PATHWAY: APDS 7311

Lecturer: Mr. Isaac Leshaba

04 September 2024

905 words

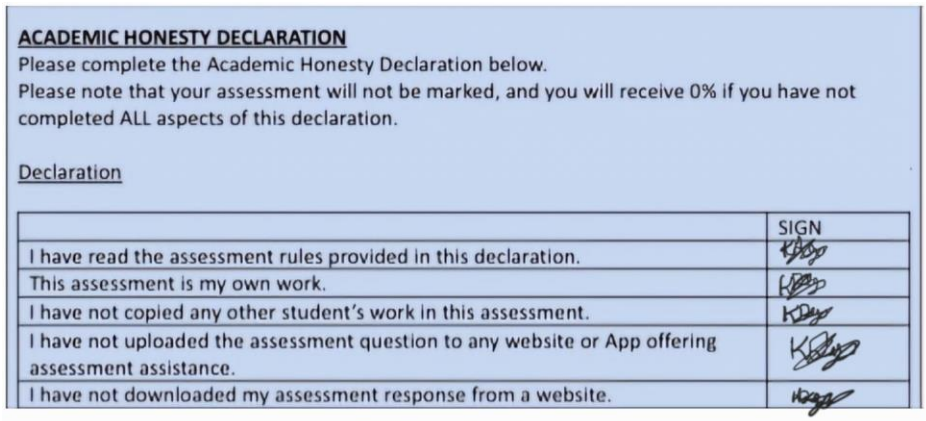
Part 1

ST10033851 – Jereshan Sinan

ST10079389 - Kaushil Dajee

ST10030291 - Avish Judnarain

ST10091324- Eben Mwema

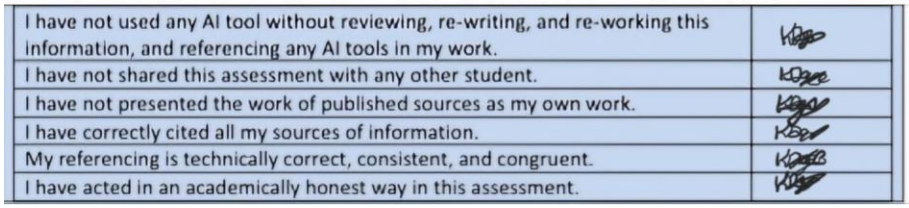


ACADEMIC HONESTY DECLARATION

Please complete the Academic Honesty Declaration below.
Please note that your assessment will not be marked, and you will receive 0% if you have not completed ALL aspects of this declaration.

Declaration

	SIGN
I have read the assessment rules provided in this declaration.	
This assessment is my own work.	
I have not copied any other student's work in this assessment.	
I have not uploaded the assessment question to any website or App offering assessment assistance.	
I have not downloaded my assessment response from a website.	



I have not used any AI tool without reviewing, re-writing, and re-working this information, and referencing any AI tools in my work.	
I have not shared this assessment with any other student.	
I have not presented the work of published sources as my own work.	
I have correctly cited all my sources of information.	
My referencing is technically correct, consistent, and congruent.	
I have acted in an academically honest way in this assessment.	

Declaration

I Eben Nkulu Mwema declare that:

	SIGN
I have read the assessment rules provided in this declaration	E.N.M
This assessment is my own work.	E.N.M
I have not copied any other student's work in this assessment.	E.N.M
I have not uploaded the assessment question to any website or App offering assessment assistance.	E.N.M
I have not downloaded my assessment response from a website.	E.N.M
I have not used any AI tool without reviewing, re-writing, and re-working this information, and referencing any AI tools in my work.	E.N.M
I have not shared this assessment with any other student.	E.N.M
I have not presented the work of published sources as my own work.	E.N.M
I have correctly cited all my sources of information.	E.N.M
My referencing is technically correct, consistent, and congruent.	E.N.M
I have acted in an academically honest way in this assessment.	E.N.M

ACADEMIC HONESTY DECLARATION

Please complete the Academic Honesty Declaration below.
Please note that your assessment will not be marked, and you will receive 0% if you have not completed ALL aspects of this declaration.

