Assignment Cover Sheet

Please fill out and insert as the first page of any essay-style Assignment.

Student Name(s) and Number(s) as per student card (s): 20070145

Programme: MSc Cyber Security

Lecturer Name: Swati Dongre

Module/Subject Title: Advanced Programming Techniques

Assignment Title: Python- based cloud misconfiguration scanner

No of Words: 3050
By submitting this assignment, I am/ we are confirming that:
This assignment is all my/our own work;
Any sources used have been referenced;
I/we have followed the Generative AI instructions/ scale set out in the Assignment Brief;
I/we have read the College rules regarding academic integrity in the QAH Part B Section 3, and the Generative AI Guidelines, and understand that penalties will be applied accordingly if work is found not to be my/our own.
I/we understand that all work uploaded is submitted via Ouriginal, whereby a text-matching report will show any similarities with other texts.

Python- based cloud misconfiguration scanner

Abstract

This project implements a Python- based cloud misconfiguration scanner focusing on Amazon S3. It detects public buckets, public objects, and risky or missing bucket- level configurations such as permissive public access blocks, missing default encryption, website hosting, disabled logging, and disabled versioning. The scanner supports both a **dummy mode** using JSON test data and a **live mode** that runs against a real AWS account using temporary credentials (designed for AWS Vault). The system produces JSON, CSV, and HTML reports, and presents a summarized table in the terminal using Rich or a custom ASCII table. The codebase is organized into small, focused modules, and is supported by automated tests, PowerShell automation scripts, and reusable configuration and data models.

1. Introduction

The goal of this project is to build a realistic, auditable S3 misconfiguration scanner that can be:

- ✓ Safely tested offline using dummy JSON data.
- ✓ Executed against a real AWS account in a controlled way using temporary credentials.
- ✓ Used to produce evidence (reports, logs, test traces) for both technical and audit purposes.

The scanner is structured as a small Python application with clear separation of concerns:

- ✓ **main.py**: CLI entrypoint and mode selection.
- ✓ **scanner/aws_s3.py**: Core S3 scanning rules and AWS interactions.
- ✓ **models.py**: Data model for scanner findings.
- ✓ **utils.py**: JSON loading, report generation (JSON/CSV/HTML), and console summary.
- ✓ **config.py**: Centralized severity and region configuration.
- ✓ **check_creds.py / test_public_bucket.py**: Operational helpers and live test harness.
- ✓ **PowerShell scripts**: End- to- end automation for Windows users.
- ✓ **tests/**: Unit and integration tests, including HTML validation.

The following sections explain each file, its role, and how it interacts with the rest of the system.

2. High- level architecture

At a high level, the scanner follows a layered structure:

- ✓ **CLI / Orchestration**: main.py
- ✓ **Scanning logic**: scanner/aws_s3.py
- ✓ **Data model**: models.py
- ✓ **Utilities / Reporting**: utils.py
- ✓ **Configuration**: config.py
- ✓ **Operational helpers**: check_creds.py, test_public_bucket.py, PowerShell scripts
- ✓ **Testing & sample data**: tests/, sample_aws_resources.json

Data flows are:

1. User invokes **main.py** in either **--mode dummy** or --**mode aws.**
2. **main.py** orchestrates:

- ✓ Dummy: loads JSON from a sample file, calls scanning functions on that data.
- ✓ AWS: creates a **boto3.Session**, enumerates real S3 buckets, and applies scanning logic.

3. **scanner/aws_s3.py** returns a list of Finding objects.
4. **utils.save_report()** turns Finding objects into JSON, CSV, and HTML reports.
5. **utils.print_summary_and_report_path()** prints a compact summary table and report locations.

3. File- by- file explanation

3.1 config.py – Central configuration

Purpose: Central place for tunable constants and defaults.

Key elements:

Severity constants

```
DEFAULT_SEVERITY_PUBLIC_BUCKET = 9
DEFAULT_SEVERITY_WARNING = 5
```

These act as a scale (0–10) to label misconfigurations as high- severity (e.g., public buckets) or medium/low warnings (e.g., missing logging).

AWS configuration

```
DEFAULT_AWS_PROFILE = None
DEFAULT_AWS_REGION = "eu-west-1"
```

The scanner is designed to use **AWS Vault** or environment variables, not static profiles in code. DEFAULT_AWS_REGION is used as the fallback region if none is provided via CLI or AWS_REGION.

Importance: Centralized config makes it easy to tune severity and region without touching logic code.

3.2 models.py – Finding data model

Purpose: Define a simple, serializable structure for scanner findings.

```
from dataclasses import dataclass, field
from typing import Dict

@dataclass
class Finding:
   resource: str
   issue: str
   severity: int
   details: str = ""
   metadata: Dict[str, str] = field(default_factory=dict)
```

- **resource:** Identifier such as s3://my-bucket or aws:s3:list_buckets.
- **issue:** Short description like "Public ACL" or "Missing Default Encryption".
- **severity:** Integer from 0–10.
- **details:** Free- form text providing context (e.g., raw ACL grants, error messages).
- **metadata:** Extra structured fields like rule IDs, settings, or region.

Importance: This structure is used consistently across dummy/live scans, report generation, and tests, giving a stable core representation.

3.3 utils.py – Utilities and reporting

Purpose: Provide reusable helpers for:

- JSON input loading.
- Report generation (JSON, CSV, HTML).
- Console summary printing (Rich or ASCII).

Key functions:

load_json_file(path: str) -> dict

- Validates file existence.
- Opens file with utf-8-sig to handle BOM.
- Parses JSON and throws a detailed error if invalid (with line and column).

save_report(findings: List[Finding], mode: str, extra: dict, out_dir: str) -> Dict[str, str>

- Ensures report directory exists.
- Builds a report dict with:
    - scan_time
    - mode
    - findings (list of Finding dicts via asdict())
    - extra metadata (e.g., region, source file).

- Writes:
    - **JSON**: full metadata and findings.
    - **CSV**: simple table of resource, issue, severity, details.
    - **HTML**: styled table with summary and optional metadata list.

Returns paths to JSON, CSV, and HTML files so they can be printed or tested.

print_summary_and_report_path(...)

- Prints a short summary with total findings.
- If there are findings:
    - Builds a simple row list from Finding objects.
    - Uses **Rich** if installed:
        - Colored severity (red for critical, yellow for warning, green for low).
        - Columns with folding / wrapping.
    - Falls back to simple_ascii_table() if Rich is unavailable.
- Prints report paths at the end.

Importance: utils.py gives you consistent output and reports, regardless of whether you run in dummy or AWS mode, and centralizes any formatting or reporting changes.

3.4 scanner/aws_s3.py – Core S3 scanning logic

Purpose: This is the heart of the scanner. It contains:

- Pure rule functions (work on dicts / strings).
- Live AWS access helpers (using boto3 clients).
- High- level orchestration for both dummy and live scans.

3.4.1 Pure rule helpers

- acl_has_public_grant(acl)
  Checks an ACL for any grant where grantee URI contains AllUsers or AuthenticatedUsers. If present, the bucket/object is considered public.

- bucket_policy_is_public(policy_text)
  Parses bucket policy JSON and looks for "Effect": "Allow" plus a "Principal" that is "*" or equivalent. Flags public bucket policies.

- object_acl_has_public_grant(acl)
  Reuses the same logic from acl_has_public_grant for object-level ACLs.

These functions are deliberately simple and conservative, suitable for a basic checker.

3.4.2 Live AWS helpers

Each of these wraps a boto3 call and returns normalized Python data:

- ✓ list_buckets_live(session) → list of bucket names.
- ✓ get_bucket_acl_live(session, bucket_name) → ACL dict.
- ✓ get_bucket_policy_live(session, bucket_name) → policy JSON string or None.
- ✓ list_objects_live(session, bucket_name) → first page of objects.
- ✓ get_object_acl_live(session, bucket_name, key) → ACL or None.
- ✓ get_public_access_block_live(session, bucket_name) → PublicAccessBlockConfiguration or None.
- ✓ get_bucket_encryption_live(...), get_bucket_website_live(...), get_bucket_logging_live(...), get_bucket_versioning_live(...) → configuration or None.

Each uses ClientError handling and returns None on error where appropriate, to avoid crashing the scanner.

3.4.3 Bucket-level rule functions

- ✓ scan_bucket_acl_from_acl_dict(bucket_name, acl)
  If the ACL has public grants, returns a Finding with:
  - issue="Public ACL"
  - severity=DEFAULT_SEVERITY_PUBLIC_BUCKET
  - metadata={"rule_id": "S3-ACL-001"}

- ✓ scan_bucket_policy_from_text(bucket_name, policy_text)
  If the policy is public (based on bucket_policy_is_public), returns:
  - issue="Public Bucket Policy"
  - rule_id="S3-POLICY-001"

- ✓ scan_objects_for_public_acls(session, bucket_name)
  Lists bucket objects (first page) and checks each object's ACL for public grants. For each public object:
  - issue="Public Object ACL"
  - rule_id="S3-OBJ-ACL-001"

- ✓ scan_bucket_configuration(session, bucket_name)
  Checks:
  - Public Access Block configuration (missing or permissive settings).
  - Default encryption enabled or not.
  - Website hosting configured.

- Logging enabled or disabled.
- Versioning enabled or not.

✓ Produces warning findings such as:

- "Public Access Block Unknown" or "Public Access Block Permissive"
- "Missing Default Encryption"
- "Website Hosting Enabled"
- "Access Logging Disabled"
- "Versioning Not Enabled"

Each finding is associated with a rule ID and uses DEFAULT_SEVERITY_WARNING.

3.4.4 High- level scan functions

scan_all_buckets_from_json(data)
Used in dummy mode. Expects data like sample_aws_resources.json:

```
{
 "buckets": [
   {
    "Name": "public-read-bucket",
    "ACL": { "Grants": [...] }
   }
 ]
}
```

✓ For each bucket entry, it:

- Extracts the name and ACL.
- Runs scan_bucket_acl_from_acl_dict and scan_bucket_policy_from_text (if policy provided).
- Returns a list of Finding objects.

✓ scan_all_buckets_live(session)

✓ Used in live mode. Steps:

1. Calls list_buckets_live(session). On failure, returns a single Finding representing the error.
2. For each bucket:
   - Tries to read ACL, adds “ACL read error” finding if it fails. -Checks bucket policy, object ACLs, and configuration via the functions above.
3. Returns a combined list of all findings for all buckets.

Importance: This module encapsulates all scanning rules and AWS calls, making it testable (via moto) and reusable from both CLI and test harness.

3.5 main.py – CLI entrypoint and orchestration

Purpose: Provide a command- line interface with two modes, dummy and aws.

Key elements:

Argument parsing using argparse:

- ✓ --mode (required): "dummy" or "aws".
- ✓ --file: JSON file for dummy mode.
- ✓ --profile: currently accepted but not used in run_aws (for backwards compatibility).
- ✓ --region: override AWS region.
- ✓ --report-dir: output directory for reports.
- ✓ --print-table: whether to print full findings table.

run_dummy(file_path, report_dir, print_table):

- ✓ Loads JSON via utils.load_json_file.
- ✓ Calls scan_all_buckets_from_json.
- ✓ Saves reports using save_report.
- ✓ Prints summary using print_summary_and_report_path.

run_aws(profile, region, report_dir, print_table):

- ✓ Resolves region: CLI argument → AWS_REGION env → DEFAULT_AWS_REGION.
- ✓ Creates a boto3.Session **without** a profile name, expecting AWS Vault or other environment- based credentials.
- ✓ Calls scan_all_buckets_live.
- ✓ Saves and prints reports as above.

main():

- ✓ Parses arguments.
- ✓ If mode is dummy, ensures --file is provided.
- ✓ Calls run_dummy or run_aws accordingly.

Importance: main.py is what users actually run, and it encodes your credential model (AWS Vault, no hard- coded profiles) and output behavior.

3.6 check_creds.py – Quick credential sanity check

Purpose: Minimal script to verify that AWS credentials are available via the environment and that boto3 can call STS.

```
sess = boto3.Session()
sts = sess.client("sts")
print(sts.get_caller_identity())
```

- ✓ No profile name; assumes AWS Vault or environment has already provided temporary credentials.
- ✓ Catches and prints ClientError and generic Exception with simple messages.

