Hybrid Cloud Security Auditing: Enhancing Posture Through Integrated Use of Native and Open-Source CSPM Tools

Implementation Manual for UCSA Framework

This implementation manual provides a detailed step by step guide on how the Unified Cloud Security Audit (UCSA) framework was executed, detailing the environment setup, CSPM tool deployment, data export, integration with Power BI for analysis and the automation process.

1. Setup Prerequisites

1.1 Tools and Services

Category	Item
Cloud & Access	- Azure for students' subscription with security admin permissions assigned. - I used my students email to create one then my credits ran out mid-way so I moved the deployments to a paid subscription using my personal studying email

Environment & Applications	- Visual studio code installed. It should also have the following extensions:
	  o Azure environment and its CLI installed and configured
	  o Azure Bicep extensions
	  o Python 3.10+ installed
	  o Have Docker installed and running.
	  o Poetry for python package management
	- Microsoft Defender for Cloud enabled
	- Docker Desktop installed and running
	- Power BI desktop application installed
	- Microsoft Excel to open the CSV files generated by the CSPM tools

2. Environment Setup

- I logged in to Azure portal from VS Code to use the terminal to handle deployments
- I created a resource group in azure

```
PS C:\Users\Administrator\Desktop\theisis-lab> az group create --name cspm-rg --location
"South Africa North"
```

3. Implementation

The process flow below details how the whole implementation was achieved.

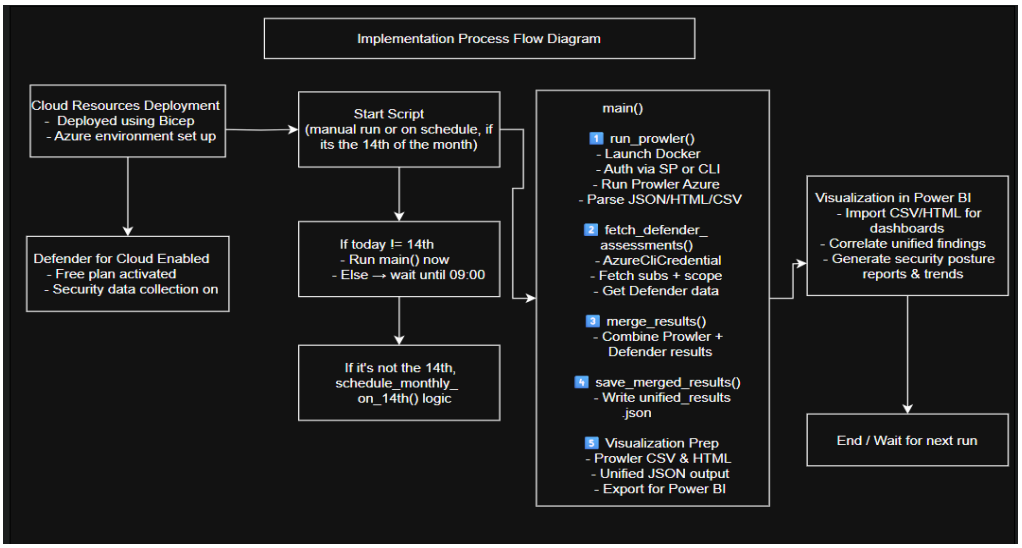

Figure 1: Implementation Flow Process

4. Resources Deployment

Bicep modules with all resources, and the main-bicep file that called all other modules. I deployed it to Azure. The bicep template deploys: 2 virtual machine, 2 storage accounts, 1 App service, 2 SQL database, appropriate NSGs and diagnostic settings

```
az deployment group create --resource-group cspm-rg \
  --template-file main.bicep \
  --parameters "@main.parameters.json"
```

5. Configuration of Microsoft Defender for Cloud

From the Microsoft Defender for Cloud section in Azure portal, I enabled Standard plan for all resource types. I also activated regulatory compliance and enabled CIS and NIST policies. It took 60 + minutes for the findings to be populated under recommendations section and for the secure score to be calculated.