Declaration

	SIGN
I have read the assessment rules provided in this declaration.	Avish
This assessment is my own work.	Avish
I have not copied any other student's work in this assessment.	Avish
I have not uploaded the assessment question to any website or App offering assessment assistance.	Avish
I have not downloaded my assessment response from a website.	Avish
I have not used any AI tool without reviewing, re-writing, and re-working this information, and referencing any AI tools in my work.	Avish
I have not shared this assessment with any other student.	Avish
I have not presented the work of published sources as my own work.	Avish
I have correctly cited all my sources of information.	Avish
My referencing is technically correct, consistent, and congruent.	Avish
I have acted in an academically honest way in this assessment.	Avish

ACADEMIC HONESTY DECLARATION

Please complete the Academic Honesty Declaration below.
Please note that your assessment will not be marked, and you will receive 0% if you have not completed ALL aspects of this declaration.

Declaration

	SIGN
I have read the assessment rules provided in this declaration.	
This assessment is my own work.	
I have not copied any other student's work in this assessment.	
I have not uploaded the assessment question to any website or App offering assessment assistance.	
I have not downloaded my assessment response from a website.	
I have not used any AI tool without reviewing, re-writing, and re-working this information, and referencing any AI tools in my work.	
I have not shared this assessment with any other student.	
I have not presented the work of published sources as my own work.	
I have correctly cited all my sources of information.	
My referencing is technically correct, consistent, and congruent.	
I have acted in an academically honest way in this assessment.	