Importance: Before debugging scanner logic, you can confirm that credentials are working.

3.7 test_public_bucket.py – Live test harness

Purpose: End-to-end functional test in a live AWS account (non-production), to prove that the scanner detects a real public bucket and then fully cleans up.

Key steps:

1. Create a unique bucket name using uuid.
2. Create the bucket in a chosen region (AWS_REGION).
3. Apply a **public ACL** and optionally a **public bucket policy**.
4. Upload a sample public object.
5. Run scan_all_buckets_live(session) from scanner.aws_s3.
6. Print findings to the console.
7. Clean up:

- ✓ Delete all objects.
- ✓ Remove bucket policy.
- ✓ Delete the bucket.

The script uses **boto3.Session** and supports optional **AWS_PROFILE** if you want a specific profile, but integrates well with AWS Vault when run through the PowerShell script.

Importance: This is strong evidence for this report—demonstrates that this can create, detect, and then safely remove an intentional misconfiguration.

3.8 PowerShell scripts – Automation for Windows

3.8.1 run_scanner_dummy.ps1

- ✓ Creates and/or activates a Python virtual environment.
- ✓ Installs dependencies from requirements.txt.
- ✓ Validates that the dummy JSON file exists.

- ✓ Runs:
- ✓ python main.py --mode dummy --file "<dummy_file>" --print-table
- ✓ Prints contents of the **reports** directory.
- ✓ Useful for:
- ✓ Teaching/demo environments.
- ✓ Quick, AWS- free test runs.

3.8.2 run_scanner_end_to_end.ps1

- ✓ Checks Python and venv.
- ✓ Installs dependencies.
- ✓ Validates AWS Vault is installed and that the given profile (e.g., **scanner-user**) can run aws **sts get-caller-identity**.
- ✓ Runs **test_public_bucket.py** via **aws-vault exec**.
- ✓ Runs **main.py --mode aws** with the chosen region.
- ✓ Shows report files afterward.

This script provides a one- click way to:

1. Prove AWS Vault is configured correctly.
2. Run a live test harness.
3. Run the scanner in AWS mode.

3.9 tests/test_aws_s3.py – Unit and integration tests

Purpose: Automated tests for the scanner's core logic and report generation.

Key tests:

test_dummy_public_bucket_and_reports:

- ✓ Creates a small JSON structure representing a bucket with a public ACL.
- ✓ Runs scan_all_buckets_from_json.
- ✓ Ensures exactly one finding exists.
- ✓ Uses save_report to produce JSON, CSV, and HTML in a temporary directory.
- ✓ Asserts that report files exist and that JSON content reflects the expected mode and finding.

test_live_public_bucket_and_reports (@mock_s3):

- ✓ Uses moto to simulate AWS S3.

- ✓ Creates a bucket and applies a public ACL.
- ✓ Calls scan_all_buckets_live(session).
- ✓ Asserts that a finding for s3://test-bucket is present.
- ✓ Saves reports and asserts their existence.

Importance: These tests show that your scanner is not just manually tested, but has automated coverage of both dummy and live modes (with AWS mocked out).

3.10 tests/test_aws_s3_html.py – HTML validation tests

Purpose: Validate that the HTML report produced by **save_report** contains the expected information.

Key tests:

For dummy data:

- ✓ Generate HTML report.
- ✓ Parse with BeautifulSoup.
- ✓ Assert that the <h2> contains "mode: dummy".
- ✓ Assert that the table exists, has rows, and includes a row with:
    - s3://public-bucket
    - Public ACL
    - Severity 9.

For live (mocked) data:

- ✓ Create mock bucket with public ACL.
- ✓ Run scan and save HTML report.
- ✓ Parse and assert s3://test-bucket appears in the table.

Importance: This is especially useful for this assignment, as it proves HTML output is structured and testable, not just "pretty".

3.11 requirements.txt – Dependencies

Lists all Python packages required:

- ✓ **boto3**, **botocore** – AWS API access.
- ✓ **moto** – AWS mocking for tests.
- ✓ **pytest** – Test runner.
- ✓ **tabulate** – (Used in other contexts, e.g., CSV readers; not core scanner).
- ✓ **beautifulsoup4** – HTML parsing for tests.
- ✓ **rich** – Colorful console output (optional but used if installed).

3.12 Sample and policy JSON files

scanner-s3-policy.json
Sample IAM policy allowing minimal S3 actions like **ListAllMyBuckets** and **GetBucketAcl**. This can be attached to your scanner role/user.

sample_aws_resources.json
Large test dataset used in dummy mode. Includes a variety of bucket ACL scenarios:

- ✓ Private only.
- ✓ Public read.
- ✓ Public read/write.
- ✓ Authenticated users.
- ✓ Mixed grants.
- ✓ Missing/empty ACLs.
- ✓ Complex grantee types.

This file is central to offline testing and demonstration.

4. OOP and design principles

Your codebase demonstrates several good practices:

Encapsulation:

- ✓ Scanning logic is encapsulated in scanner/aws_s3.py.
- ✓ Credential handling and region resolution are encapsulated in main.py.
- ✓ Reporting logic is encapsulated in utils.py.

Separation of concerns:

- ✓ CLI/UX (main.py) vs scanning rules (scanner/aws_s3.py) vs reporting (utils.py) vs config (config.py).

Testability:

- ✓ Core logic uses small, pure helper functions.
- ✓ Use of moto and pytest for automated tests.
- ✓ HTML output is tested structurally with BeautifulSoup.

Safety and auditability:

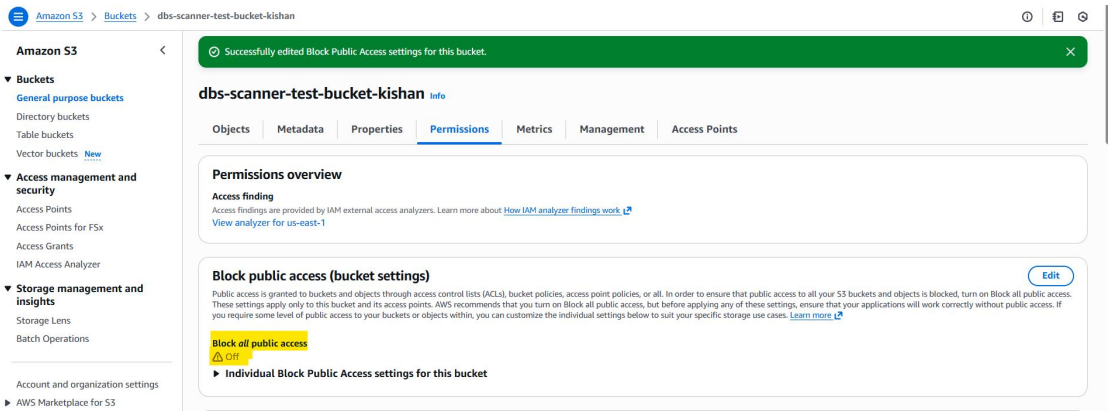
- ✓ Dummy mode avoids any AWS interaction.
- ✓ Live tests create temporary resources and clean them up.
- ✓ Findings are persisted in multiple formats for later audit (JSON/CSV/HTML).

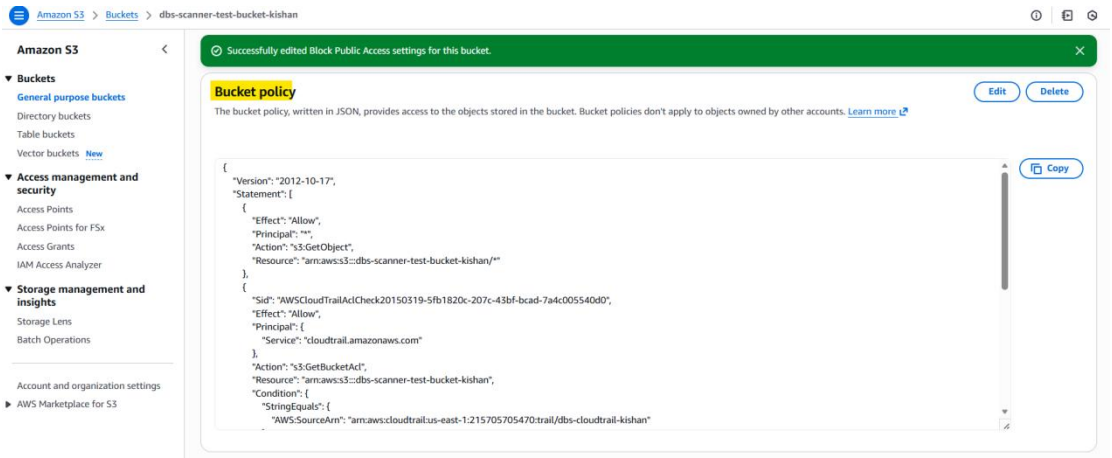
5. Test execution

Misconfigurations test data setup to trigger CloudTrail & Cloud Misconfigurations Scanner

S3 Bucket name - dbs-scanner-test-bucket-kishan

1. Disable “Block Public Access”





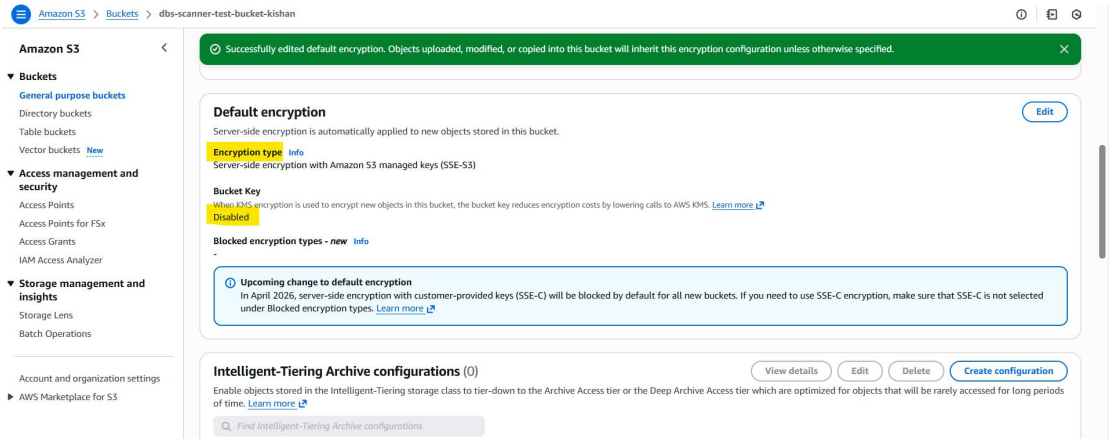

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::dbs-scanner-test-bucket-kishan/*"
    },
    {
      "Sid": "AWSCloudTrailAclCheck20150319-5fb1820c-207c-43bf-bcad-7a4c005540d0",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
```

