|  |
| --- |
|  |
| APDS7311 POE Task 1  Solutions Architecture: Plan your security and test security tools |
| |  |  |  | | --- | --- | --- | | Wonder-Developers | 9/4/24 | APDS7311 | |

***Group members***

Ashwin Pillay – ST10088734

Jaiden Naidoo – ST10218221

Tristan Singh – ST10022401

Mehluli Booi – ST10028039

Table of Contents

[**Question 1: Project Planning and Security: Data Flow** 3](#_Toc176340595)

[1.Data Flow Diagram 3](#_Toc176340596)

[a) Security Information Provided as Input 4](#_Toc176340597)

[b) Securing Data in Transit 4](#_Toc176340598)

[c) Hardening the Portal Against Attacks 4](#_Toc176340599)

[**Question 2: MobSF and ScoutSuite** 6](#_Toc176340600)

[a) Report on MobSF 6](#_Toc176340601)

[1. INTRODUCTION 6](#_Toc176340602)

[2. FINDINGS FROM MOBSF TEST ON “WONDER-TIME” 7](#_Toc176340603)

[3. SUPPORTING THE USE OF MOBSF 14](#_Toc176340604)

[4. CONCLUSION 14](#_Toc176340605)

[b/c/d) Report on ScoutSuite 15](#_Toc176340606)

[1. SCOUTSUITE RUN AGAINST AWS INSTANCE 15](#_Toc176340607)

[2. CRITICAL SECURITY ISSUES RELATED TO THE IAM PASSWORD POLICIES IDENTIFIED BY SCOUTSUITE 16](#_Toc176340608)

[**REFERENCE LIST** 18](#_Toc176340609)

# **Question 1: Project Planning and Security: Data Flow**

## 1.Data Flow Diagram

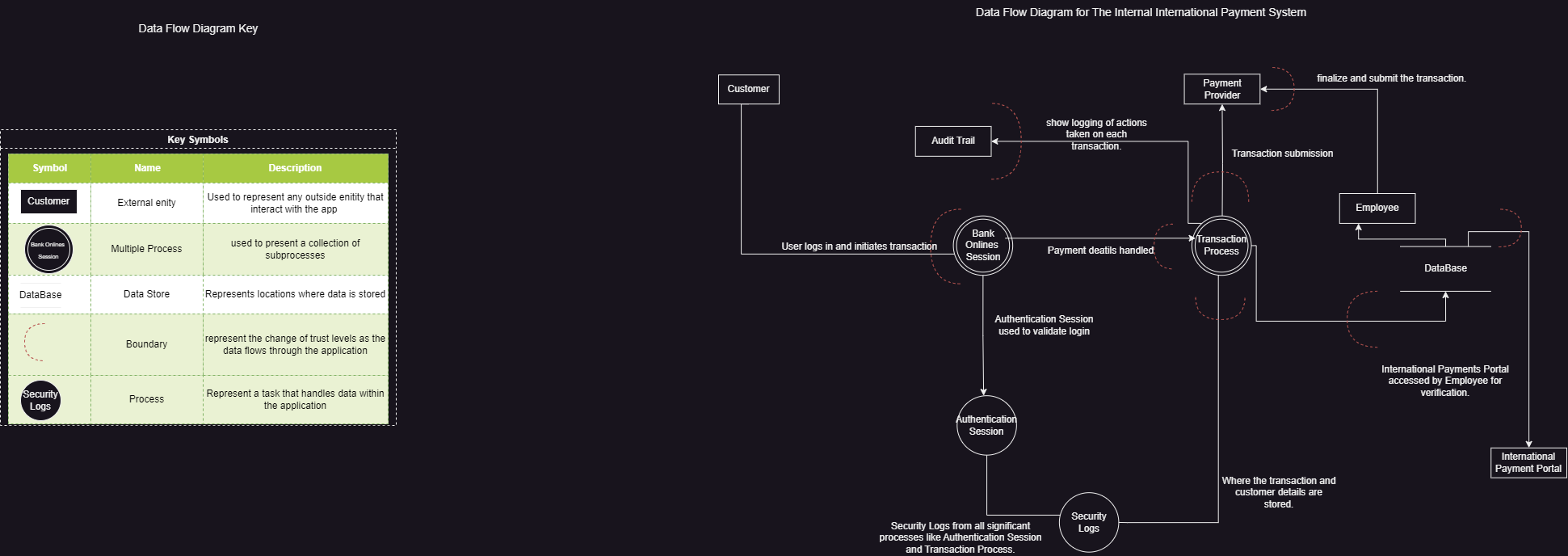


Figure 1: Booi, M.O. 2024 Diagram showing the flow of data. [Personal Drawing]. Johannesburg: Unpublished.