.

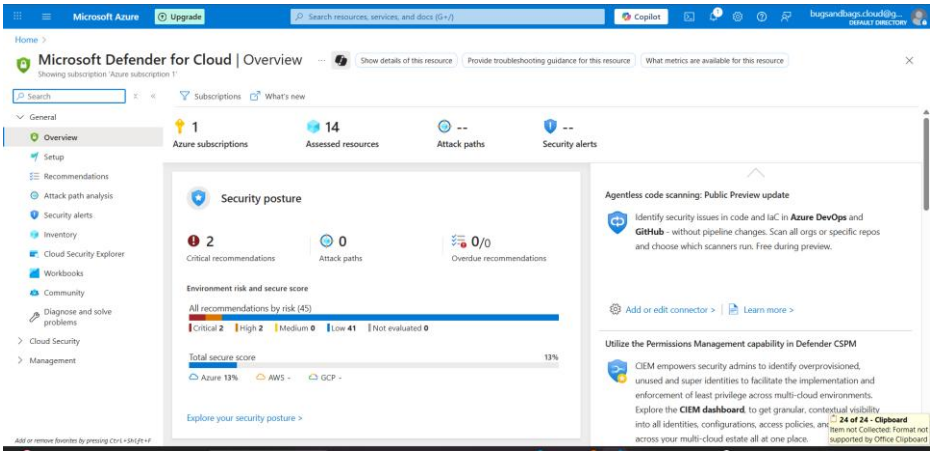

Figure 2: Microsoft Defender Secure Score

6. Setting up and running Prowler

- Install Docker for Desktop and authenticate into it. Prowler is run from a dockerized container to have a simplified integration process without the endless dependencies' installation, to make it lightweight and portable. Docker is mounted to drive so that files are saved to the local device.
- Running Prowler for Azure and saving outputs in CSV and HTML formats

```
-   subprocess.run([
-       "docker", "run", "--rm",
-       "--env-file", ".env",
-       "-v", f"{output_dir}:/prowler/output",
-       "prowlercloud/prowler:latest",
-       "azure",
-       "--sp-env-auth",
-       "--output-formats", "csv", "html", "json-ocsf",
-       "--output-directory", "/prowler/output"
-   ],
```

7. Script deployment: The scripty unified_cspm.py is deployed and runs 4 phases:
   a) Installation & running of Prowler
   b) Pulling defender for Cloud assessment results using Azure SDKs

c) Merging the results to unified JSON while still preserving the individual CSV files
d) Automating the scans