Contents

Figures

1. SECURITY SOLUTIONS

1.1 User Flow

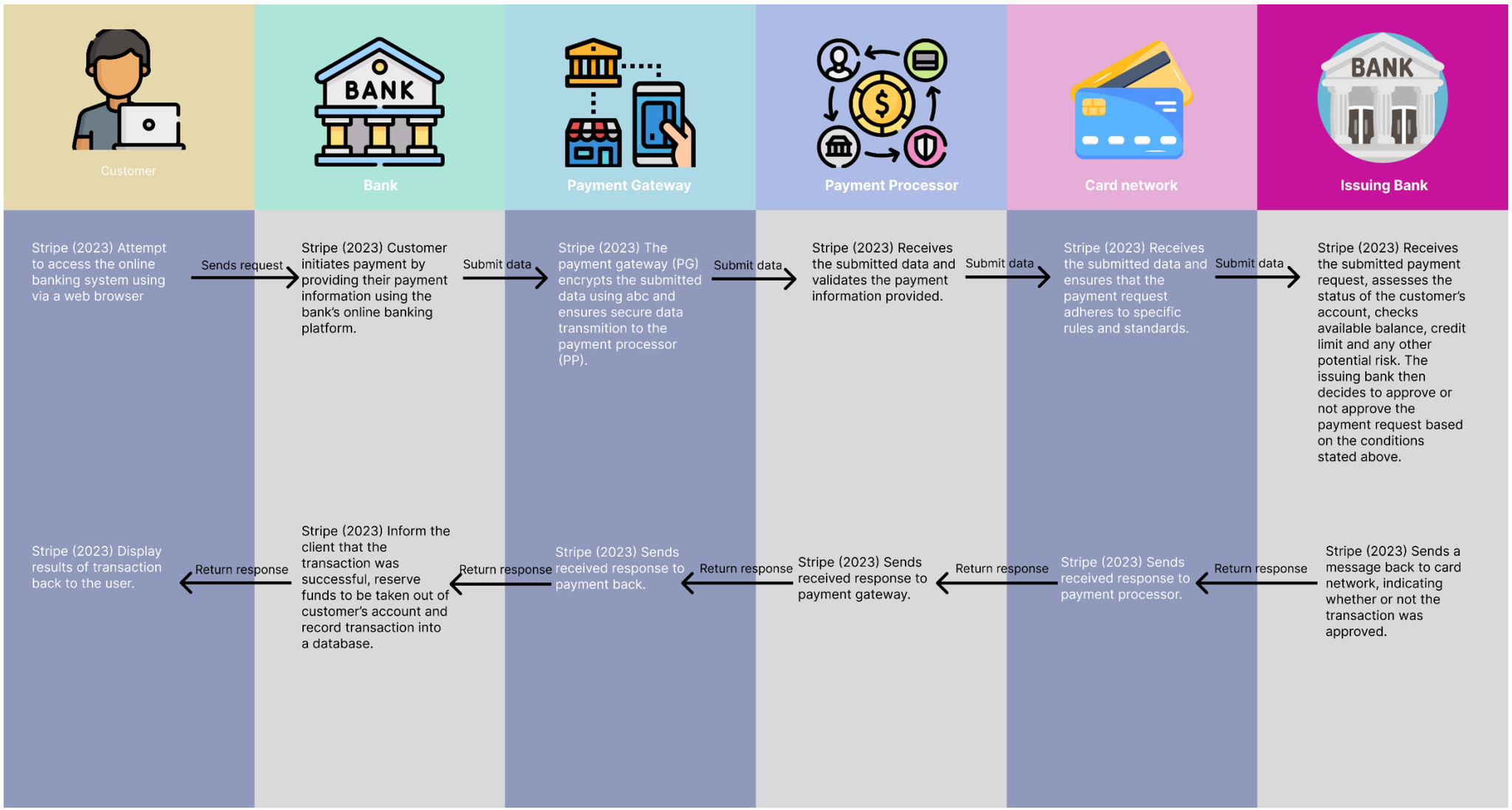


Figure 1: Data Flow (Mwema, 2024)