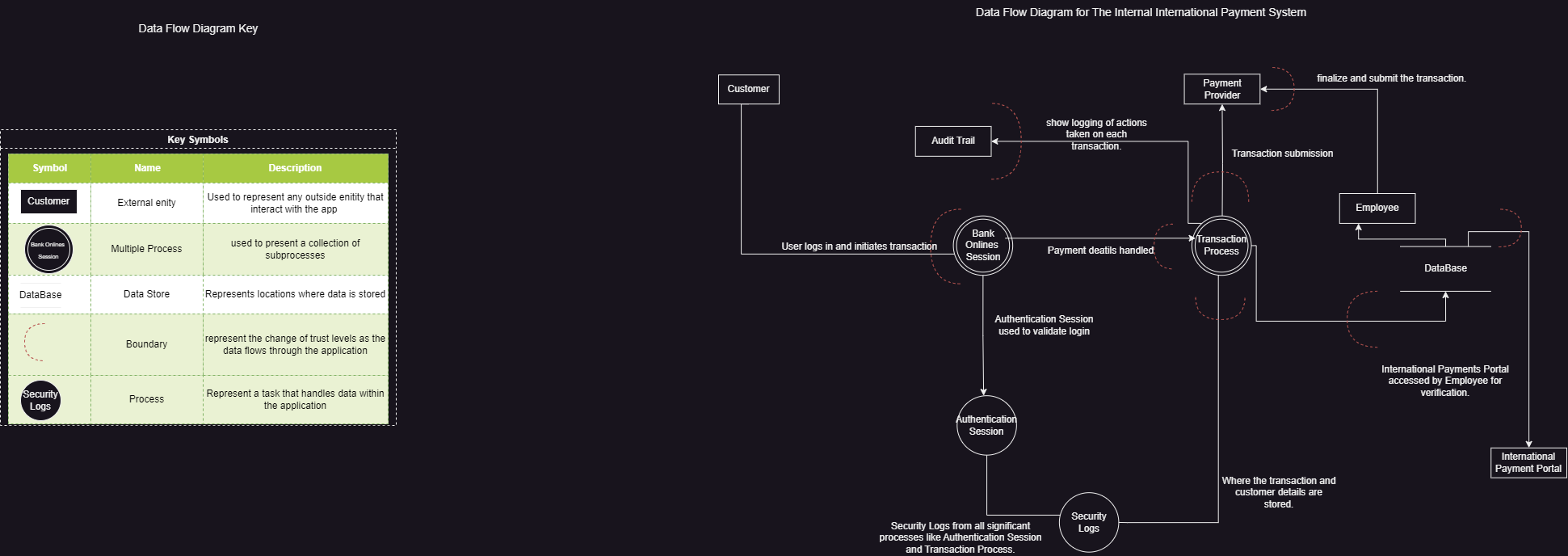


Figure 2: Booi, M.O. 2024 Key symbols for the data flow diagram. [Personal Drawing]. Johannesburg: Unpublished.

### Security Information Provided as Input

* + **Encryption**: Advanced encryption methods like AES-256 should be used to secure sensitive data before it is sent within the system, including login credentials and transaction details. Throughout the system, this encryption needs to be used consistently, particularly when a client, the Bank Online Session, and the Authentication Session are interacting. AES-256's 256-bit key length, which protects it against brute-force assaults, offers a strong degree of security. To further guarantee that keys are not readily compromised, encryption should be used in conjunction with safe key management procedures (Kiteworks, 2023).
  + **Input Validation**: To stop potentially dangerous inputs from being processed, each item of incoming data needs to be carefully vetted. Because input validation cleans data before it enters the database, it acts as the first line of defense against SQL injection attacks. Strict validation guidelines, such those offered by OWASP's Enterprise Security API (ESAPI), can be used to make sure that only data that is properly structured is handled. Code injection attacks can be reduced by using ESAPI to impose rules that encode or remove harmful characters (OWASP, 2023).
  + **Multi-Factor Authentication (MFA):** Using MFA requires users to authenticate themselves during the Authentication Session using several channels, such SMS, email, or authenticator applications. This provides a crucial layer of protection. This lessens the possibility of unwanted access, even in the event that login information is stolen. MFA must be required for all significant system transactions since it is particularly crucial for protecting access to sensitive data (National Institute of Standards and Technology [NIST], 2022).
  + **Hashing and Salting Passwords**: By hashing and salting passwords, security can be greatly improved. Passwords are changed by hashing into fixed-length character strings that are almost hard to decipher. By giving each password a distinct value prior to hashing, salting provides an extra degree of security by guaranteeing that even similar passwords produce distinct hashes. Because they are resistant to brute-force assaults, contemporary hashing algorithms like bcrypt and Argon2 are advised for adoption (OWASP, 2023).