```
        "Action": "s3:GetBucketAcl",
        "Resource": "arn:aws:s3:::dbs-scanner-test-bucket-kishan",
        "Condition": {
           "StringEquals": {
              "AWS:SourceArn":          "arn:aws:cloudtrail:us-east-1:215705705470:trail/dbs-cloudtrail-
kishan"
           }
        }
     },
     {
        "Sid": "AWSCloudTrailWrite20150319-fff7ad90-334c-4af2-90f2-01fb150b022f",
        "Effect": "Allow",
        "Principal": {
           "Service": "cloudtrail.amazonaws.com"
        },
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3:::dbs-scanner-test-bucket-kishan/AWSLogs/215705705470/*",
        "Condition": {
           "StringEquals": {
              "AWS:SourceArn":          "arn:aws:cloudtrail:us-east-1:215705705470:trail/dbs-cloudtrail-
kishan",
              "s3:x-amz-acl": "bucket-owner-full-control"
           }
        }
     }
  ]
}
```

Notes:

Safe if the bucket is empty
Scanner will detect “public bucket risk”

2. Remove default encryption

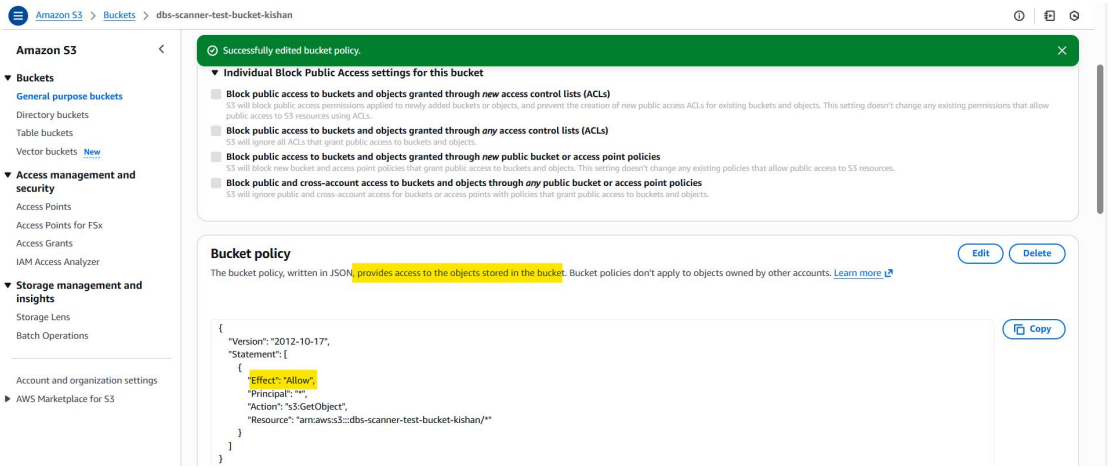


Notes:

Safe
Scanner will detect “unencrypted bucket”

3. Add a bucket policy that allows public read

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::dbs-scanner-test-bucket-kishan/*"
   }
 ]
}
```



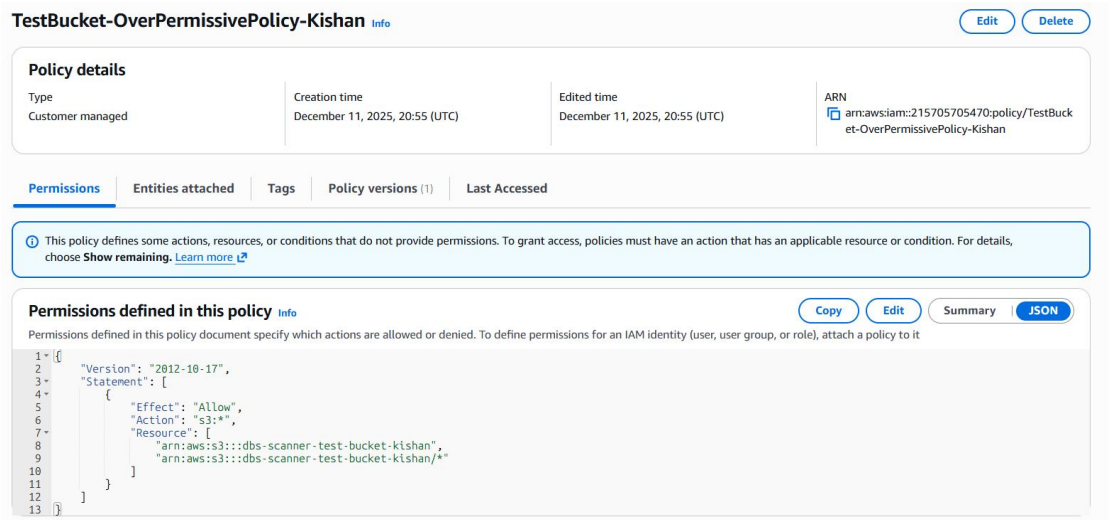
Notes:

Safe if no files inside
Scanner will detect “public access via bucket policy”

4. Add an overly permissive IAM policy

Policy name - TestBucket-OverPermissivePolicy-Kishan

Each action generates CloudTrail events like:
- DetachUserPolicy
- AttachUserPolicy
- PutUserPolicy (if inline)
- PutRolePolicy (if role-based)
These will appear in your S3 CloudTrail logs**.**



Action: "s3:*"

This allows **every possible S3 action**, including:

- Read
- Write
- Delete
- ACL changes
- Bucket policy changes
- Versioning changes
- Encryption changes
- Public access changes

This is far beyond least privilege.

Because:

- s3:* = **all S3 actions** (read, write, delete, list, ACL, policy, etc.)
- Resource includes both the bucket and all objects
- No conditions
- No restrictions

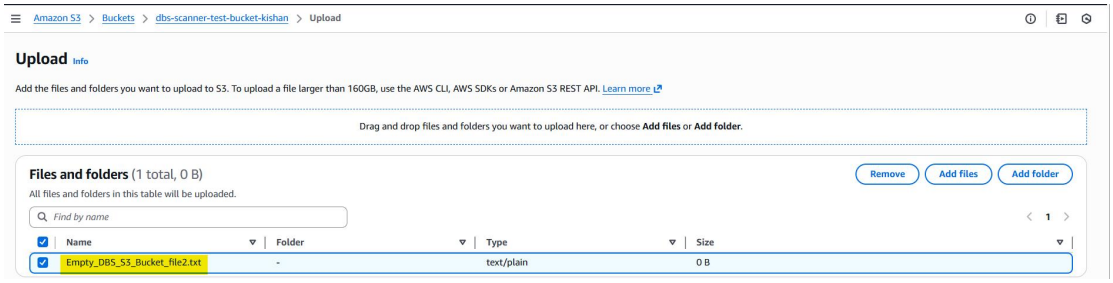
This is exactly the kind of misconfiguration cloud security tools flag

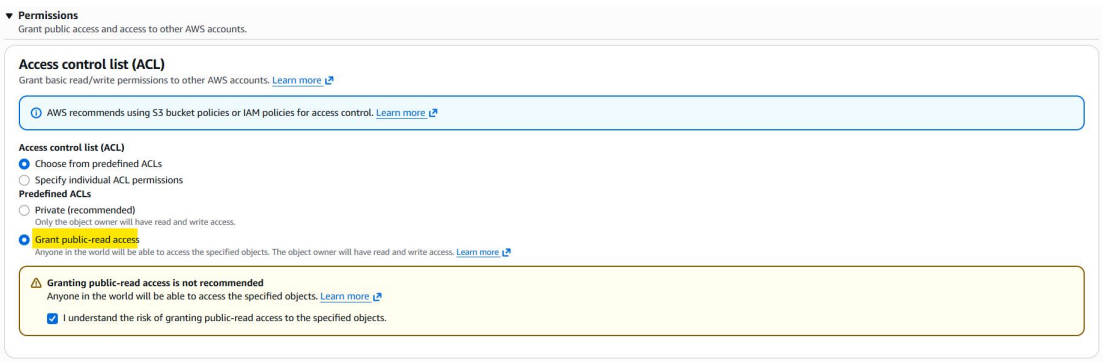
Example: s3:* on the bucket
Safe
Scanner will detect “excessive permissions”

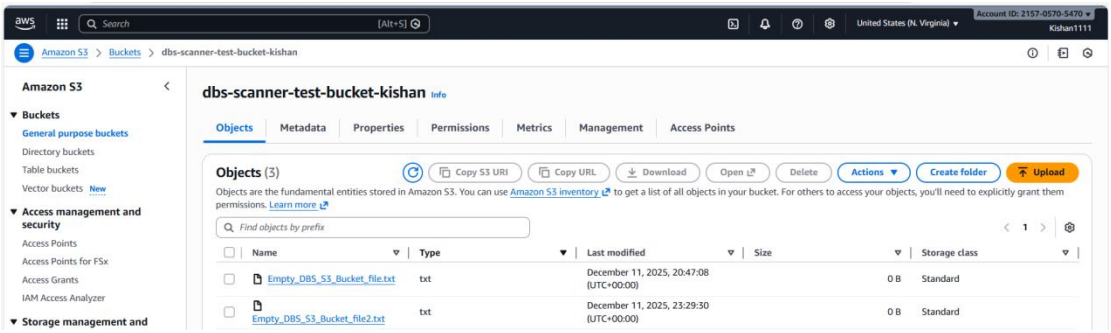
CloudTrail will record:

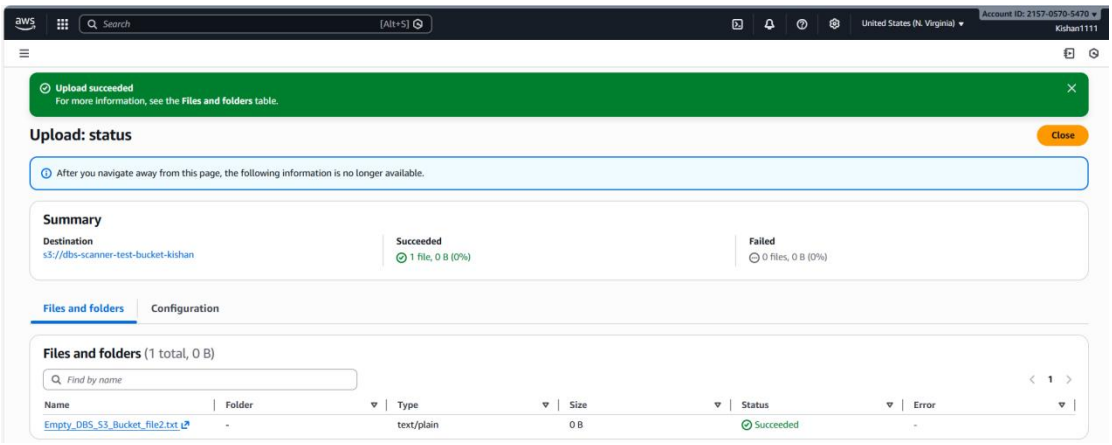
- Who made the IAM change
- When it happened
- What policy was attached
- What permissions it contained
- Which bucket it affects
- The source IP
- The AWS console or API used

5. Upload a dummy file and make it public





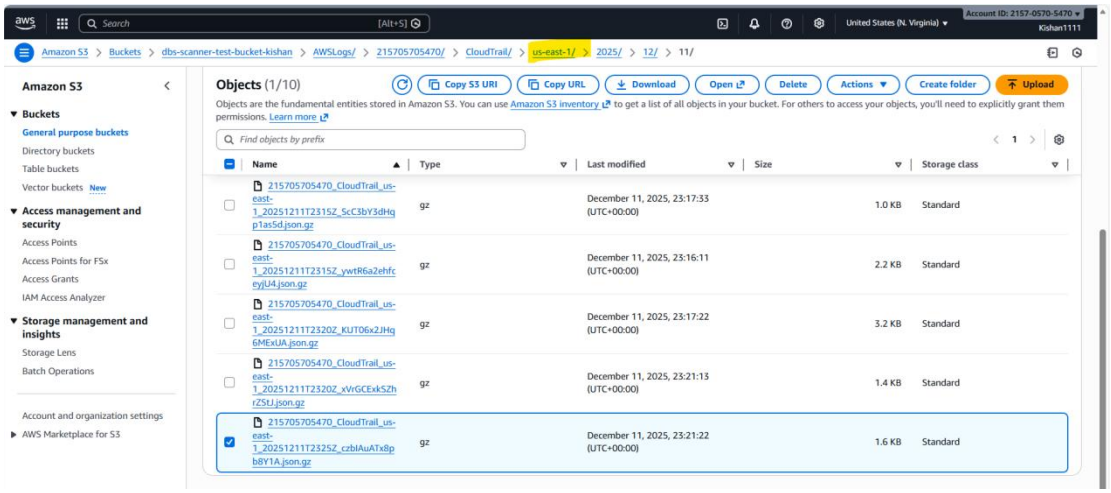




Safe
Scanner will detect “public object”

