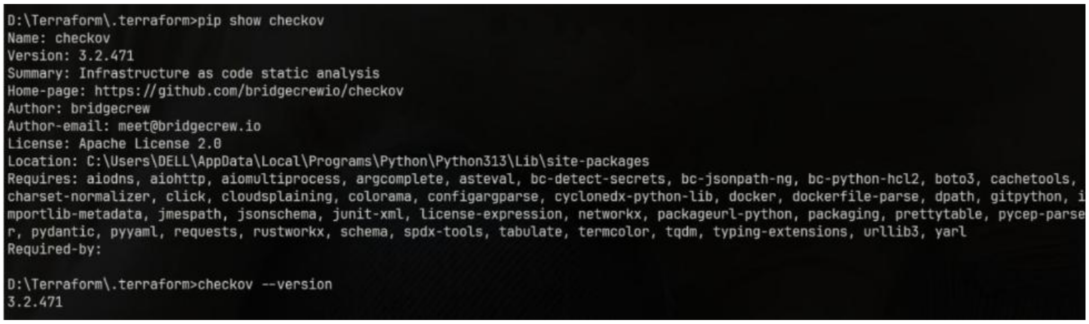
Lab Exercise 18- Scanning IaC Templates for Vulnerabilities

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)



```
D:\Terraform\.terraform>pip show checkov
Name: checkov
Version: 3.2.471
Summary: Infrastructure as code static analysis
Home-page: https://github.com/bridgecrewio/checkov
Author: bridgecrew
Author-email: meet@bridgecrew.io
License: Apache License 2.0
Location: C:\Users\DELL\AppData\Local\Programs\Python\Python313\Lib\site-packages
Requires: aiodns, aiohttp, aiomultiprocess, argcomplete, asteval, bc-detect-secrets, bc-jsonpath-ng, bc-python-hcl2, boto3, cachetools,
charset-normalizer, click, cloudsplaining, colorama, configargparse, cyclonedx-python-lib, docker, dockerfile-parse, dpath, gitpython, i
mportlib-metadata, jmespath, jsonschema, junit-xml, license-expression, networkx, packageurl-python, packaging, prettytable, pycep-parse
r, pydantic, pyyaml, requests, rustworkx, schema, spdx-tools, tabulate, termcolor, tqdm, typing-extensions, urllib3, yarl
Required-by:

D:\Terraform\.terraform>checkov --version
3.2.471
```

Step 1: Create an Insecure IaC Template

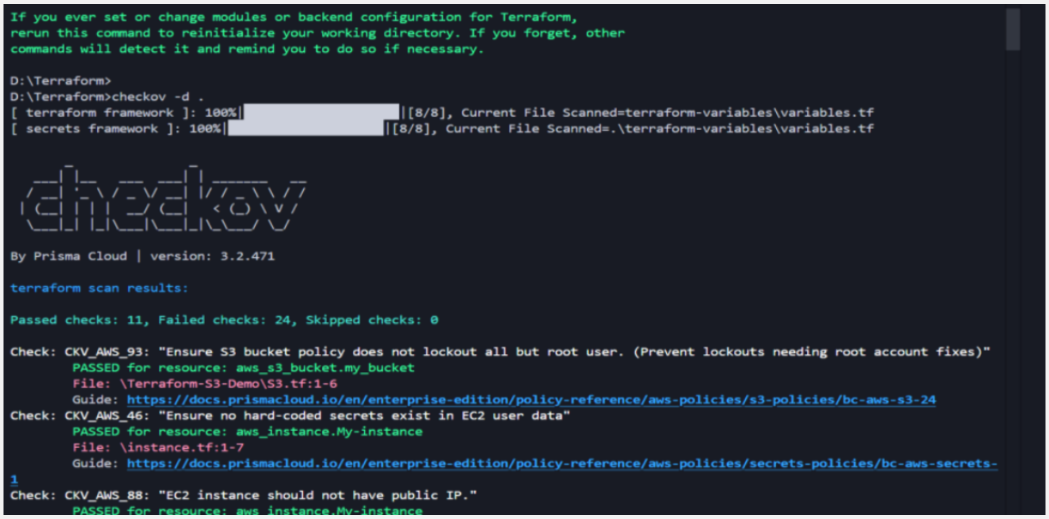
Create a file named main.tf with the following Terraform code:

```
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 .
```



```
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

D:\Terraform>
D:\Terraform>checkov -d .
[ terraform framework ]: 100%|████████████████████|[8/8], Current File Scanned=terraform-variables\variables.tf
[ secrets framework ]: 100%|████████████████████|[8/8], Current File Scanned=.\terraform-variables\variables.tf

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

By Prisma Cloud | version: 3.2.471

terraform scan results:

Passed checks: 11, Failed checks: 24, Skipped checks: 0

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.my_bucket
        File: \Terraform-S3-Demo\S3.tf:1-6
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-24
Check: CKV_AWS_46: "Ensure no hard-coded secrets exist in EC2 user data"
        PASSED for resource: aws_instance.My-instance
        File: \instance.tf:1-7
        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.My-instance
```