### Securing Data in Transit

* + **SSL/TLS Encryption**: SSL/TLS protocols must be used to encrypt all data sent between the client and the server, especially during the Bank Online Session, Transaction Process, and communications with the Payment Provider. To guarantee that data is shielded from interception or manipulation during transmission, TLS 1.2 or 1.3 should be used. Disabling antiquated protocols like SSLv3, which are susceptible to known weaknesses like the POODLE attack, is essential. You should also upgrade encryption libraries on a regular basis to avoid using compromised algorithms (Qualys, 2023).
  + **Perfect Forward Secrecy (PFS):** PFS should be used to safeguard session keys even in the case of a server private key compromise. To prevent session keys from being decrypted in the past, PFS uses ephemeral Diffie-Hellman key exchanges. To increase security, using secure key groups like ffdhe3072 is advised. Even in the event of further significant breaches, PFS implementation is crucial to preserving the privacy of previous conversations (Kuhn & Metzger, 2023).

### Hardening the Portal Against Attacks

1. **Session Management:** Regenerating session IDs following a successful login and saving session tokens in secure cookies are two actions that improve session security and lessen the possibility of session hijacking (OWASP, 2023).
2. **Clickjacking Protection:** The X-Frame-Options HTTP header should be used to stop your International Payment Portal from being included in a frame on other websites. This will stop clickjacking and protect users from harmful acts that appear to be legitimate (Mozilla Developer Network, 2023).
3. **SQL Injection Prevention:** When using parameterized queries or Object-Relational Mappers (ORMs) to interact with the database, user inputs are strictly regarded as data and not as executable code. SQL injection attacks are avoided as a result (OWASP, 2023).
4. **Cross-Site Scripting (XSS) Mitigation:** Sanitizing all user inputs prior to their presentation on the interface and putting in place a Content Security Policy (CSP) that limits the sites from which scripts can be loaded are essential steps in preventing XSS attacks (Mozilla Developer Network, 2023).
5. **Prevention of Man-in-the-Middle (MitM) Attacks:** Preventing MitM attacks requires ensuring that all communications, particularly those involving the Customer and the Transaction Process, are secured using TLS. Further reducing these dangers is the implementation of HTTP Strict Transport Security (HSTS) to enforce HTTPS. According to Qualys (2023).
6. **DDoS Protection:** Use rate limiting and a Web Application Firewall (WAF) to block harmful traffic aimed at the International Payment Portal and Transaction Process to protect against Distributed Denial of Service (DDoS) attacks (Cloudflare, 2023).

# **Question 2: MobSF and ScoutSuite**

## Report on MobSF

### INTRODUCTION

According to Akhil (2023), ensuring the security of mobile applications is extremely important in today’s fast-paced digital environment. Mobile applications that are secure, prevent and defend against the theft of login credentials, personal data and financial information of all who use the app. What forms part of a continual dedication to protecting and securing IT Systems, MobSF (Mobile Security Framework) is used which is, according to Abraham (2024), a pioneering open-source instrument developed to automatically evaluate mobile application. The process of auto evaluation consists of security testing, malware detection, and the assessment of vulnerabilities.

This report aims to present the findings from a MobSF analysis carried out on the Wonder-Time Application and provides an assessment of the instrument’s effectiveness in improving security measures to be taken. This will be represented in a structured manner consisting of a 1. Introduction, 2. Finding from the test on an application called Wonder-Time, 3. A discussion supporting the use of the MobSF instrument, 4. a conclusion, and 5. a reference list.

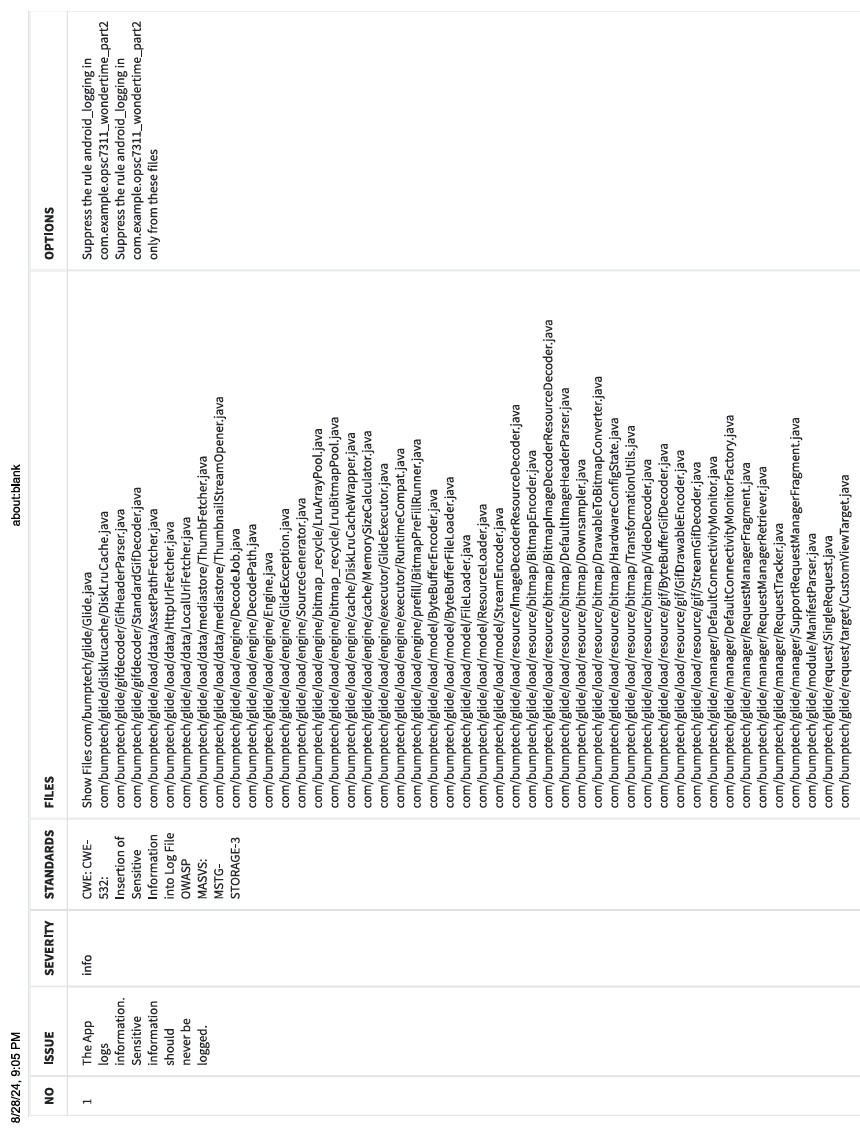
### FINDINGS FROM MOBSF TEST ON “WONDER-TIME”