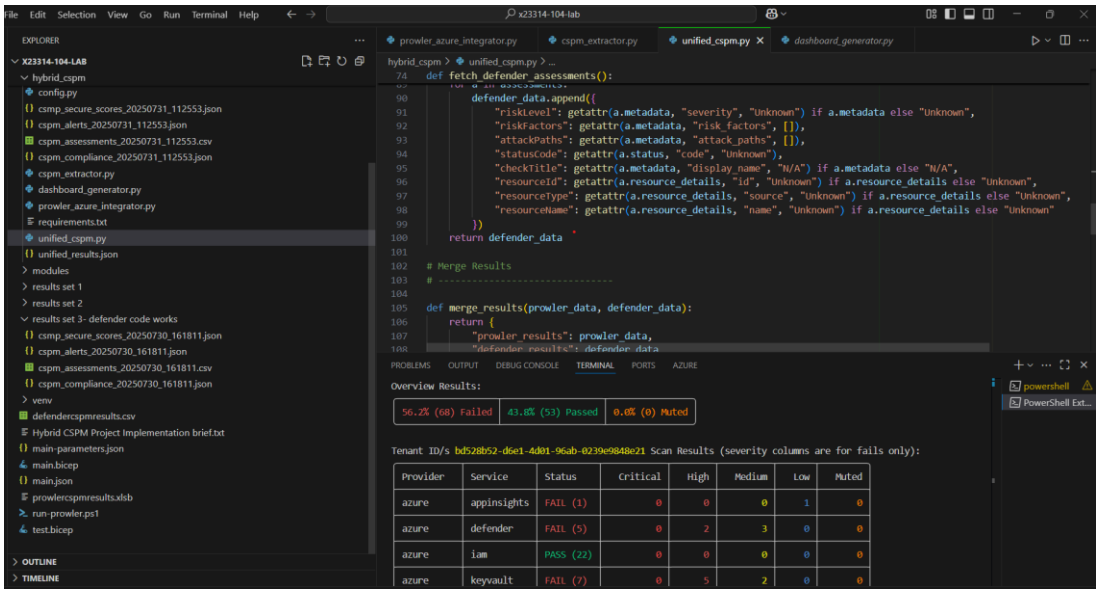

Figure 3: Script Running Successfully

8. Exporting of Results

The results from both CSPM tools are exported in CSV version to allow for further results evaluation. The results are stored on the local machine.

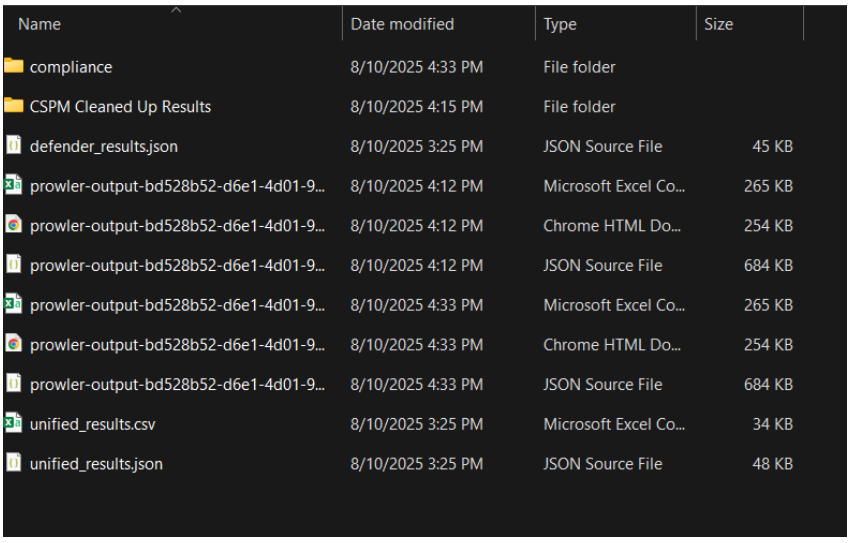

Figure 4: Hybrid CSPM Generated Results

9. Importing to Power BI Desktop and visualization

The results are cleaned up and added to the CSPM cleaned up results folder. I used two approaches:

- A manual review of the generated CSV results of both MDC and prowler results to inspect their quality of results, overlaps, gaps and strong suits of each tool
- I also cleaned up the CSV files to clean up the extra fields that can be considered padding for the file. I stripped those and used the cleaned up files to visualize on create the PowerBI visualizations.