Expected Findings:

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

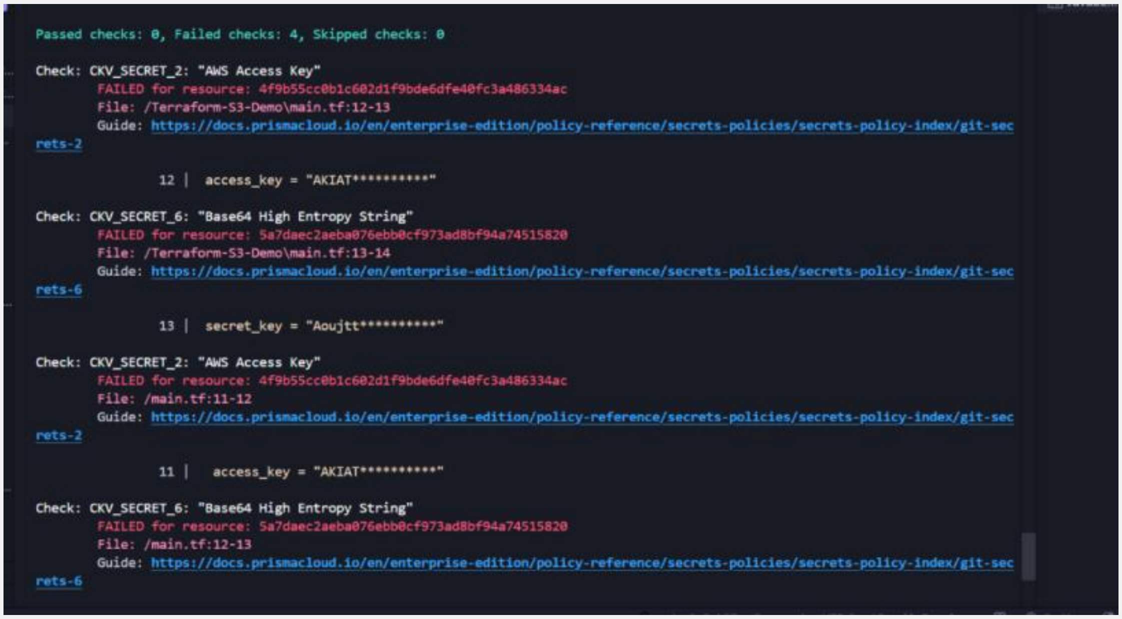
Expected Findings:

- 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
```



```
Passed checks: 0, Failed checks: 4, Skipped checks: 0

Check: CKV_SECRET_2: "AWS Access Key"
        FAILED for resource: 4f9b55cc0b1c602d1f9bde6dfe40fc3a486334ac
        File: /Terraform-S3-Demo\main.tf:12-13
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/secrets-policies/secrets-policy-index/git-sec
rets-2

                12 |  access_key = "AKIAT**********"

Check: CKV_SECRET_6: "Base64 High Entropy String"
        FAILED for resource: 5a7daec2aeba076ebb0cf973ad8bf94a74515820
        File: /Terraform-S3-Demo\main.tf:13-14
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/secrets-policies/secrets-policy-index/git-sec
rets-6

                13 |  secret_key = "Aoujtt**********"

Check: CKV_SECRET_2: "AWS Access Key"
        FAILED for resource: 4f9b55cc0b1c602d1f9bde6dfe40fc3a486334ac
        File: /main.tf:11-12
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/secrets-policies/secrets-policy-index/git-sec
rets-2

                11 |   access_key = "AKIAT**********"

