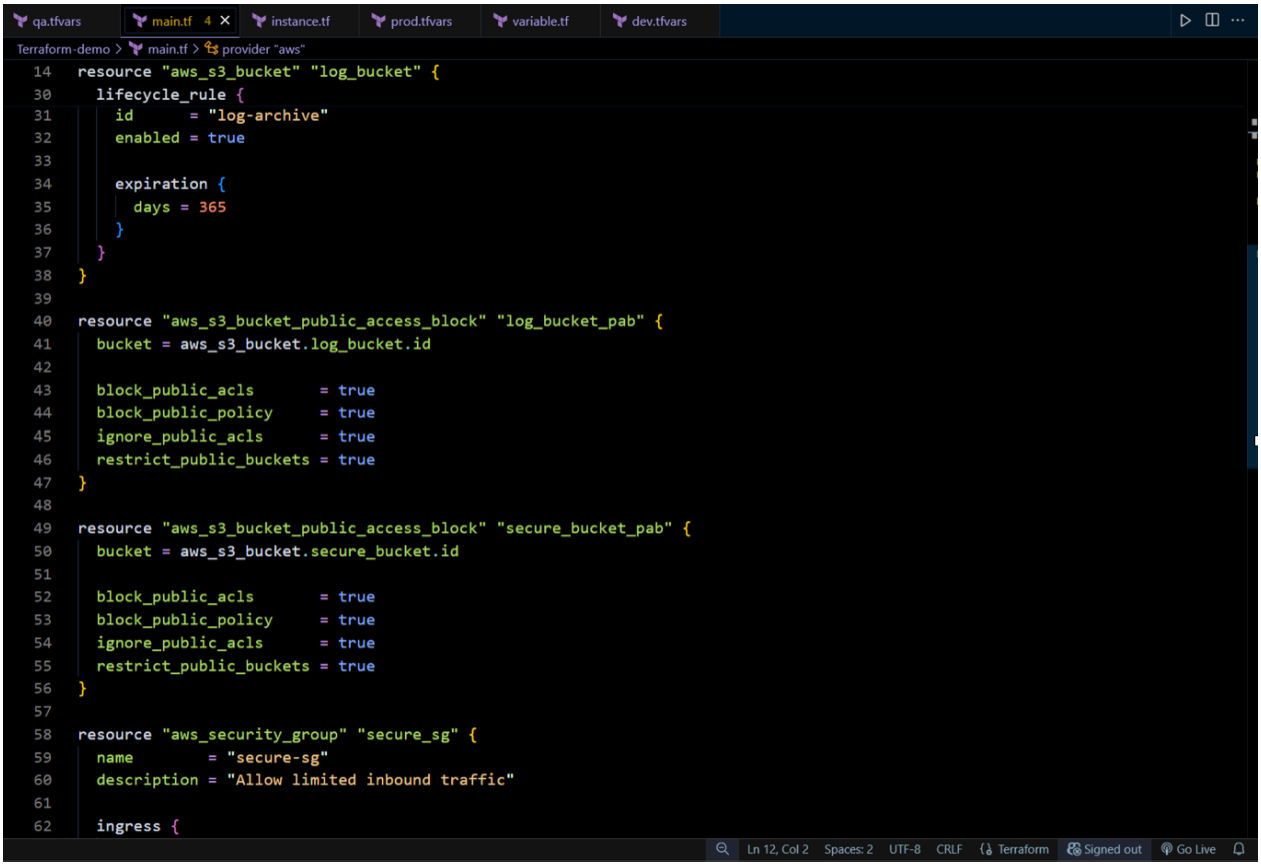

---

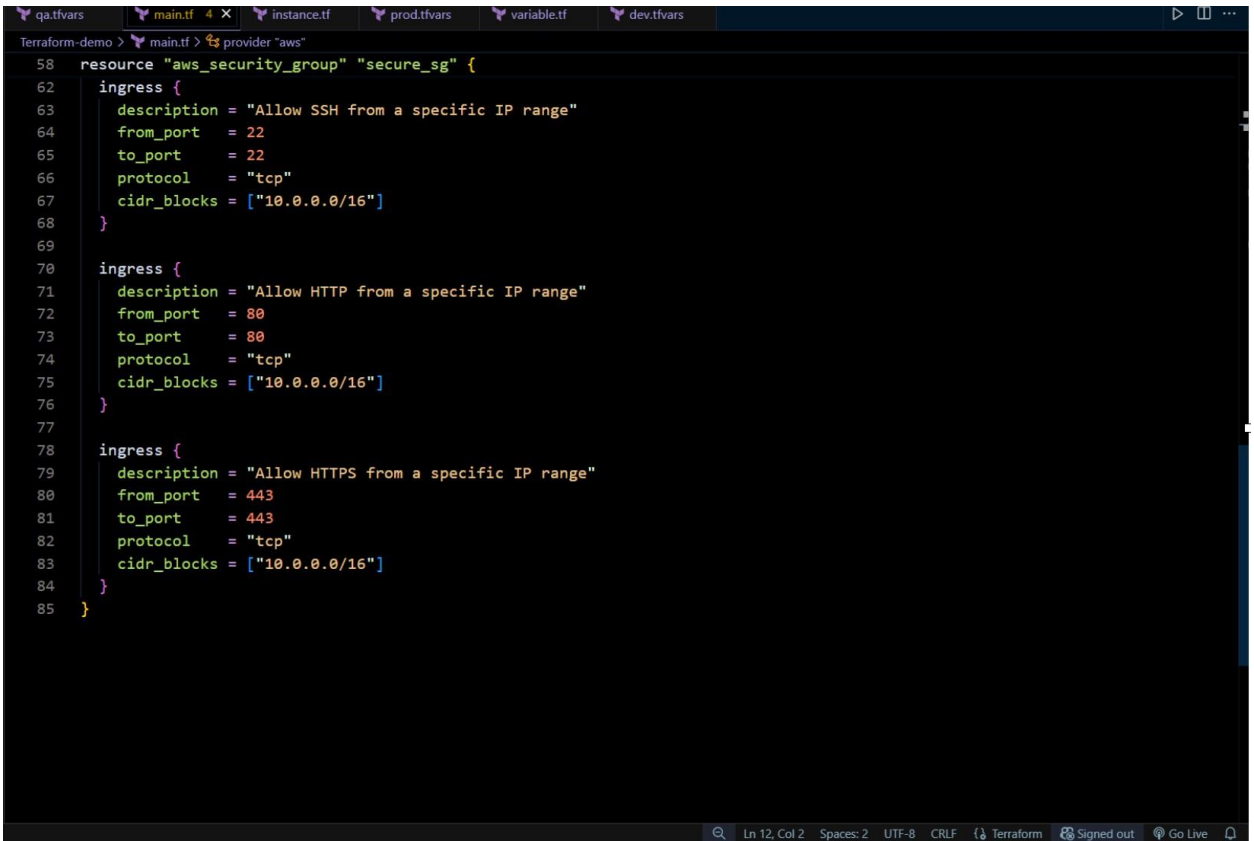
### Step 5: Apply Fixes (Optional)

Modify the IaC template to:

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

### Code:

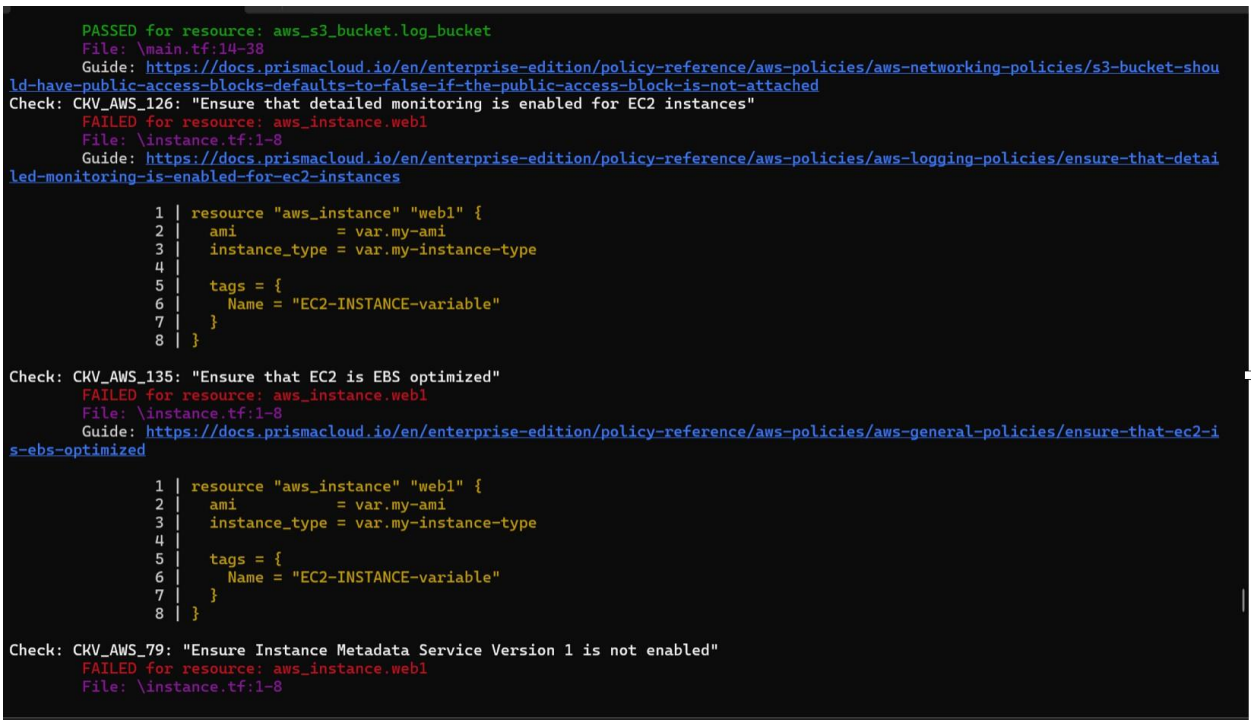

qa.tfvars
main.tf 4
instance.tf
prod.tfvars
variable.tf
dev.tfvars
Terraform-demo > main.tf > provider "aws"
14 resource "aws_s3_bucket" "log_bucket" {
30 lifecycle_rule {
31 id = "log-archive"
32 enabled = true
33
34 expiration {
35 days = 365
36 }
37 }
38 }
39
40 resource "aws_s3_bucket_public_access_block" "log_bucket_pab" {
41 bucket = aws_s3_bucket.log_bucket.id
42
43 block_public_acls = true
44 block_public_policy = true
45 ignore_public_acls = true
46 restrict_public_buckets = true
47 }
48
49 resource "aws_s3_bucket_public_access_block" "secure_bucket_pab" {
50 bucket = aws_s3_bucket.secure_bucket.id
51
52 block_public_acls = true
53 block_public_policy = true
54 ignore_public_acls = true
55 restrict_public_buckets = true
56 }
57
58 resource "aws_security_group" "secure_sg" {
59 name = "secure-sg"
60 description = "Allow limited inbound traffic"
61
62 ingress {
Ln 12, Col 2 Spaces: 2 UTF-8 CRLF Terraform Signed out Go Live


qa.tfvars
main.tf 4
instance.tf
prod.tfvars
variable.tf
dev.tfvars
Terraform-demo > main.tf > provider "aws"
58 resource "aws_security_group" "secure_sg" {
62 ingress {
63 description = "Allow SSH from a specific IP range"
64 from_port = 22
65 to_port = 22
66 protocol = "tcp"
67 cidr_blocks = ["10.0.0.0/16"]
68 }
69
70 ingress {
71 description = "Allow HTTP from a specific IP range"
72 from_port = 80
73 to_port = 80
74 protocol = "tcp"
75 cidr_blocks = ["10.0.0.0/16"]
76 }
77
78 ingress {
79 description = "Allow HTTPS from a specific IP range"
80 from_port = 443
81 to_port = 443
82 protocol = "tcp"
83 cidr_blocks = ["10.0.0.0/16"]
84 }
85 }
Ln 12, Col 2 Spaces: 2 UTF-8 CRLF Terraform Signed out Go Live

### Step 6: Rescan the Template

Run the scan again:

```
checkov -d .
```




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

The failed checks now reduced to 8 those falied check is now due to enabling AES-256 encryption and checkov by default follow KMS configuration