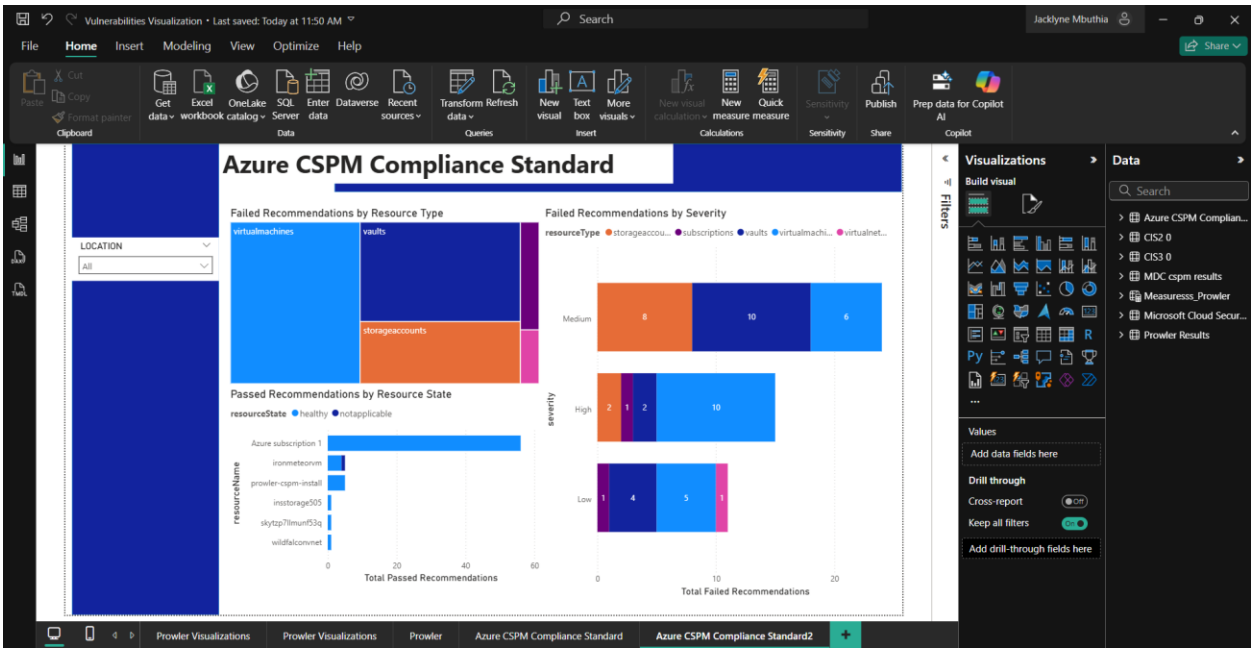

Figure 5: Power Bi Visualization Boards - 1

Image two:

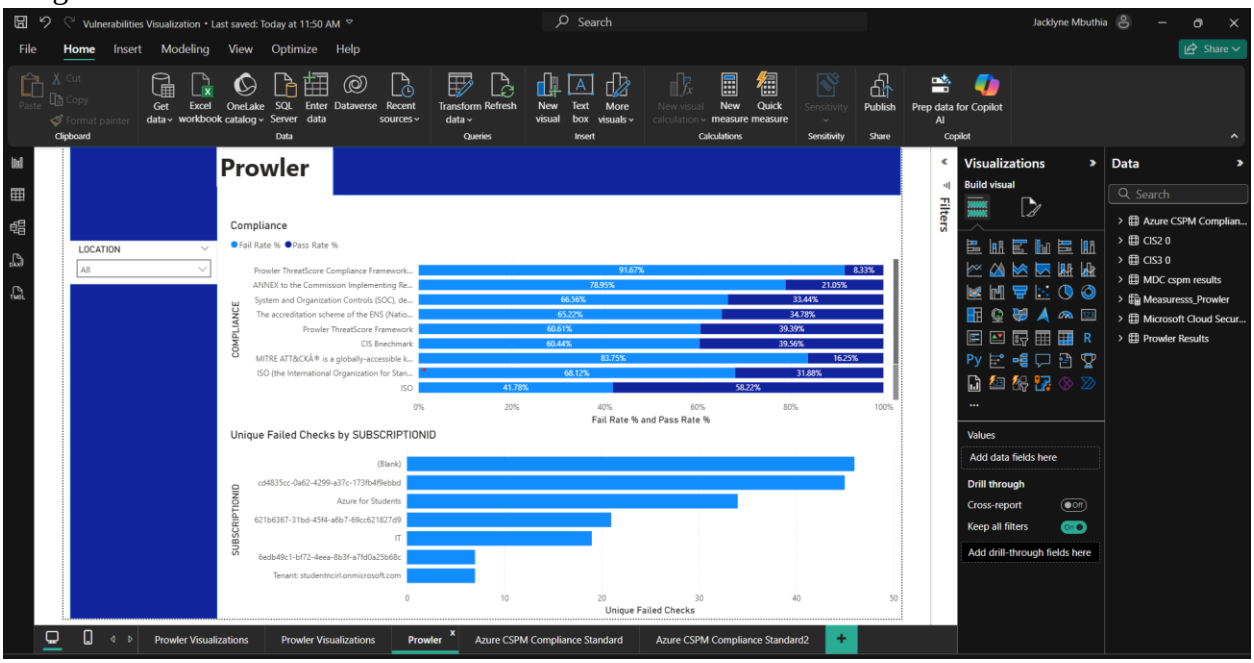

Figure 6:Power Bi Visualization Boards -2

Below is a screenshot of my project implementation brief that I added to my GitHub files.

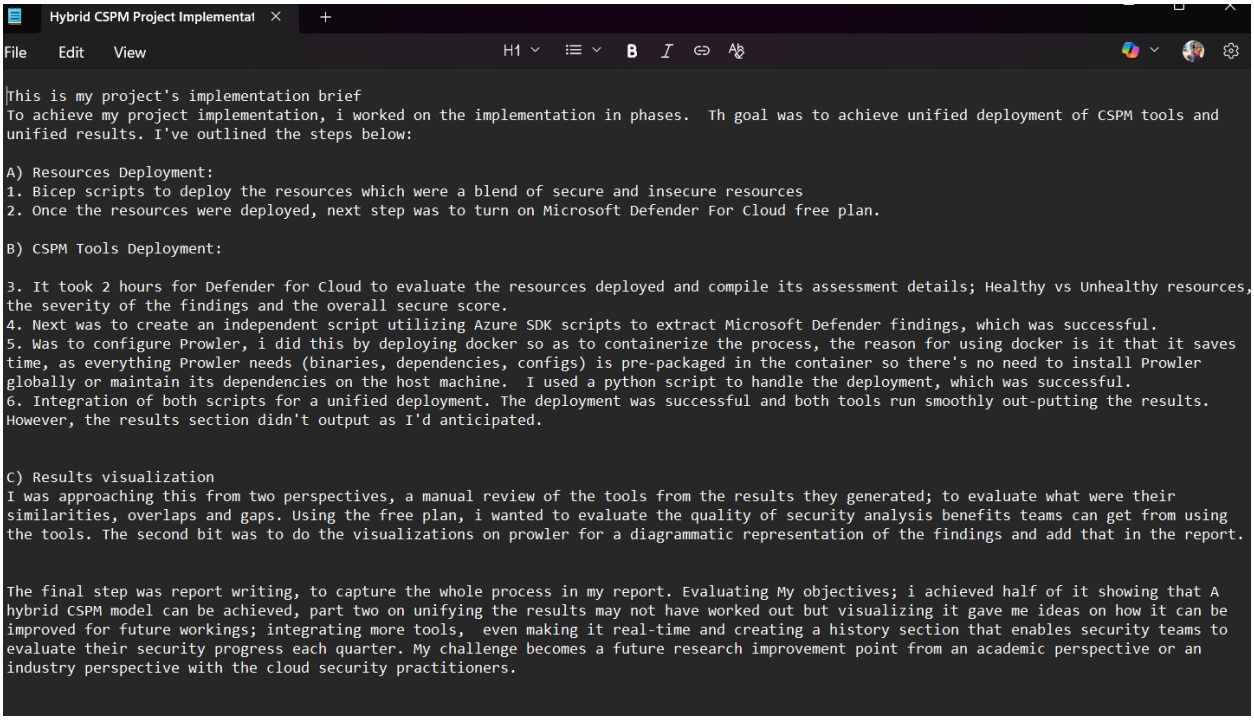

Figure 7: Project Implementation Brief