Check: CKV_SECRET_6: "Base64 High Entropy String"
        FAILED for resource: 5a7daec2aeba076ebb0cf973ad8bf94a74515820
        File: /main.tf:12-13
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/secrets-policies/secrets-policy-index/git-sec
rets-6
```

Step 5: Apply Fixes (Optional)

Modify the IaC template to:

- Set S3 bucket ACL to private

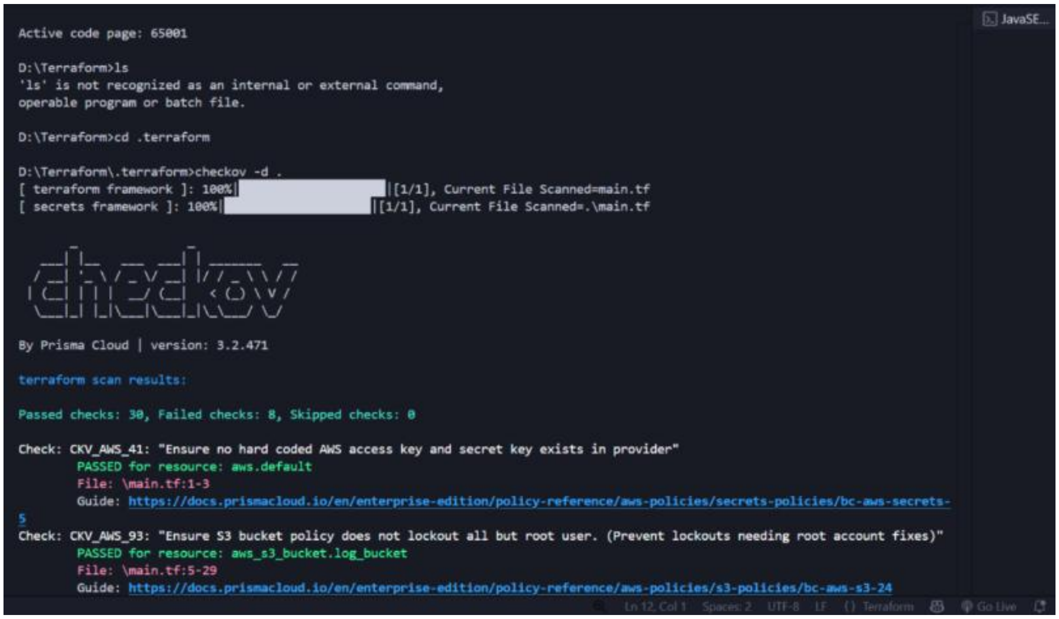
- Enable encryption (AES256)
- Restrict Security Group to specific IP ranges

Step 6: Rescan the Template

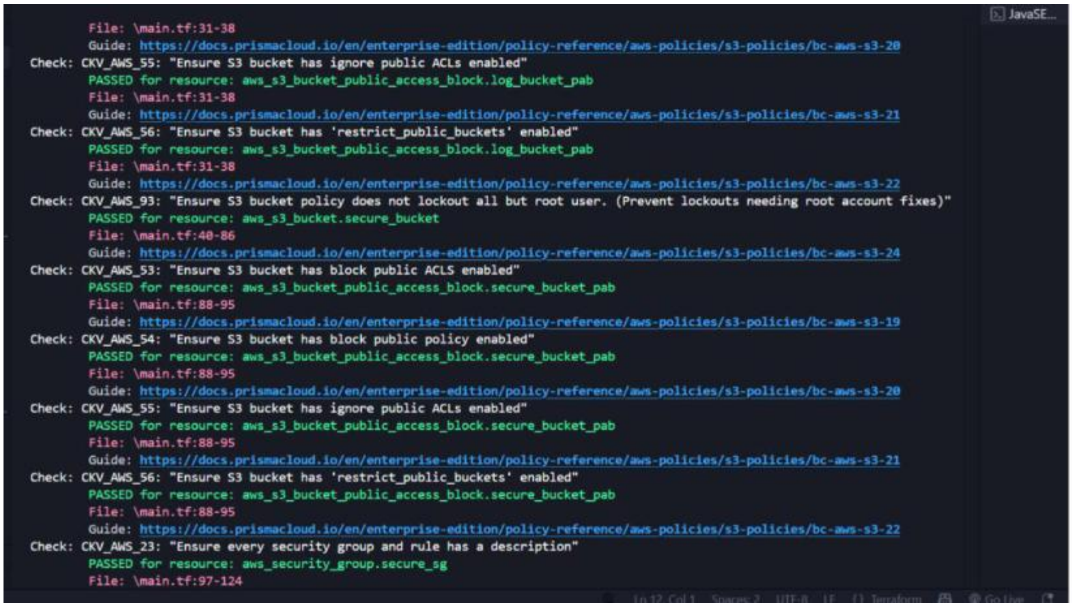
Run the scan again:

```
checkov -d .
```

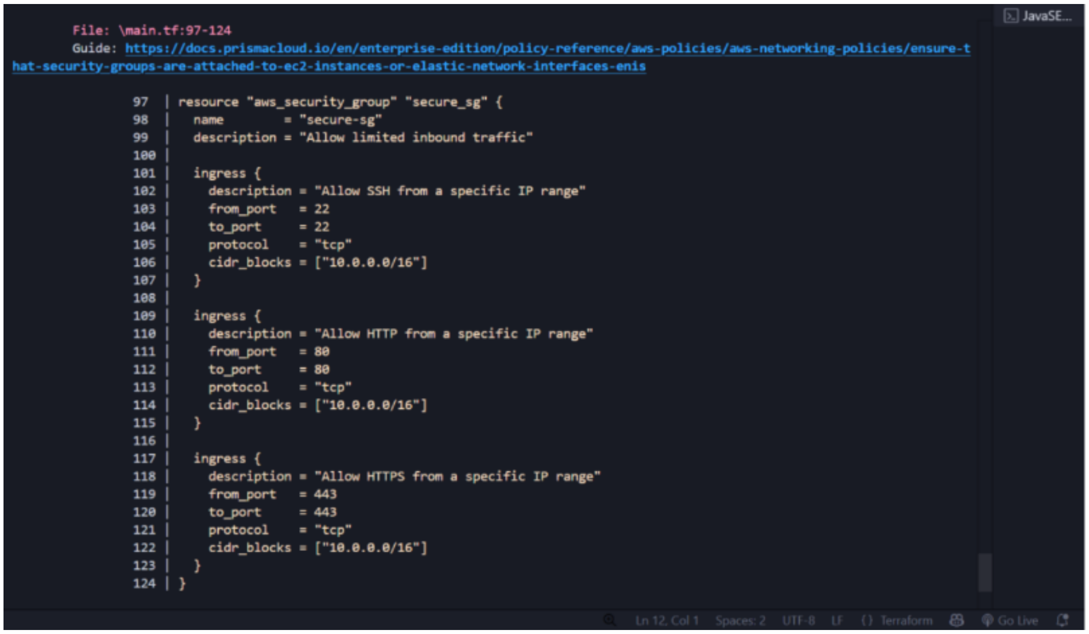
Now the findings should be **resolved or reduced**.



```
Active code page: 65001

D:\Terraform>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

D:\Terraform>cd .terraform

D:\Terraform\.terraform>checkov -d .
[ terraform framework ]: 100%|████████████████████|[1/1], Current File Scanned=main.tf
[ secrets framework ]: 100%|████████████████████|[1/1], Current File Scanned=.\main.tf

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

By Prisma Cloud | version: 3.2.471

terraform scan results:

Passed checks: 30, Failed checks: 8, 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.log_bucket
        File: \main.tf:5-29
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-24
```



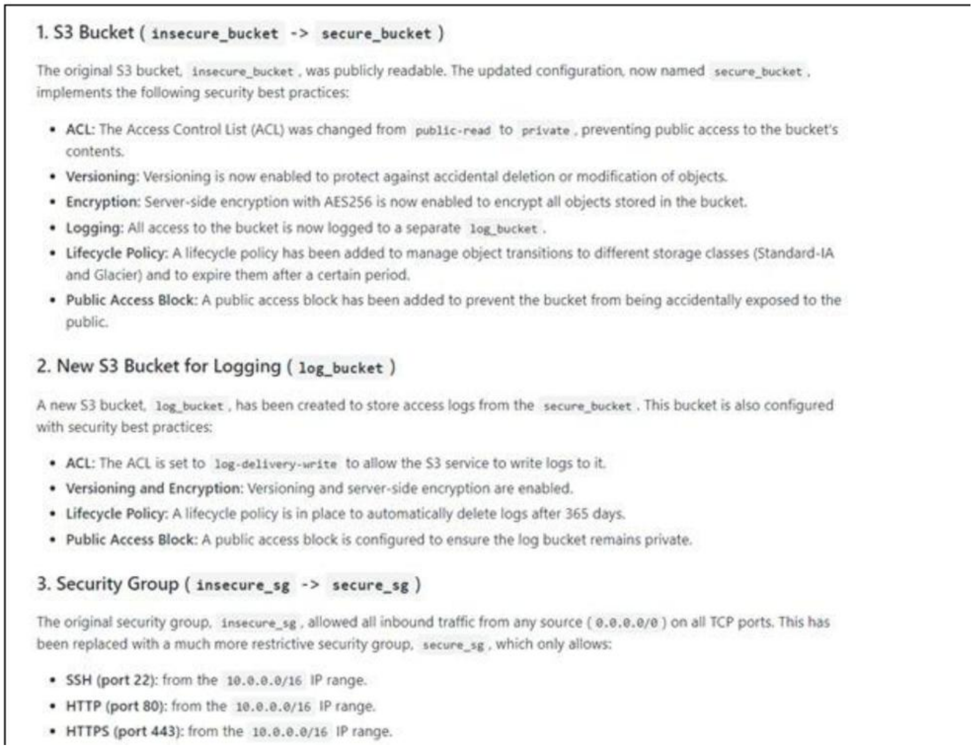
```
        File: \main.tf:31-38
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-20
Check: CKV_AWS_55: "Ensure S3 bucket has ignore public ACLs enabled"
        PASSED for resource: aws_s3_bucket_public_access_block.log_bucket_pab
        File: \main.tf:31-38
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-21
Check: CKV_AWS_56: "Ensure S3 bucket has 'restrict_public_buckets' enabled"
        PASSED for resource: aws_s3_bucket_public_access_block.log_bucket_pab
        File: \main.tf:31-38
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-22
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.secure_bucket
        File: \main.tf:40-86
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-24
Check: CKV_AWS_53: "Ensure S3 bucket has block public ACLS enabled"
        PASSED for resource: aws_s3_bucket_public_access_block.secure_bucket_pab
        File: \main.tf:88-95
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-19
Check: CKV_AWS_54: "Ensure S3 bucket has block public policy enabled"
        PASSED for resource: aws_s3_bucket_public_access_block.secure_bucket_pab
        File: \main.tf:88-95
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-20
Check: CKV_AWS_55: "Ensure S3 bucket has ignore public ACLs enabled"
        PASSED for resource: aws_s3_bucket_public_access_block.secure_bucket_pab
        File: \main.tf:88-95
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-21
Check: CKV_AWS_56: "Ensure S3 bucket has 'restrict_public_buckets' enabled"
        PASSED for resource: aws_s3_bucket_public_access_block.secure_bucket_pab
        File: \main.tf:88-95
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/s3-policies/bc-aws-s3-22
Check: CKV_AWS_23: "Ensure every security group and rule has a description"
        PASSED for resource: aws_security_group.secure_sg
        File: \main.tf:97-124
```



```
        File: \main.tf:97-124
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-policies/ensure-t
hat-security-groups-are-attached-to-ec2-instances-or-elastic-network-interfaces-enis

                97  | resource "aws_security_group" "secure_sg" {
                98  |   name        = "secure-sg"
                99  |   description = "Allow limited inbound traffic"
                100 |
                101 |   ingress {
                102 |     description = "Allow SSH from a specific IP range"
                103 |     from_port   = 22
                104 |     to_port     = 22
                105 |     protocol    = "tcp"
                106 |     cidr_blocks = ["10.0.0.0/16"]
                107 |   }
                108 |
                109 |   ingress {
                110 |     description = "Allow HTTP from a specific IP range"
                111 |     from_port   = 80
                112 |     to_port     = 80
                113 |     protocol    = "tcp"
                114 |     cidr_blocks = ["10.0.0.0/16"]
                115 |   }
                116 |
                117 |   ingress {
                118 |     description = "Allow HTTPS from a specific IP range"
                119 |     from_port   = 443
                120 |     to_port     = 443
                121 |     protocol    = "tcp"
                122 |     cidr_blocks = ["10.0.0.0/16"]
                123 |   }
                124 | }
```

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.