Below are the most important results that MobSF has provided after a thorough analysis on the Wonder-Time Application, these results represented in this report are screenshots from the PDF files provided by MobSF and are presented in a table form with headings such as permission, status, info, description, info, severity, description, options, and code mappings Under which all the data with regards to security concerns of the app are tabulated:



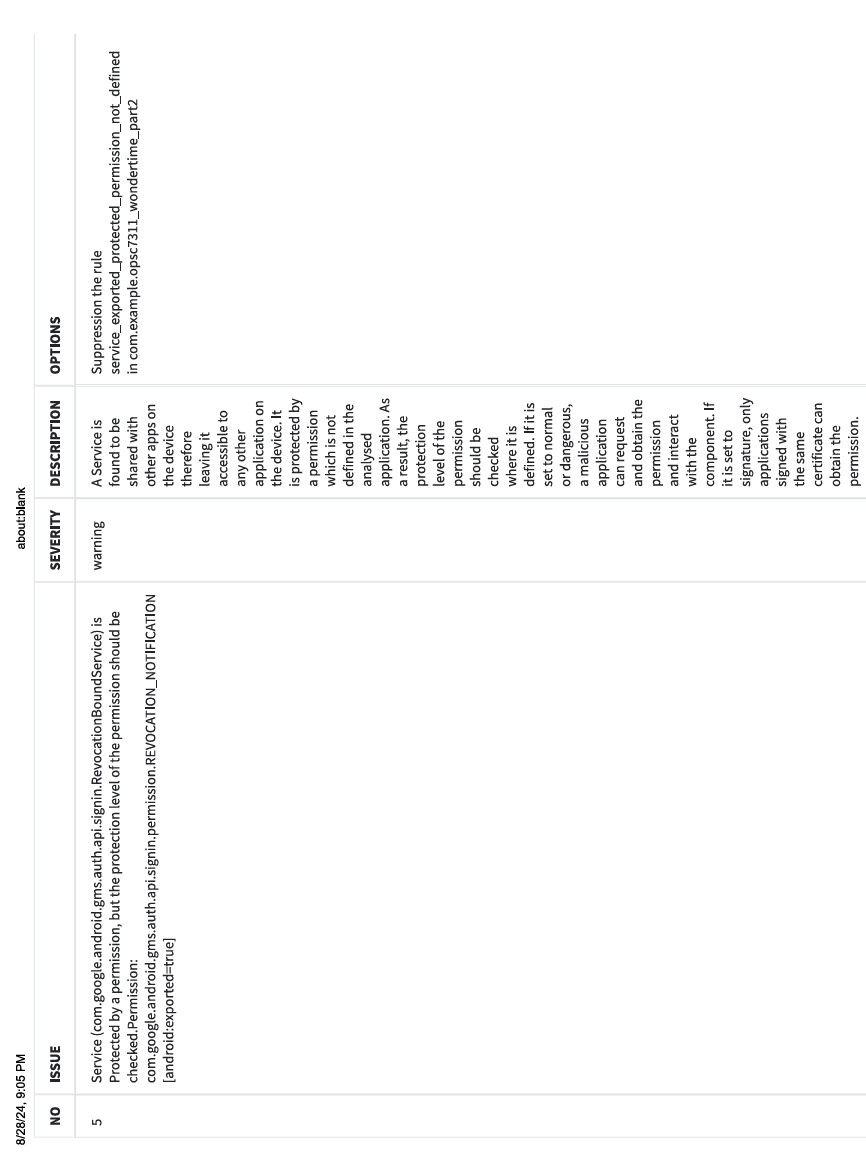




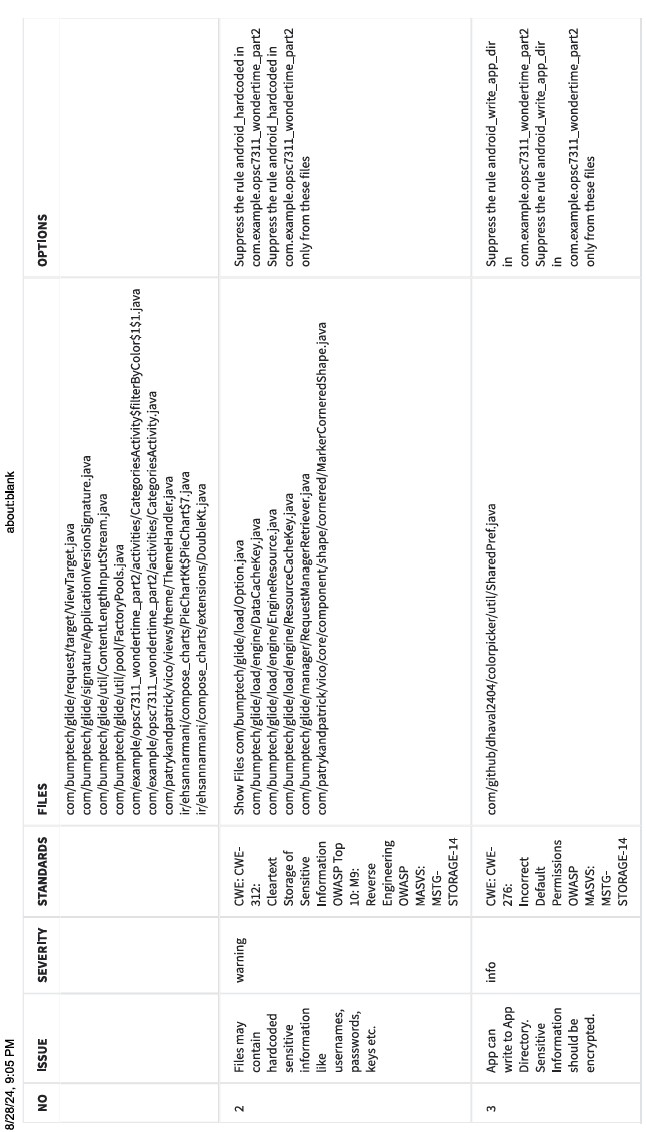














### SUPPORTING THE USE OF MOBSF

Based on the findings from the analysis done on the Wonder-Time application, it is clear that MobSF is functional and can identify potential security breaches that developers may not; additionally, it can be stated that MobSF has shown to be a very robust and comprehensive instrument for mobile application security analyses as it provides a variety of robust features, making it a vital asset for the organization’s security protocols (OpenAI, 2024). Below are attributes of MobSF that further support its use, relevance and need:

* **Automated Analysis**: By automating the security assessment procedure, MobSF makes it possible to quickly identify vulnerabilities without a lot of manual labour. This saves a great deal of time and work when doing comprehensive security inspections (OpenAI, 2024).
* **Extensive Reporting**: The tool offers thorough reports that contain a range of security metrics, including dangerous API requests, unsafe permissions, and possible vulnerabilities. With the use of these insights, developers may proactively handle security concerns throughout the development process (OpenAI, 2024).