Verify cloud misconfigurations via cloud Trail

Cloud Bucket name - dbs-cloudtrail-logs-kishan



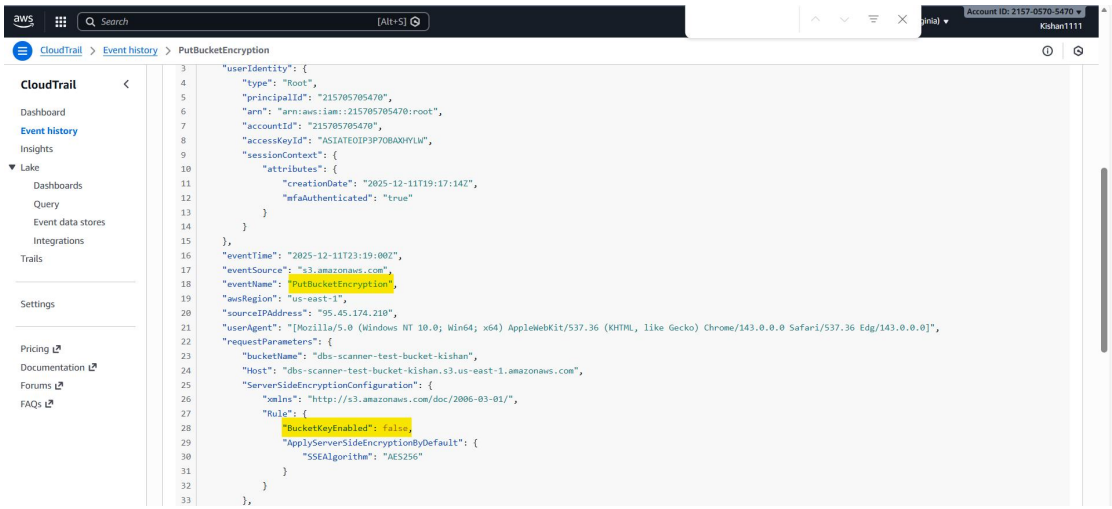
PutBucketPolicy → when you added public access

A misconfiguration was created
It was done by the root user
From your IP
At a specific timestamp
With the exact policy that made the bucket public
Logged correctly by CloudTrail

This means:
Anyone (Principal: "*")
Can read objects (s3:GetObject)
In your test bucket

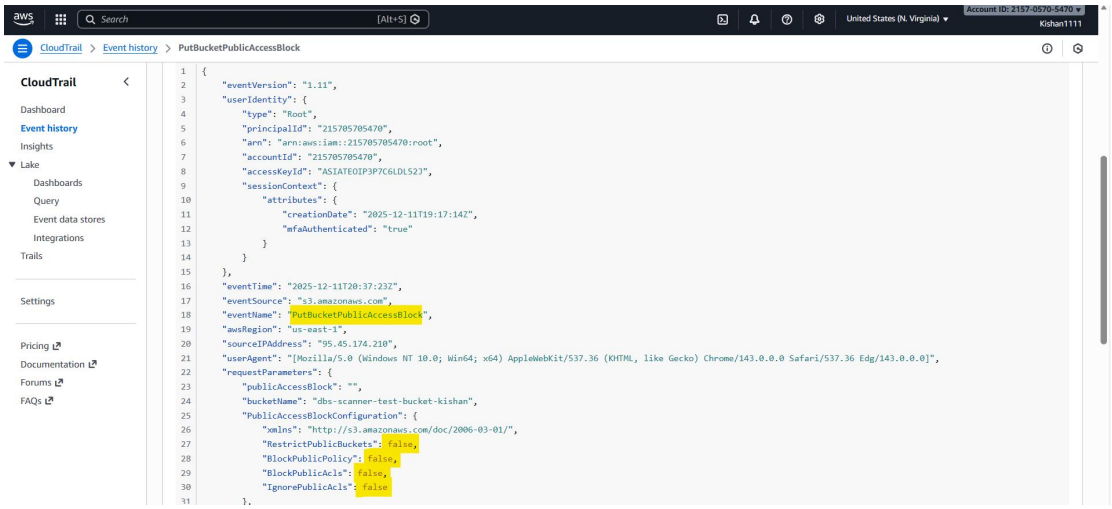
PutBucketEncryption → when you disabled encryption

A sensitive S3 security configuration change
Performed by the root user
From your IP
At a specific timestamp
With the exact encryption settings
Logged correctly by CloudTrail



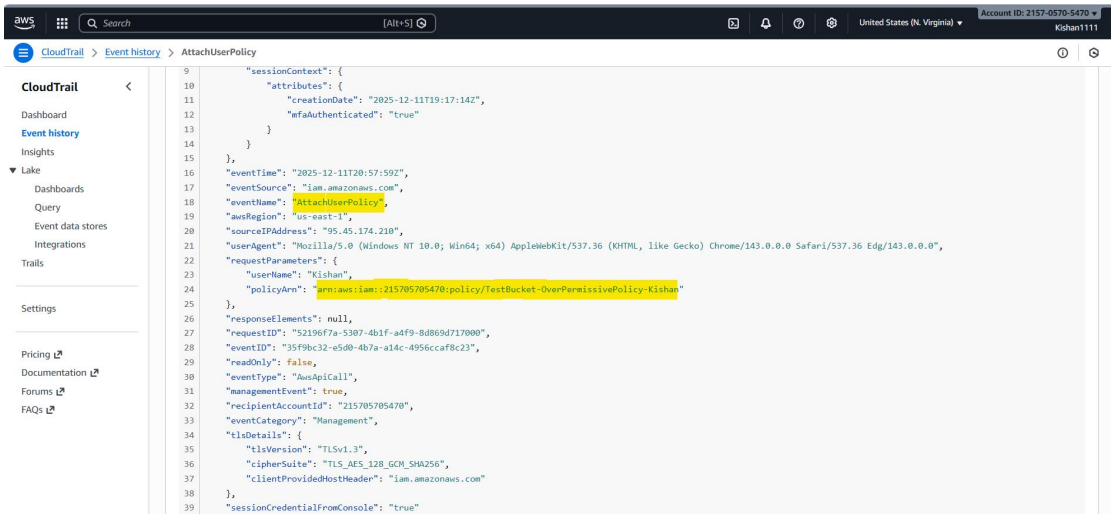
PutBucketPublicAccessBlock→ when you turned off block public access

A high- risk S3 misconfiguration
Performed by the root user
From your IP
At a specific timestamp
With the exact BPA settings disabled
Logged correctly by CloudTrail



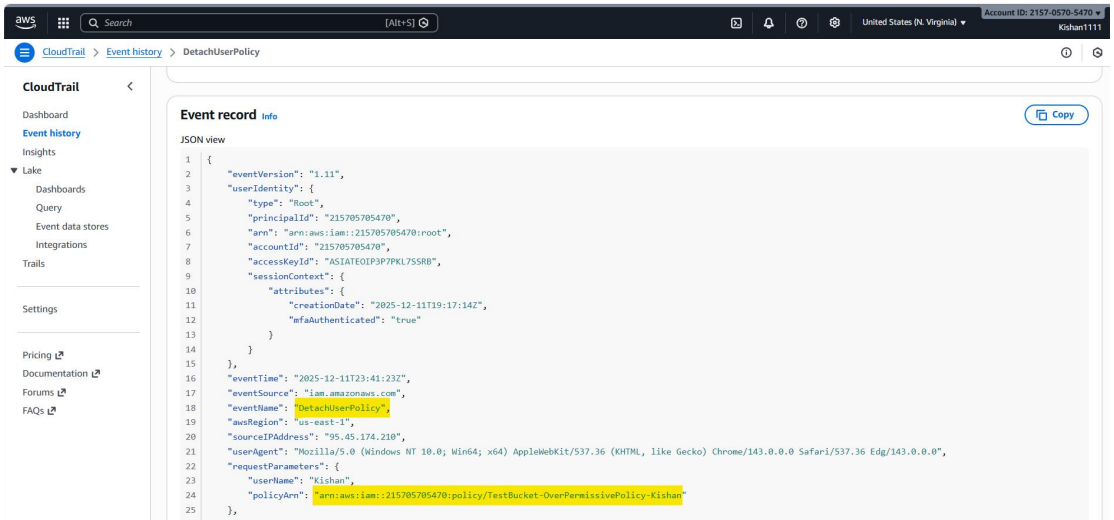
AttachUserPolicy → when you reattached the IAM policy

A misconfiguration was created
It was done by the root user
It was done from your IP
It targeted the IAM user “Kishan”
It attached an overly permissive policy
CloudTrail captured the exact timestamp and details



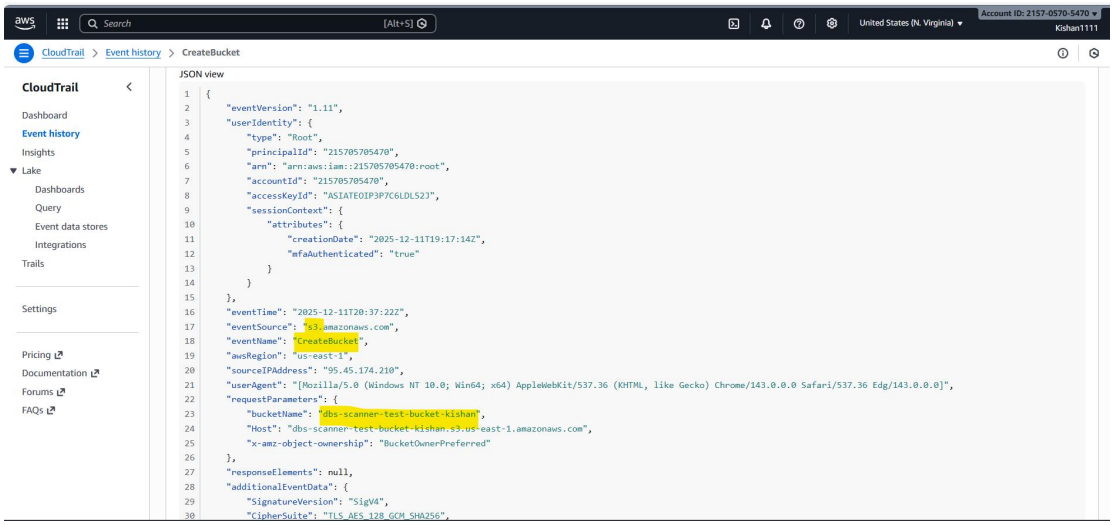
DetachUserPolicy → when you removed it

A sensitive IAM permission change
Performed by the root user
From your IP
At a specific timestamp
Targeting the IAM user “Kishan”
Removing an overly permissive policy
Logged correctly by CloudTrail



CreateBucket → when you created the test bucket

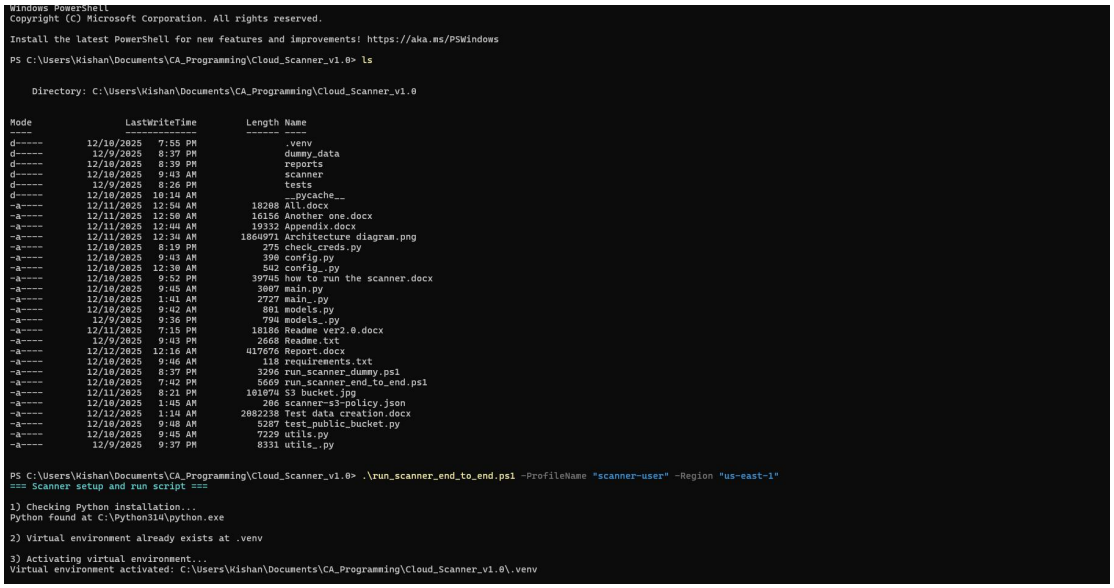
When your test bucket was created
Who created it
From which IP
Using which browser
With what ownership settings
Logged correctly by CloudTrail



Run live AWS session via Cloud Misconfigurations scanner to validate S3 cloud misconfigurations are captured