## Step 7: Document Findings

Create a simple findings log:

### 1. S3 Bucket ( `insecure_bucket` -> `secure_bucket` )

The original S3 bucket, `insecure_bucket`, was publicly readable. The updated configuration, now named `secure_bucket`, implements the following security best practices:

- **ACL:** The Access Control List (ACL) was changed from `public-read` to `private`, preventing public access to the bucket's contents.
- **Versioning:** Versioning is now enabled to protect against accidental deletion or modification of objects.
- **Encryption:** Server-side encryption with AES256 is now enabled to encrypt all objects stored in the bucket.
- **Logging:** All access to the bucket is now logged to a separate `log_bucket`.
- **Lifecycle Policy:** A lifecycle policy has been added to manage object transitions to different storage classes (Standard-IA and Glacier) and to expire them after a certain period.
- **Public Access Block:** A public access block has been added to prevent the bucket from being accidentally exposed to the public.

### 2. New S3 Bucket for Logging ( `log_bucket` )

A new S3 bucket, `log_bucket`, has been created to store access logs from the `secure_bucket`. This bucket is also configured with security best practices:

- **ACL:** The ACL is set to `log-delivery-write` to allow the S3 service to write logs to it.
- **Versioning and Encryption:** Versioning and server-side encryption are enabled.
- **Lifecycle Policy:** A lifecycle policy is in place to automatically delete logs after 365 days.
- **Public Access Block:** A public access block is configured to ensure the log bucket remains private.

### 3. Security Group ( `insecure_sg` -> `secure_sg` )

The original security group, `insecure_sg`, allowed all inbound traffic from any source ( `0.0.0.0/0` ) on all TCP ports. This has been replaced with a much more restrictive security group, `secure_sg`, which only allows:

- **SSH (port 22):** from the `10.0.0.0/16` IP range.
- **HTTP (port 80):** from the `10.0.0.0/16` IP range.
- **HTTPS (port 443):** from the `10.0.0.0/16` IP range.