1.2 Protecting input data in transit

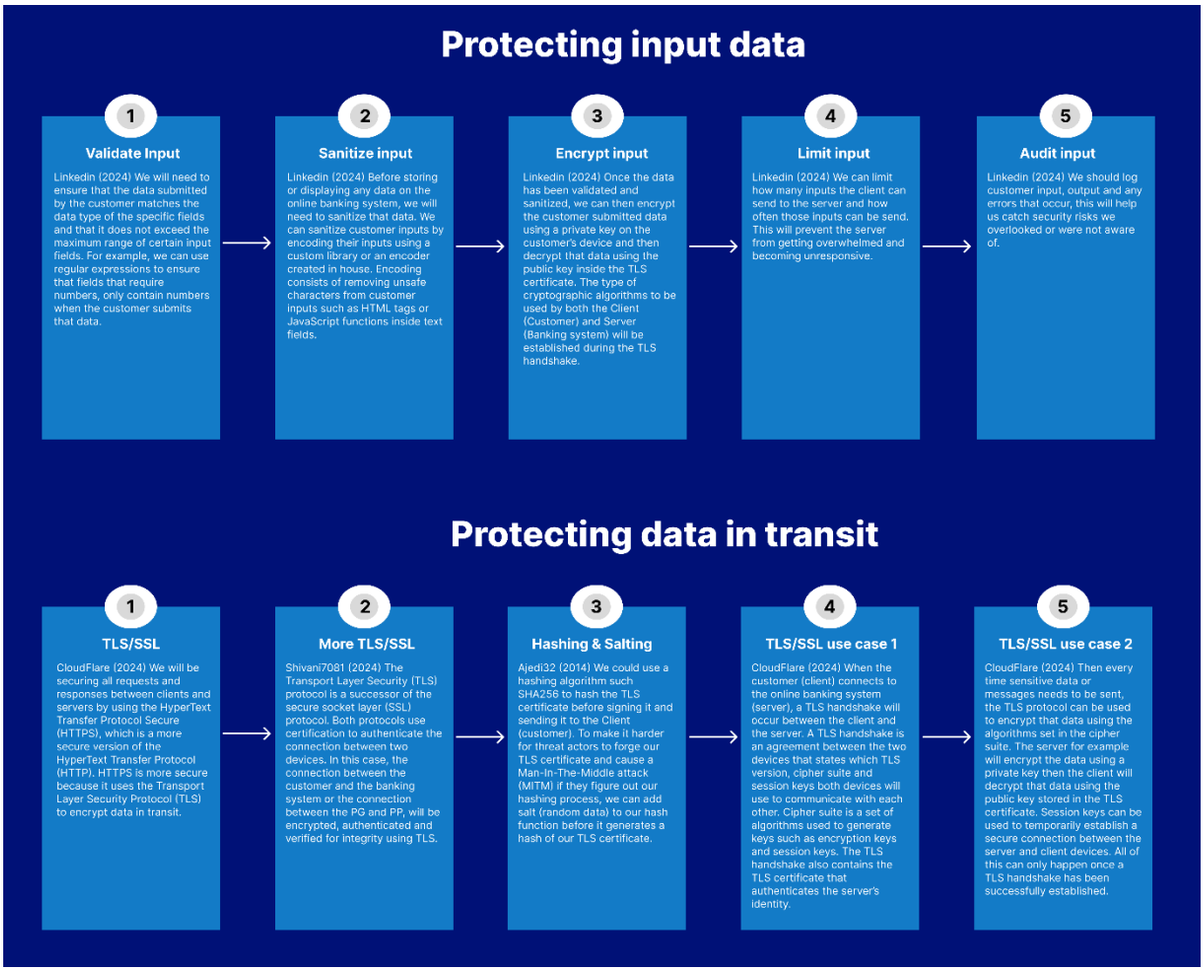


Figure 2: Data Protection (Eben, 2024)

1.3 Session Jacking attack

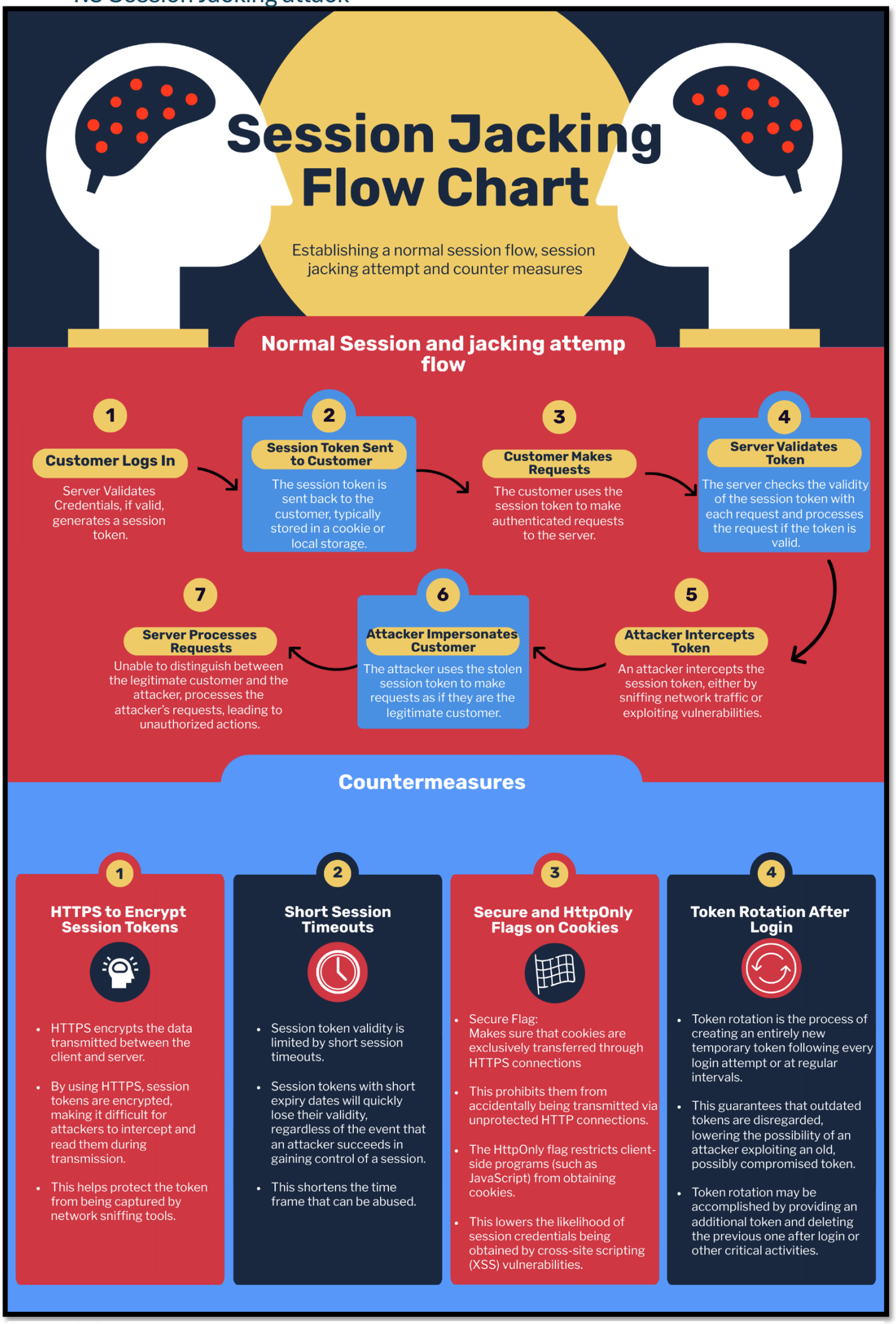
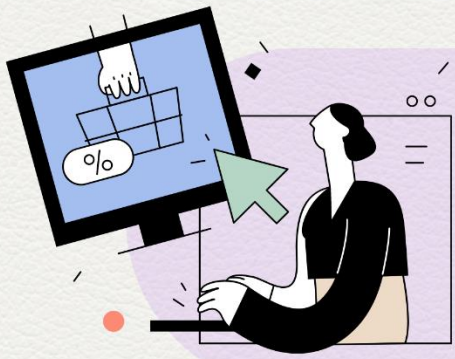