PowerShell script (automates setup and end- to- end run)

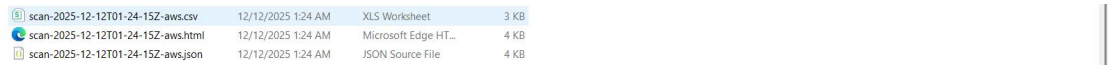
Commnad as below:
.\run_scanner_end_to_end.ps1 -ProfileName "scanner-user" -Region "us-east-1"



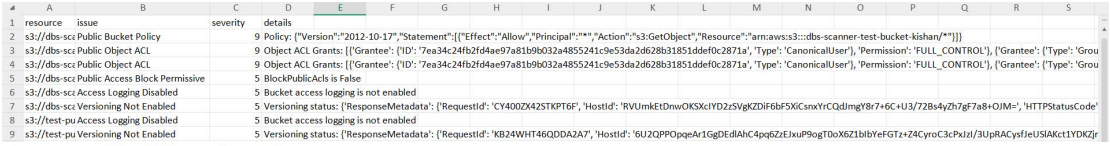

Windows PowerShell
2) Virtual environment already exists at .venv
3) Activating virtual environment...
Virtual environment activated: C:\Users\Kishan\Documents\CA_Programming\Cloud_Scanner_v1.0\.venv
4) Installing Python dependencies from requirements.txt...
Requirement already satisfied: pip in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (25.3)
Requirement already satisfied: boto3 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from -r requirements.txt (line 2)) (1.42.6)
Requirement already satisfied: botocore in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from -r requirements.txt (line 3)) (1.42.6)
Requirement already satisfied: moto in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from -r requirements.txt (line 4)) (5.1.18)
Requirement already satisfied: pytest in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from -r requirements.txt (line 5)) (9.0.2)
Requirement already satisfied: tabulate in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from -r requirements.txt (line 6)) (0.9.0)
Requirement already satisfied: beautifulsoup4 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from -r requirements.txt (line 7)) (4.14.3)
Requirement already satisfied: rich in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from -r requirements.txt (line 8)) (14.2.0)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from boto3->-r requirements.txt (line 2)) (1.0.1)
Requirement already satisfied: s3transfer<0.17.0,>=0.16.0 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from boto3->-r requirements.txt (line 2)) (0.16.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from botocore->-r requirements.txt (line 3)) (2.9.0.post0)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from botocore->-r requirements.txt (line 3)) (2.6.1)
Requirement already satisfied: six>=1.5 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from python-dateutil<3.0.0,>=2.1->botocore->-r requirements.txt (line 3)) (1.17.0)
Requirement already satisfied: cryptography>=35.0.0 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from moto->-r requirements.txt (line 4)) (46.0.3)
Requirement already satisfied: requests>=2.6 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from moto->-r requirements.txt (line 4)) (2.32.5)
Requirement already satisfied: xmltodict in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from moto->-r requirements.txt (line 4)) (1.0.2)
Requirement already satisfied: werkzeug!=2.2.0,!=2.2.1,>=0.5 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from moto->-r requirements.txt (line 4)) (3.1.4)
Requirement already satisfied: responses!=0.25.5,>=0.15.0 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from moto->-r requirements.txt (line 4)) (0.25.8)
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from moto->-r requirements.txt (line 4)) (3.1.6)
Requirement already satisfied: colorama>=0.4 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from pytest->-r requirements.txt (line 5)) (0.4.6)
Requirement already satisfied: iniconfig>=1.0.1 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from pytest->-r requirements.txt (line 5)) (2.3.0)
Requirement already satisfied: packaging>=22 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from pytest->-r requirements.txt (line 5)) (25.0)
Requirement already satisfied: pluggy<2,>=1.5 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from pytest->-r requirements.txt (line 5)) (1.6.0)
Requirement already satisfied: pygments>=2.7.2 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from pytest->-r requirements.txt (line 5)) (2.19.2)
Requirement already satisfied: soupsieve>=1.6.1 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from beautifulsoup4->-r requirements.txt (line 7)) (2.8)
Requirement already satisfied: typing-extensions>=4.0.0 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from beautifulsoup4->-r requirements.txt (line 7)) (4.15.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from rich->-r requirements.txt (line 8)) (4.0.0)
Requirement already satisfied: cffi>=2.0.0 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from cryptography>=35.0.0->moto->-r requirements.txt (line 4)) (2.0.0)
Requirement already satisfied: pycparser in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from cffi>=2.0.0->cryptography>=35.0.0->moto->-r requirements.txt (line 4)) (2.23)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from Jinja2>=2.10.1->moto->-r requirements.txt (line 4)) (3.0.3)
Requirement already satisfied: mdurl~=0.1 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from markdown-it-py>=2.2.0->rich->-r requirements.txt (line 8)) (0.1.2)
Requirement already satisfied: charset_normalizer<4,>=2 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from requests>=2.5->moto->-r requirements.txt (line 4)) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from requests>=2.5->moto->-r requirements.txt (line 4)) (3.11)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from requests>=2.5->moto->-r requirements.txt (line 4)) (2025.11.12)
Requirement already satisfied: pyyaml in c:\users\kishan\documents\ca_programming\cloud_scanner_v1.0\.venv\lib\site-packages (from responses!=0.25.5,>=0.15.0->moto->-r requirements.txt (line 4)) (6.0.3)
5) Verifying AWS CLI and credentials for profile 'scanner-user'...
Attempting to call sts get-caller-identity with profile scanner-user...
AWS identity (raw):
{ "UserId": "AIDATEOIP3P7BE3VO26HA", "Account": "215705705470", "Arn": "arn:aws:iam::215705705470:user/Kishan" }
7) Running test harness (creates temporary public bucket, runs scanner, then cleans up)...
Creating test bucket: test-public-bucket-5fee266a in region us-east-1
AWS API error: An error occurred (InvalidAccessKeyId) when calling the CreateBucket operation: The AWS Access Key Id you provided does not exist in our records.
Cleaning up test bucket and objects...
Cleanup error (manual cleanup may be required): An error occurred (InvalidAccessKeyId) when calling the ListObjectsV2 operation: The AWS Access Key Id you provided does not exist in our records.
Test harness completed successfully.
Test harness completed successfully.
8) Running full live scanner (mode aws) and saving reports...
INFO:cloud_scanner:Running in live AWS mode (profile=scanner-user region=us-east-1)
INFO:botocore.credentials:Found credentials in shared credentials file: ~/.aws/credentials
Scan summary:
- Total findings: 8
Resource
Issue
Severity
Details
s3://dbs-scanner-test-bucket-kishan
Public Bucket Policy
9
Policy: {"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":"*","Action":"s3:GetObject","Resource":"arn:aws:s3:::dbs-scanner-test-bucket-kishan/*"}]}
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file.txt
Public Object ACL
9
Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a48552 41c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file2.txt
Public Object ACL
9
Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a48552 41c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]
s3://dbs-scanner-test-bucket-kishan
Public Access Block Permissive
5
BlockPublicAcls is False
s3://dbs-scanner-test-bucket-kishan
Access Logging Disabled
5
Bucket access logging is not enabled
s3://dbs-scanner-test-bucket-kishan
Versioning Not Enabled
5
Versioning status: {'ResponseMetadata': {'RequestId': 'CYH00ZX42STKPT6F', 'HostId': 'RVUmkEtDnwOKSXcIYD2zSVgkZDiF6bF5X iCsnxYrCQdJmgY8r7+6C+U3/72Bs4yZh7g F7a8+OJM=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': 'RVUmkEtDnwOKSXcIYD2zSVgkZDiF6bF5X iCsnxYrCQdJmgY8r7+6C+U3/72Bs4yZh7g F7a8+OJM=', 'x-amz-request-id': 'CYH00ZX42STKPT6F', 'date': 'Fri, 12 Dec 2025 01:24:12 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'},
'RVUmkEtDnwOKSXcIYD2zSVgkZDiF6bF5X iCsnxYrCQdJmgY8r7+6C+U3/72Bs4yZh7g F7a8+OJM=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': 'RVUmkEtDnwOKSXcIYD2zSVgkZDiF6bF5X iCsnxYrCQdJmgY8r7+6C+U3/72Bs4yZh7g F7a8+OJM=', 'x-amz-request-id': 'CYH00ZX42STKPT6F', 'date': 'Fri, 12 Dec 2025 01:24:12 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 0}}
s3://test-public-acl-12345
Access Logging Disabled
5
Bucket access logging is not enabled
s3://test-public-acl-12345
Versioning Not Enabled
5
Versioning status: {'ResponseMetadata': {'RequestId': 'KB2HWHT46QDDA2A7', 'HostId': '6U2QPPOpqeAr1GpDEdlAhC4pq6ZzEJxuP 9ogT0oX6Z1bIbYeFGTz+Z4CyroC3cPxJzI /3UpRACysfJeUSlAKct1YDKZjr5K7', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': '6U2QPPOpqeAr1GpDEdlAhC4pq6ZzEJxuP 9ogT0oX6Z1bIbYeFGTz+Z4CyroC3cPxJzI /3UpRACysfJeUSlAKct1YDKZjr5K7', 'x-amz-request-id': 'KB2HWHT46QDDA2A7', 'date': 'Fri, 12 Dec 2025 01:24:16 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 0}}
Saved reports:
- JSON: reports\scan-2025-12-12T01-24-15Z-aws.json
- CSV: reports\scan-2025-12-12T01-24-15Z-aws.csv
- HTML: reports\scan-2025-12-12T01-24-15Z-aws.html
Scanner run completed successfully.

```
Saved reports:
- JSON:  reports\scan-2025-12-12T01-24-15Z-aws.json
- CSV:  reports\scan-2025-12-12T01-24-15Z-aws.csv
- HTML: reports\scan-2025-12-12T01-24-15Z-aws.html

Scanner run completed successfully.

9) Listing reports directory...

Name                                LastWriteTime          Length
----                                -------------          ------
scan-2025-12-12T01-24-15Z-aws.html  12/12/2025 1:24:15 AM    3421
scan-2025-12-12T01-24-15Z-aws.csv   12/12/2025 1:24:15 AM    2364
scan-2025-12-12T01-24-15Z-aws.json  12/12/2025 1:24:15 AM    3807
scan-2025-12-10T20-39-51Z-dummy.html 12/10/2025 8:39:51 PM    2824
scan-2025-12-10T20-39-51Z-dummy.csv  12/10/2025 8:39:51 PM    1851
scan-2025-12-10T20-39-51Z-dummy.json 12/10/2025 8:39:51 PM    2924
scan-2025-12-10T20-38-24Z-dummy.html 12/10/2025 8:38:24 PM    2824
scan-2025-12-10T20-38-24Z-dummy.csv  12/10/2025 8:38:24 PM    1851
scan-2025-12-10T20-38-24Z-dummy.json 12/10/2025 8:38:24 PM    2924
scan-2025-12-10T20-29-17Z-aws.html   12/10/2025 8:29:17 PM    1402
scan-2025-12-10T20-29-17Z-aws.csv    12/10/2025 8:29:17 PM     633
scan-2025-12-10T20-29-17Z-aws.json   12/10/2025 8:29:17 PM    1079
scan-2025-12-10T10-28-38Z-dummy.html 12/10/2025 10:28:38 AM  2824
scan-2025-12-10T10-28-38Z-dummy.csv  12/10/2025 10:28:38 AM  1851
scan-2025-12-10T10-28-38Z-dummy.json 12/10/2025 10:28:38 AM  2924
scan-2025-12-10T10-14-50Z-aws.html   12/10/2025 10:14:50 AM  1442
scan-2025-12-10T10-14-50Z-aws.csv    12/10/2025 10:14:50 AM   673
scan-2025-12-10T10-14-50Z-aws.json   12/10/2025 10:14:50 AM  1119
scan-2025-12-10T09-32-45Z-aws.html   12/10/2025 9:32:45 AM    698
scan-2025-12-10T09-32-45Z-aws.csv    12/10/2025 9:32:45 AM     33
scan-2025-12-10T09-32-45Z-aws.json   12/10/2025 9:32:45 AM    206
scan-2025-12-10T02-08-45Z-aws.html   12/10/2025 2:08:45 AM    698
scan-2025-12-10T02-08-45Z-aws.csv    12/10/2025 2:08:45 AM     33
scan-2025-12-10T02-08-45Z-aws.json   12/10/2025 2:08:45 AM    206
scan-2025-12-10T02-07-54Z-aws.html   12/10/2025 2:07:54 AM    698
scan-2025-12-10T02-07-54Z-aws.csv    12/10/2025 2:07:54 AM     33
scan-2025-12-10T02-07-54Z-aws.json   12/10/2025 2:07:54 AM    206
scan-2025-12-10T02-05-56Z-aws.csv    12/10/2025 2:05:56 AM     33
scan-2025-12-10T02-05-56Z-aws.html   12/10/2025 2:05:56 AM    698
scan-2025-12-10T02-05-56Z-aws.json   12/10/2025 2:05:56 AM    206
scan-2025-12-10T02-02-42Z-aws.html   12/10/2025 2:02:42 AM    698
scan-2025-12-10T02-02-42Z-aws.csv    12/10/2025 2:02:42 AM     33
scan-2025-12-10T02-02-42Z-aws.json   12/10/2025 2:02:42 AM    206
scan-2025-12-10T02-01-58Z-dummy.html 12/10/2025 2:01:58 AM   3436
scan-2025-12-10T02-01-58Z-dummy.csv  12/10/2025 2:01:58 AM   1698
scan-2025-12-10T02-01-58Z-dummy.json 12/10/2025 2:01:58 AM   3447
scan-2025-12-10T02-00-57Z-aws.csv    12/10/2025 2:00:57 AM     33
scan-2025-12-10T02-00-57Z-aws.html   12/10/2025 2:00:57 AM    698
scan-2025-12-10T02-00-57Z-aws.json   12/10/2025 2:00:57 AM    206
scan-2025-12-10T01-22-58Z-aws.html   12/10/2025 1:22:58 AM   1029
scan-2025-12-10T01-22-58Z-aws.csv    12/10/2025 1:22:58 AM    311
scan-2025-12-10T01-22-58Z-aws.json   12/10/2025 1:22:58 AM    605
scan-2025-12-10T01-03-12Z-aws.html   12/10/2025 1:03:12 AM    930
scan-2025-12-10T01-03-12Z-aws.csv    12/10/2025 1:03:12 AM    218
scan-2025-12-10T01-03-12Z-aws.json   12/10/2025 1:03:12 AM    806


Open the latest HTML report in your browser to review findings.

Script finished.
(.venv) PS C:\Users\Kishan\Documents\CA_Programming\Cloud_Scanner_v1.0>
```