* **Multi-Platform Support**: MobSF is adaptable to our development requirements because it can run both iOS and Android apps. We can keep a uniform security standard across all mobile goods thanks to this cross-platform feature (OpenAI, 2024).
* **Connectivity with CI/CD Pipelines**: MobSF can be connected to CI/CD (Continuous

Integration/Continuous Deployment) pipelines, enabling ongoing security observation during the development phase. As a result, there are fewer possible weaknesses in the finished product since risks are identified and mitigated early (OpenAI, 2024).

* **Open-Source Flexibility**: MobSF is an open-source solution that can be adapted and adjusted to fit certain organisational needs. The tool can adapt to our changing security requirements thanks to its versatility (OpenAI, 2024).

### CONCLUSION

improving mobile application security and identifying security flaws. MobSF helps our organization's security efforts with its automated analysis, comprehensive reporting, multi-platform support, and seamless integration capabilities.  
It is highly advised that we incorporate MobSF into our routine security procedures in order to protect our applications and user data, as it can expedite security testing and proactively handle possible problems.

## b/c/d) Report on ScoutSuite

### SCOUTSUITE RUN AGAINST AWS INSTANCE

In cloud security, ensuring that AWS environments are configured according to best practices is crucial for maintaining robust defences. The combination of Scout Suite and AWS CLI offers a practical approach for managing and auditing AWS settings. This report summaries the findings from running Scout Suite against an AWS instance, highlighting key security issues and providing recommendations for improving password policies.