Exported reports as below



scan-2025-12-12T01-24-15Z-aws.csv	12/12/2025 1:24 AM	XLS Worksheet	3 KB
scan-2025-12-12T01-24-15Z-aws.html	12/12/2025 1:24 AM	Microsoft Edge HT...	4 KB
scan-2025-12-12T01-24-15Z-aws.json	12/12/2025 1:24 AM	JSON Source File	4 KB



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	resource	issue	severity	details															
2	s3://dbs-scc	Public Bucket Policy	9	Policy: {"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":"*","Action":"s3:GetObject","Resource":"arn:aws:s3:::dbs-scanner-test-bucket-kishan/*"}]}															
3	s3://dbs-scc	Public Object ACL	9	Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Grou															
4	s3://dbs-scc	Public Object ACL	9	Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Grou															
5	s3://dbs-scc	Public Access Block Permissive	5	BlockPublicAcls is False															
6	s3://dbs-scc	Access Logging Disabled	5	Bucket access logging is not enabled															
7	s3://dbs-scc	Versioning Not Enabled	5	Versioning status: {'ResponseMetadata': {'RequestId': 'CY400ZX42STKPT6F', 'HostId': 'RVUmkEtDnwOKSXcIYD2zSVgKZDiF6bF5XiCsnxYrCQdJmgY8r7+6C+U3/72Bs4yZh7gF7a8+OJM=', 'HTTPStatusCode'															
8	s3://test-pu	Access Logging Disabled	5	Bucket access logging is not enabled															
9	s3://test-pu	Versioning Not Enabled	5	Versioning status: {'ResponseMetadata': {'RequestId': 'KB24WHT46QDDA2A7', 'HostId': '6U2QPPOpqeAr1GgDEdlAhC4pq6ZzEJxuP9ogT0oX6Z1blbYeFGTz+Z4CyroC3cPxJzI/3UpRACysfJeUSlAKct1YDKZjr															

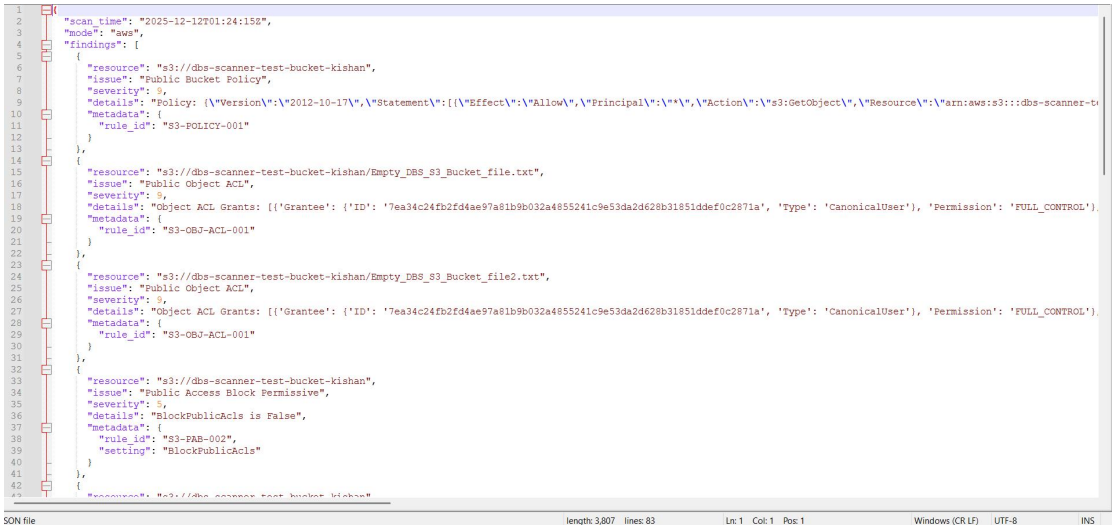
Scan Report - 2025-12-12T01:24:15Z - mode: aws

Total findings: 8

Metadata:

- profile: scanner-user
- region: us-east-1

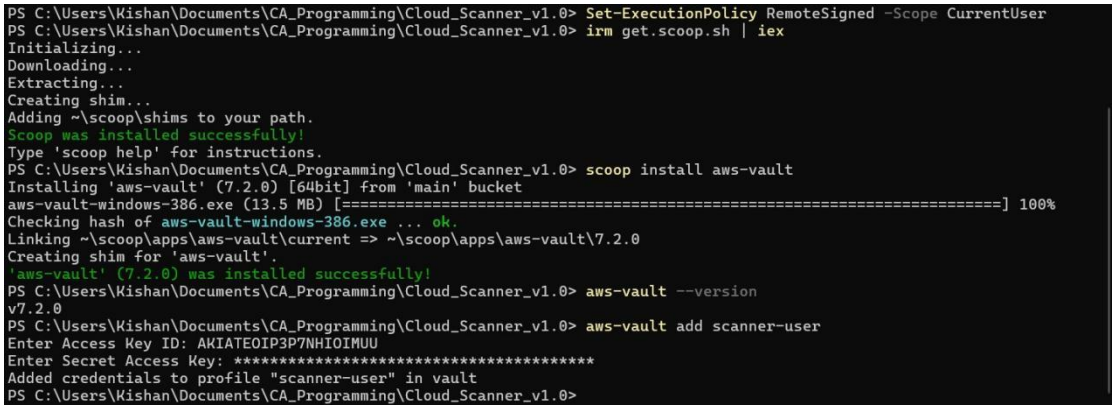
Resource	Issue	Severity	Details
s3://dbs-scanner-test-bucket-kishan	Public Bucket Policy	9	`Policy: {"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":"*","Action":"s3:GetObject","Resource":"arn:aws:s3:::dbs-scanner-test-bucket-kishan/*"}]}`
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file.txt	Public Object ACL	9	`Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]`
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file2.txt	Public Object ACL	9	`Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]`
s3://dbs-scanner-test-bucket-kishan	Public Access Block Permissive	5	`BlockPublicAcls is False`
s3://dbs-scanner-test-bucket-kishan	Access Logging Disabled	5	`Bucket access logging is not enabled`
s3://dbs-scanner-test-bucket-kishan	Versioning Not Enabled	5	`Versioning status: {'ResponseMetadata': {'RequestId': 'CY400ZX42STKPT6F', 'HostId': 'RVUmkEtDnwOKSXcIYD2zSVgKZDiF6bF5XiCsnxYrCQdJmgY8r7+6C+U3/72Bs4yZh7gF7a8+OJM=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': 'RVUmkEtDnwOKSXcIYD2zSVgKZDiF6bF5XiCsnxYrCQdJmgY8r7+6C+U3/72Bs4yZh7gF7a8+OJM=', 'x-amz-request-id': 'CY400ZX42STKPT6F', 'date': 'Fri, 12 Dec 2025 01:24:12 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 0}}`
s3://test-public-acl-12345	Access Logging Disabled	5	`Bucket access logging is not enabled`
s3://test-public-acl-12345	Versioning Not Enabled	5	`Versioning status: {'ResponseMetadata': {'RequestId': 'KB24WHT46QDDA2A7', 'HostId': '6U2QPPOpqeAr1GgDEdlAhC4pq6ZzEJxuP9ogT0oX6Z1blbYeFGTz+Z4CyroC3cPxJzI/3UpRACysfJeUSlAKct1YDKZjr5K7', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': '6U2QPPOpqeAr1GgDEdlAhC4pq6ZzEJxuP9ogT0oX6Z1blbYeFGTz+Z4CyroC3cPxJzI/3UpRACysfJeUSlAKct1YDKZjr5K7', 'x-amz-request-id': 'KB24WHT46QDDA2A7', 'date': 'Fri, 12 Dec 2025 01:24:16 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 0}}`



AWS Scanner runs with AWS vault security encryption

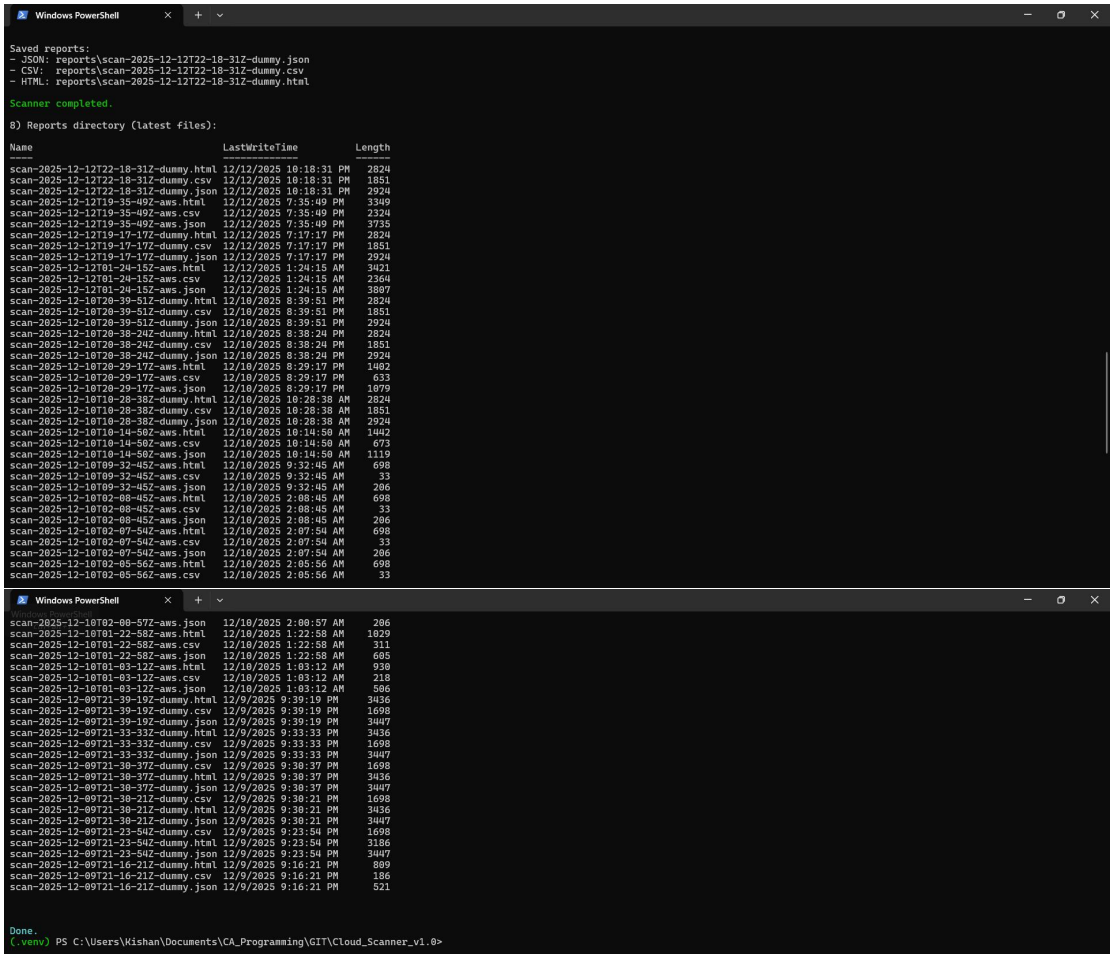
AWS Vault is an open-source tool developed at 99designs to securely store and manage Amazon Web Services credentials in development environments. It encrypts your long-term IAM credentials, typically an Access Key and a Secret Key, and generates short-term credentials that can be exposed to your shell and applications.Using AWS Vault limits the risk of malicious applications or dependencies gaining unauthorized access to your AWS account through engineering credentials. In addition, the likelihood of accidentally storing IAM credentials in version control (GitHub, Bitbucket, etc.) reduces significantly.

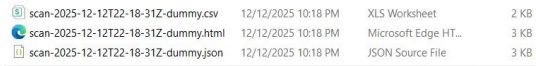
Doppler (2024) AWS Vault: A Secure Way to Store and Access AWS Credentials. Available at: https://www.doppler.com/guides/aws-guides/aws-vault
(Accessed: [12-12-2025]).



AWS Scanner runs with dummy data

.\run_scanner_dummy.ps1





	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	resource	issue	severity	details															
2	s3://public-read-bucket	Public ACL	9	ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]															
3	s3://public-read-write-bucket	Public ACL	9	ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'WRITE'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/gl																
4	s3://authenticated-users-bucket	Public ACL	9	ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AuthenticatedUsers'}, 'Permission': 'READ'}]															
5	s3://mixed-grants-bucket	Public ACL	9	ACL Grants: [{'Grantee': {'Type': 'CanonicalUser', 'ID': 'user-123'}, 'Permission': 'READ'}, {'Grantee': {'Type': 'CanonicalUser', 'ID': 'user-456'}, 'Permission': 'WRITE'}, {'Grantee': {'Type': 'Group', 'URI': '																
6	s3://large-grants-bucket	Public ACL	9	ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/glo																
7	s3://public-multiple-perms-bucket	Public ACL	9	ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/glo																

Scan Report - 2025-12-12T22:18:31Z - mode: dummy

Total findings: 6

Metadata:

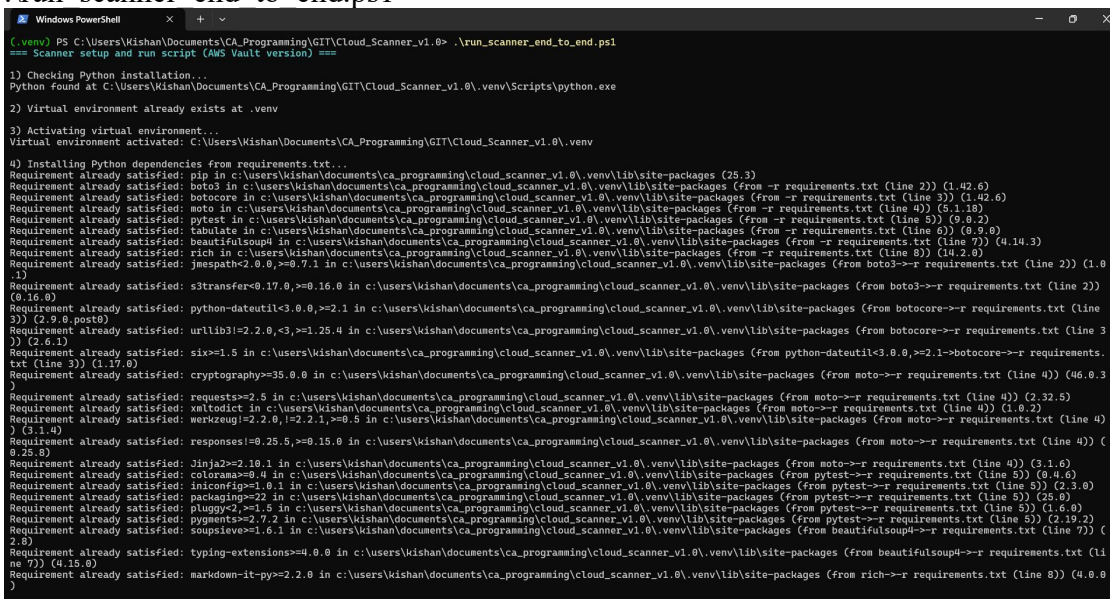
- source_file: dummy_data/sample_aws_resources.json

Resource	Issue	Severity	Details
s3://public-read-bucket	Public ACL	9	`ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]`
s3://public-read-write-bucket	Public ACL	9	`ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'WRITE'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]`
s3://authenticated-users-bucket	Public ACL	9	`ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AuthenticatedUsers'}, 'Permission': 'READ'}]`
s3://mixed-grants-bucket	Public ACL	9	`ACL Grants: [{'Grantee': {'Type': 'CanonicalUser', 'ID': 'user-123'}, 'Permission': 'READ'}, {'Grantee': {'Type': 'CanonicalUser', 'ID': 'user-456'}, 'Permission': 'WRITE'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]`
s3://large-grants-bucket	Public ACL	9	`ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AuthenticatedUsers'}, 'Permission': 'READ'}, {'Grantee': {'Type': 'CanonicalUser', 'ID': 'user-001'}, 'Permission': 'WRITE'}, {'Grantee': {'Type': 'CanonicalUser', 'ID': 'user-002'}, 'Permission': 'READ'}]`
s3://public-multiple-perms-bucket	Public ACL	9	`ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'WRITE'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AuthenticatedUsers'}, 'Permission': 'READ'}]`

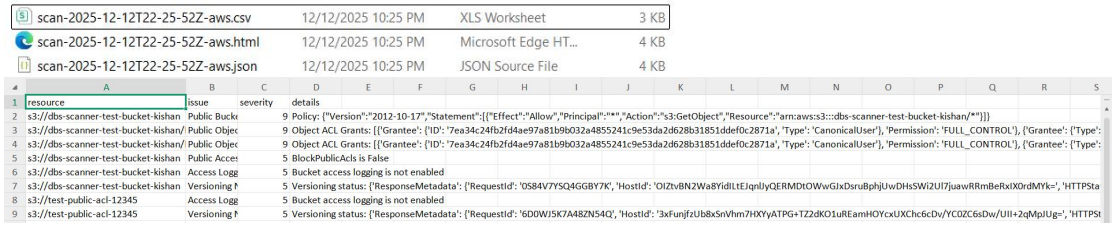
```
{
  "scan_time": "2025-12-12T22:18:31Z",
  "mode": "dummy",
  "findings": [
    {
      "resource": "s3://public-read-bucket",
      "issue": "Public ACL",
      "severity": 9,
      "details": "ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]",
      "metadata": {
        "rule_id": "S3-ACL-001"
      }
    },
    {
      "resource": "s3://public-read-write-bucket",
      "issue": "Public ACL",
      "severity": 9,
      "details": "ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'WRITE'}, {'Grantee': {'Type': 'Group', 'URI':
      "metadata": {
        "rule_id": "S3-ACL-001"
      }
    },
    {
      "resource": "s3://authenticated-users-bucket",
      "issue": "Public ACL",
      "severity": 9,
      "details": "ACL Grants: [{'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AuthenticatedUsers'}, 'Permission': 'READ'}]",
      "metadata": {
        "rule_id": "S3-ACL-001"
      }
    },
    {
      "resource": "s3://mixed-grants-bucket",
      "issue": "Public ACL",
      "severity": 9,
      "details": "ACL Grants: [{'Grantee': {'Type': 'CanonicalUser', 'ID': 'user-123'}, 'Permission': 'READ'}, {'Grantee': {'Type': 'CanonicalUser', 'ID': 'user-456'}, 'Permission'
      "metadata": {
        "rule_id": "S3-ACL-001"
      }
    },
    {
      "resource": "s3://large-grants-bucket",
```


AWS Scanner runs with AWS vault security encryption

.\run_scanner_end_to_end.ps1




Windows PowerShell
AWS identity (raw):
{ "UserId": "AIDATEOIP3P7BE3VO26HA", "Account": "215705705470", "Arn": "arn:aws:iam::215705705470:user/Kishan" }
6) Running test harness (temporary public bucket)...
Creating test bucket: test-public-bucket-354f77d1 in region us-east-1
Applying public ACL to bucket
AWS API error: An error occurred (AccessDenied) when calling the PutBucketAcl operation: User: arn:aws:iam::215705705470:user/Kishan is not authorized to perform: s3:PutBucketAcl on resource: "arn:aws:s3:::test-public-bucket-354f77d1" because public ACLs are prevented by the BlockPublicAcls setting in S3 Block Public Access.
Cleaning up test bucket and objects...
Cleanup complete.
Test harness completed successfully.
7) Running full live scanner (mode aws) and saving reports...
INFO:cloud_scanner:Running in live AWS mode (region=eu-west-1)
INFO:botocore.credentials:Found credentials in environment variables.
Scan summary:
- Total findings: 8
Resource
Issue
Severity
Details
s3://dbs-scanner-test-bucket-kishan
Public Bucket Policy
9
Policy: {"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":"*","Action":"s3:GetObject","Resource":"arn:aws:s3:::dbs-scanner-test-bucket-kishan/*"}]}
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file.txt
Public Object ACL
9
Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file2.txt
Public Object ACL
9
Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]
s3://dbs-scanner-test-bucket-kishan
Public Access Block Permissive
5
BlockPublicAcls is False
s3://dbs-scanner-test-bucket-kishan
Access Logging Disabled
5
Bucket access logging is not enabled
s3://dbs-scanner-test-bucket-kishan
Versioning Not Enabled
5
Versioning status: {'ResponseMetadata': {'RequestId': '0S84V7YSQ4GGBY7K', 'HostId': 'OIZtvBN2Wa8YidILtEJqnlJyQERMDtOWwGJxDsruBphjUwDHsSWi2Ul7juawRRmBeRxIX0rdMYk=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': 'OIZtvBN2Wa8YidILtEJqnlJyQERMDtOWwGJxDsruBphjUwDHsSWi2Ul7juawRRmBeRxIX0rdMYk=', 'x-amz-request-id': '0S84V7YSQ4GGBY7K', 'date': 'Fri, 12 Dec 2025 22:25:44 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 1}}
s3://test-public-acl-12345
Access Logging Disabled
5
Bucket access logging is not enabled
s3://test-public-acl-12345
Versioning Not Enabled
5
Versioning status: {'ResponseMetadata': {'RequestId': '6D0WJ5K7A48ZN54Q', 'HostId':
Resource
Issue
Severity
Details
s3://dbs-scanner-test-bucket-kishan
Public Bucket Policy
9
Policy: {"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":"*","Action":"s3:GetObject","Resource":"arn:aws:s3:::dbs-scanner-test-bucket-kishan/*"}]}
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file.txt
Public Object ACL
9
Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file2.txt
Public Object ACL
9
Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]
s3://dbs-scanner-test-bucket-kishan
Public Access Block Permissive
5
BlockPublicAcls is False
s3://dbs-scanner-test-bucket-kishan
Access Logging Disabled
5
Bucket access logging is not enabled
s3://dbs-scanner-test-bucket-kishan
Versioning Not Enabled
5
Versioning status: {'ResponseMetadata': {'RequestId': '0S84V7YSQ4GGBY7K', 'HostId': 'OIZtvBN2Wa8YidILtEJqnlJyQERMDtOWwGJxDsruBphjUwDHsSWi2Ul7juawRRmBeRxIX0rdMYk=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': 'OIZtvBN2Wa8YidILtEJqnlJyQERMDtOWwGJxDsruBphjUwDHsSWi2Ul7juawRRmBeRxIX0rdMYk=', 'x-amz-request-id': '0S84V7YSQ4GGBY7K', 'date': 'Fri, 12 Dec 2025 22:25:44 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 1}}
s3://test-public-acl-12345
Access Logging Disabled
5
Bucket access logging is not enabled
s3://test-public-acl-12345
Versioning Not Enabled
5
Versioning status: {'ResponseMetadata': {'RequestId': '6D0WJ5K7A48ZN54Q', 'HostId': '3xFunjfzUb8xSnVhm7HXYyATPG+TZ2dKO1uREamHOYcxUXChc6cDv/YC0ZC6sDw/UII+2qMpJUg=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': '3xFunjfzUb8xSnVhm7HXYyATPG+TZ2dKO1uREamHOYcxUXChc6cDv/YC0ZC6sDw/UII+2qMpJUg=', 'x-amz-request-id': '6D0WJ5K7A48ZN54Q', 'date': 'Fri, 12 Dec 2025 22:25:51 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 1}}
Saved reports:
- JSON: reports\scan-2025-12-12T22-25-52Z-aws.json
- CSV: reports\scan-2025-12-12T22-25-52Z-aws.csv
- HTML: reports\scan-2025-12-12T22-25-52Z-aws.html
Scanner run completed successfully.
Saved reports:
- JSON: reports\scan-2025-12-12T22-25-52Z-aws.json
- CSV: reports\scan-2025-12-12T22-25-52Z-aws.csv
- HTML: reports\scan-2025-12-12T22-25-52Z-aws.html
Scanner run completed successfully.
8) Listing reports directory...
Name
LastWriteTime
Length
scan-2025-12-12T22-25-52Z-aws.html 12/12/2025 10:25:52 PM 3349
scan-2025-12-12T22-25-52Z-aws.csv 12/12/2025 10:25:52 PM 2324
scan-2025-12-12T22-25-52Z-aws.json 12/12/2025 10:25:52 PM 3735
scan-2025-12-12T22-18-31Z-dummy.html 12/12/2025 10:18:31 PM 2824
scan-2025-12-12T22-18-31Z-dummy.csv 12/12/2025 10:18:31 PM 1851
scan-2025-12-12T22-18-31Z-dummy.json 12/12/2025 10:18:31 PM 2924
scan-2025-12-09T21-16-21Z-dummy.json 12/9/2025 9:16:21 PM 321
Open the latest HTML report in your browser to review findings.
Script finished.
(.venv) PS C:\Users\Kishan\Documents\CA_Programming\GIT\Cloud_Scanner_v1.0>