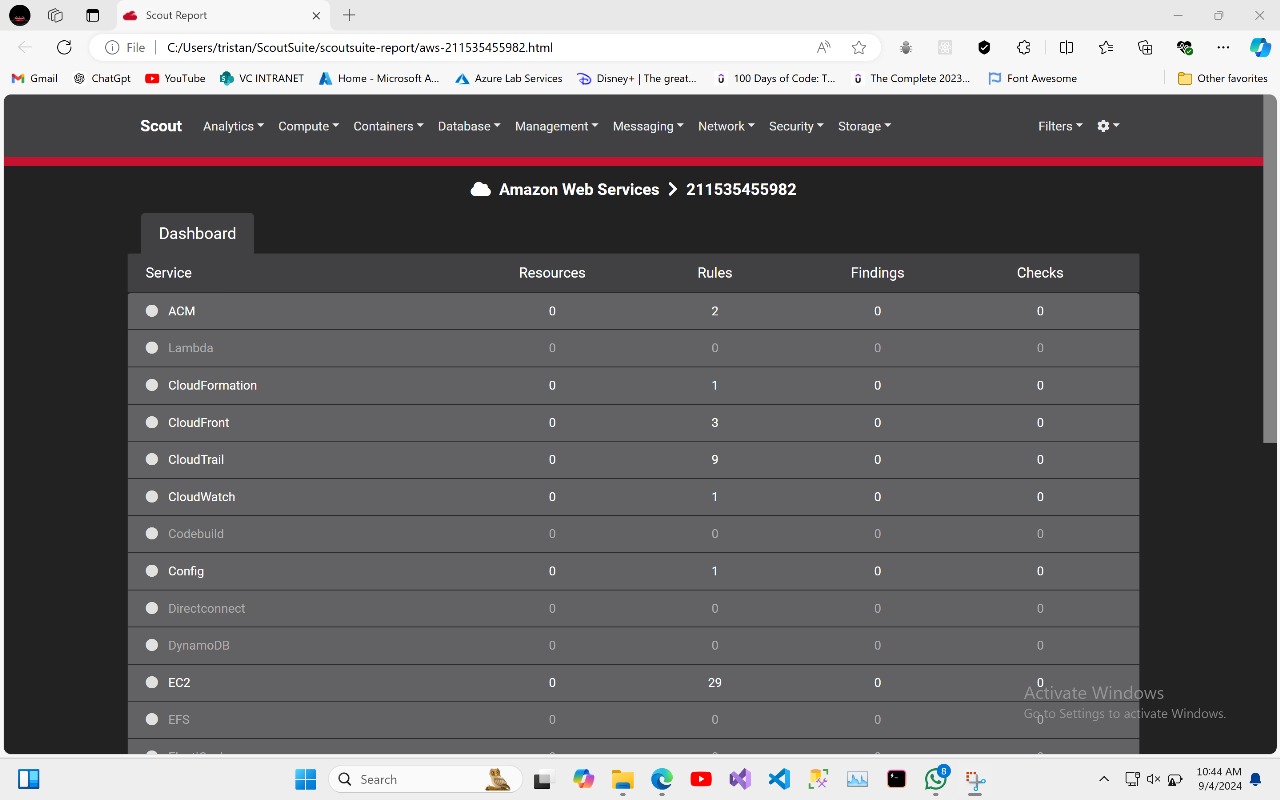
Scout Suite and AWS CLI provide an effective combination for administering and auditing AWS settings, providing a balance of simplicity and productivity. AWS CLI, as a command-line tool, enables users to directly interface with AWS services, providing precise control and automation possibilities. It's simple to configure and integrate.