scan-2025-12-12T22-25-52Z-aws.csv 12/12/2025 10:25 PM XLS Worksheet 3 KB
scan-2025-12-12T22-25-52Z-aws.html 12/12/2025 10:25 PM Microsoft Edge HT... 4 KB
scan-2025-12-12T22-25-52Z-aws.json 12/12/2025 10:25 PM JSON Source File 4 KB
resource issue severity details
s3://dbs-scanner-test-bucket-kishan Public Bucke 9 Policy: {"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":"*","Action":"s3:GetObject","Resource":"arn:aws:s3:::dbs-scanner-test-bucket-kishan/*"}]}
s3://dbs-scanner-test-bucket-kishan/ Public Objec 9 Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type':
s3://dbs-scanner-test-bucket-kishan/ Public Objec 9 Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type':
s3://dbs-scanner-test-bucket-kishan Public Acce 5 BlockPublicAcls is False
s3://dbs-scanner-test-bucket-kishan Access Logg 5 Bucket access logging is not enabled
s3://dbs-scanner-test-bucket-kishan Versioning N 5 Versioning status: {'ResponseMetadata': {'RequestId': '0S84V7YSQ4GGBY7K', 'HostId': 'OIZtvBN2Wa8YidILtEJqnlJyQERMDtOWwGJxDsruBphjUwDHsSWi2Ul7juawRRmBeRxIX0rdMYk=', 'HTTPSta
s3://test-public-acl-12345 Access Logg 5 Bucket access logging is not enabled
s3://test-public-acl-12345 Versioning N 5 Versioning status: {'ResponseMetadata': {'RequestId': '6D0WJ5K7A48ZN54Q', 'HostId': '3xFunjfzUb8xSnVhm7HXYyATPG+TZ2dKO1uREamHOYcxUXChc6cDv/YC0ZC6sDw/UII+2qMpJUg=', 'HTTPSt

Scan Report - 2025-12-12T22:25:52Z - mode: aws

Total findings: 8

Metadata:

- region: eu-west-1

Resource	Issue	Severity	Details
s3://dbs-scanner-test-bucket-kishan	Public Bucket Policy	9	Policy: {"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":"*","Action":"s3:GetObject","Resource":"arn:aws:s3:::dbs-scanner-test-bucket-kishan/*"}]}
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file.txt	Public Object ACL	9	Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]
s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file2.txt	Public Object ACL	9	Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}, {'Grantee': {'Type': 'Group', 'URI': 'http://acs.amazonaws.com/groups/global/AllUsers'}, 'Permission': 'READ'}]
s3://dbs-scanner-test-bucket-kishan	Public Access Block Permissive	5	BlockPublicAcls is False
s3://dbs-scanner-test-bucket-kishan	Access Logging Disabled	5	Bucket access logging is not enabled
s3://dbs-scanner-test-bucket-kishan	Versioning Not Enabled	5	Versioning status: {'ResponseMetadata': {'RequestId': '0S84V7YSQ4GGBY7K', 'HostId': 'OIZtvBN2WaBYidLltE3qnl3yQERMDtOWwG3xDsruBphjUwDHsSWl2U17juawRRmBeRxIX0rdMYk=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': 'OIZtvBN2WaBYidLltE3qnl3yQERMDtOWwG3xDsruBphjUwDHsSWl2U17juawRRmBeRxIX0rdMYk=', 'x-amz-request-id': '0S84V7YSQ4GGBY7K', 'date': 'Fri, 12 Dec 2025 22:25:44 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 1}}
s3://test-public-acl-12345	Access Logging Disabled	5	Bucket access logging is not enabled
s3://test-public-acl-12345	Versioning Not Enabled	5	Versioning status: {'ResponseMetadata': {'RequestId': '6DNJ5K7A48ZN54Q', 'HostId': '3xFunjfzUb8xSnYhm7HXYyATPG+TZ2dKO1uREamMDYcxUKChc6cDv/YC0ZC6sDw/UII+2qMpJUg=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': '3xFunjfzUb8xSnYhm7HXYyATPG+TZ2dKO1uREamMDYcxUKChc6cDv/YC0ZC6sDw/UII+2qMpJUg=', 'x-amz-request-id': '6DNJ5K7A48ZN54Q', 'date': 'Fri, 12 Dec 2025 22:25:51 GMT', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 1}}

scan-2025-12-12T22-25-52Z-aws.json

C:\Users\Kishan\Documents\CA_Programming\GIT\Cloud_Scanner_v1.0\reports\scan-2025-12-12T22-25-52Z-aws.json


```
{
  "scan_time": "2025-12-12T22:25:52Z",
  "mode": "aws",
  "findings": [
    {
      "resource": "s3://dbs-scanner-test-bucket-kishan",
      "issue": "Public Bucket Policy",
      "severity": 9,
      "details": "Policy: {\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"s3:GetObject\",\"Resource\":\"arn:aws:s3:::dbs
      "metadata": {
        "rule_id": "S3-POLICY-001"
      }
    },
    {
      "resource": "s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file.txt",
      "issue": "Public Object ACL",
      "severity": 9,
      "details": "Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL
      "metadata": {
        "rule_id": "S3-OBJ-ACL-001"
      }
    },
    {
      "resource": "s3://dbs-scanner-test-bucket-kishan/Empty_DBS_S3_Bucket_file2.txt",
      "issue": "Public Object ACL",
      "severity": 9,
      "details": "Object ACL Grants: [{'Grantee': {'ID': '7ea34c24fb2fd4ae97a81b9b032a4855241c9e53da2d628b31851ddef0c2871a', 'Type': 'CanonicalUser'}, 'Permission': 'FULL
      "metadata": {
        "rule_id": "S3-OBJ-ACL-001"
      }
    },
    {
      "resource": "s3://dbs-scanner-test-bucket-kishan",
      "issue": "Public Access Block Permissive",
      "severity": 5,
      "details": "BlockPublicAcls is False",
      "metadata": {
```


Git hub link - https://github.com/KishanDBS/Programming_CA2.git

6. Conclusion

This cloud misconfiguration scanner is a well- structured, realistic example of an AWS S3 security tool. It supports both offline and live scanning, uses temporary credentials for safety, and produces rich outputs and reports suitable for security reviews and compliance evidence. The modular design, test coverage, and automation scripts show a strong focus on professional realism and maintainability.

8. References

AWS Documentation

Amazon Web Services (AWS) 2025, *Amazon S3: Security Best Practices*, AWS Documentation. Available at: https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-best-practices.html (Accessed: 10 November 2025).

Amazon Web Services (AWS) 2025, *Amazon S3 Block Public Access*, AWS Documentation. Available at: https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-control-block-public-access.html (Accessed: 10 November 2025).

Amazon Web Services (AWS) 2025, *Amazon S3 Bucket Policies*, AWS Documentation. Available at: https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-policies.html (Accessed: 10 November 2025).

Amazon Web Services (AWS) 2025, *Amazon S3 Server-Side Encryption*, AWS Documentation. Available at: https://docs.aws.amazon.com/AmazonS3/latest/userguide/serv-side-encryption.html (Accessed: 10 November 2025).

Amazon Web Services (AWS) 2025, *AWS CloudTrail User Guide*, AWS Documentation. Available at: https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html (Accessed: 10 November 2025).

Amazon Web Services (AWS) 2025, *AWS IAM Policies and Permissions*, AWS Documentation. Available at: https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html (Accessed: 10 November 2025).

Python & Libraries

Python Software Foundation 2025, *Dataclasses — Python 3 Documentation*. Available at: https://docs.python.org/3/library/dataclasses.html (Accessed: 10 November 2025).

Python Software Foundation 2025, *json — JSON Encoder and Decoder*. Available at: https://docs.python.org/3/library/json.html (Accessed: 10 November 2025).

Amazon Web Services (AWS) 2025, *boto3: AWS SDK for Python*. Available at: https://boto3.amazonaws.com/v1/documentation/api/latest/index.html (Accessed: 10 November 2025).

Git, GitHub & Git LFS

GitHub 2025, *Git Large File Storage (LFS) Documentation*. Available at: https://docs.github.com/en/repositories/working-with-files/managing-large-files/about-git-large-file-storage (Accessed: 10 November 2025).

Git SCM 2025, *Git Documentation*. Available at: https://git-scm.com/doc (Accessed: 10 November 2025).

GitHub 2025, *GitHub Repository: Programming_CA2*. Available at: https://github.com/KishanDBS/Programming_CA2 (Accessed: 10 November 2025).

AWS CLI & Tools

Amazon Web Services (AWS) 2025, *AWS CLI Command Reference*. Available at: https://docs.aws.amazon.com/cli/latest/index.html (Accessed: 10 November 2025).

Amazon Web Services (AWS) 2025, *AWS Vault – Secure Access to AWS*. Available at: https://github.com/99designs/aws-vault (Accessed: 10 November 2025).

Cloud Security Best Practices

OWASP 2025, *OWASP Cloud Security Guidelines*. Available at: https://owasp.org/www-project-cloud-security/ (Accessed: 10 November 2025).

Center for Internet Security (CIS) 2025, *CIS Amazon Web Services Foundations Benchmark*. Available at: https://www.cisecurity.org/benchmark/amazon_web_services (Accessed: 10 November 2025).