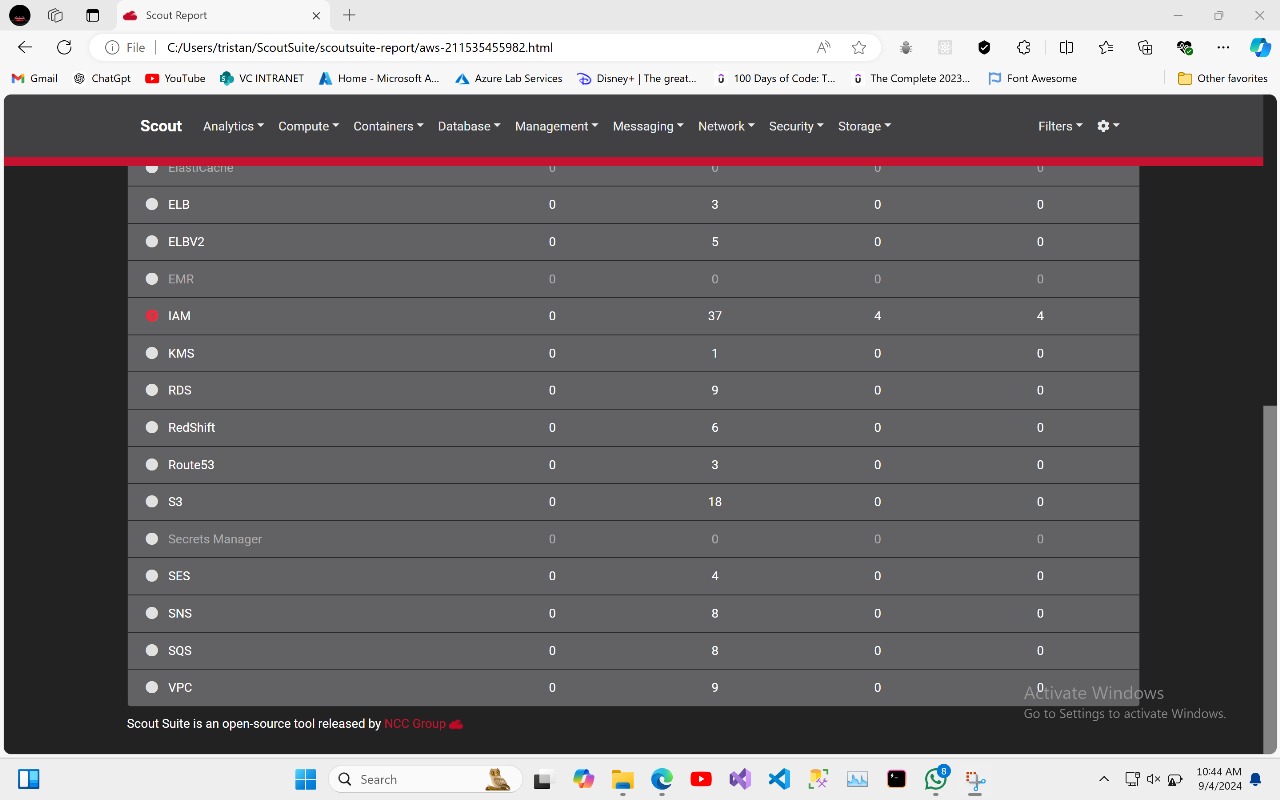
Scout Suite, on the other hand, enhances the AWS CLI by providing a more visual and comprehensive security auditing tool. Its ease of use is demonstrated by its ability to generate rich, interactive HTML reports that identify potential security concerns across many AWS services. Running Scout Suite after configuring AWS CLI is relatively simple, requiring only a few command executions to scan the environment.

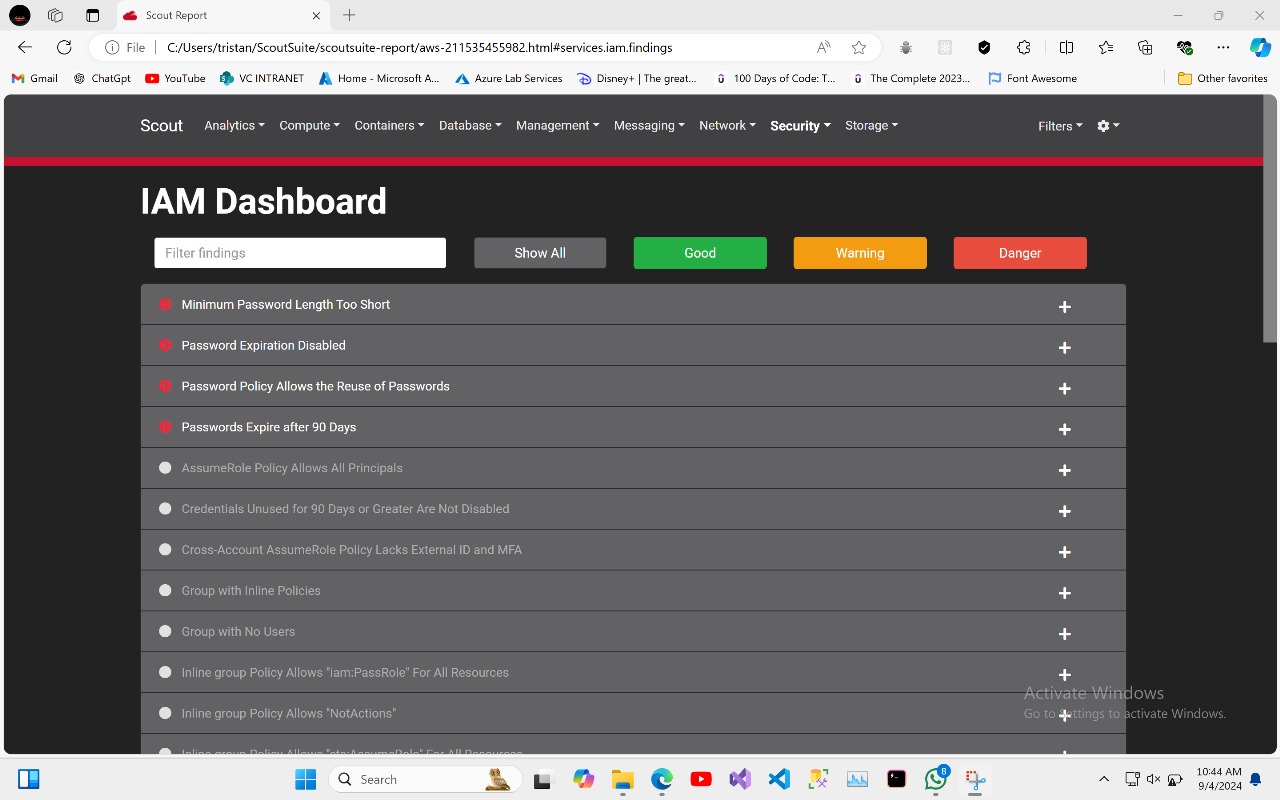
Overall, the combination of AWS CLI with Scout Suite provides a good blend of command-line management and user-friendly visual reporting. While some initial configuration is required, particularly when setting AWS CLI and installing Scout Suite, the tools are simple and effective for both new and experienced users. Scout Suite generates detailed reports that make it easier to discover and handle security risks, hence improving overall AWS management.

### CRITICAL SECURITY ISSUES RELATED TO THE IAM PASSWORD POLICIES IDENTIFIED BY SCOUTSUITE

2.1) Screenshots







2.2) Discussion

* **The minimum password length is too short**: The password policy enables passwords less than 14 characters, which is contrary to suggested security best practices. To improve security, it is recommended that a password be at least 14 characters long.
* **Password Expiration Disabled**: Password expiration is not enforced, allowing hacked credentials to be used forever. It is recommended to implement a policy requiring password changes on a regular basis, ideally every 90 days.
* **Password Reuse Allowed**: The current policy allows password reuse, which reduces overall security. To reduce risks, the policy should be set to prohibit users from reusing past passwords.

These flaws indicate significant weaknesses in password management that could be exploited if not corrected. Remediation steps are recommended to match with the CIS Amazon Web Services Foundations benchmarks, ensuring that password policies adhere to established security guidelines.

The Scout Suite report has revealed several critical vulnerabilities in the IAM password policies that could compromise AWS security. The identified issues, including insufficient password length, disabled password expiration, and allowed password reuse, present significant risks. Addressing these concerns by aligning with CIS Amazon Web Services Foundations benchmarks will enhance the overall security posture of the AWS environment. By taking the recommended remediation steps, organizations can fortify their security practices and better protect against potential threats.

# **REFERENCE LIST**

Abraham, A., 2024. *Github.* [Online]   
Available at: https://github.com/MobSF/Mobile-Security-Framework-MobSF  
[Accessed 28 August 2024].

Akhil, 2023. *Apptunix.* [Online]   
Available at: https://www.apptunix.com/blog/why-is-mobile-application-security-important-best-practices/  
[Accessed 28 August 2024].

OpenAI, 2024. *OprnAI.* [Online]   
[Accessed 29 August 2024].

Cloudflare. 2023. Distributed Denial of Service (DDoS) protection. [Online]. Available at: <https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/> [Accessed 3 September 2024]

Kiteworks. 2023. AES-256 Encryption: Best practices for securing data. [Online]. Available at: <https://www.kiteworks.com/encryption/aes-256-encryption/> [Accessed 3 September 2024].

Kuhn, A. & Metzger, B. 2023. Perfect Forward Secrecy: How to implement it using Diffie-Hellman. [Online]. Available at: [https://www.cybersecurity-insiders.com/perfect-forward-secrecy /](https://www.cybersecurity-insiders.com/perfect-forward-secrecy%20/) [Accessed 3 September 2024].

Mozilla Developer Network. 2023. X-Frame-Options. [Online]. Available at: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options> [Accessed 3 September 2024].

National Institute of Standards and Technology (NIST). 2022. Multi-Factor Authentication (MFA) guidelines. [Online]. Available at: <https://www.nist.gov/mfa-guidelines> [Accessed 3 September 2024].

OWASP. 2023. ESAPI project: Enterprise Security API. [Online]. Available at: <https://owasp.org/www-project-enterprise-security-api/> [Accessed 3 September 2024].

Qualys. 2023. SSL/TLS security: Best practices. [Online]. Available at: <https://www.ssllabs.com/ssltest/> [Accessed 3 September 2024].

OWASP. 2024. Threat Modeling Process. [Online]. Available at: https://owasp.org/www-community/Threat\_Modeling\_Process [Accessed 4 September 2024].

Drummond, J. 2023. OWASP Threat Modeling. [YouTube]. Available at: https://youtu.be/rEnJYNkUde0?si=ik-1UbePADs5W\_PR [Accessed 4 September 2024].