Figure 3 Session Jacking

1.4. Click Jacking Attack

Clickjacking Attack

Flow of how an attacker tricks a user into clicking on something different from what the user perceives



1

User Visits Malicious Website

The attacker builds a fake website with an undetectable iframe layered on the top of a genuine financial website.

2

User visits malicious website

The victim is deceived into accessing the attacker's website, usually using phishing emails or deceptive links.

Legitimate Webpage Loaded in Hidden iframe

The financial institution's official webpage is displayed within the fraudulent site's invisible iframe, with important buttons such as "Pay Now" placed behind the visible ones.

3

4

User clicks on visible element

The user attempts to engage with the visible content within the malicious site, including clicking a button they believe is secure, unaware that the action they take is being recorded on the undetected iframe.

5

Hidden Click Triggered

The invisible iframe captures the user's click, causing an unexpected event on the website of the financial institution, such as verifying a payment or updating account information.

6

Unintended Action Executed

The bank's website treats the unintentional operation as if it were purposeful, which might lead to unauthorised payments or other negative activity.

Preventative measures

X-Frame-Options Headers

- The X-Frame-Options header stops third-party websites from loading our webpages as iframes.
- Configuring the header's value to DENY or SAMEORIGIN ensures that the website won't be framed, thereby preventing clickjacking attacks.

UI Redress Techniques

- Frame Busting Scripts: These scripts identify whether our webpage is rendered within an iframe and, if so, force the website to exit free from the frame.
- Visual verification: Before doing crucial tasks (e.g., verifying a transaction), users must complete an extra action, such as clicking a non-embedded graphic component, such as a CAPTCHA.

Content Security Policy (CSP)

- CSP is an encryption protocol designed to avoid a range of threats, especially clickjacking, by limiting the sites that can frame our material.
- This prevents framing of the material by including the frame-ancestors 'none'; directives in the CSP, which also protects from clickjacking attacks.

Figure 4 Click Jacking (Imperva, 2019)

1.5 SQL Injection Attack

ATTACKER DISCOVERS UNPROTECTED INPUT FIELD

The attacker discovers an input field (e.g., registration form, search bar) on the web page of a website that allows direct interaction with the database's contents.

01

ATTACKER INJECTS MALICIOUS SQL CODE 02

Instead of simply typing the anticipated input, the attacker inserts illicit SQL code into the message box. For example, attackers may type '; DROP TABLE users; --' to try to destroy each user in the table.

SQL INJECTION ATTACKS

CounterMeasures

WHITELIST APPROPRIATE INPUTS USING REGEX:

- Input validation entails ensuring that user inputs adhere to specified forms before processing them.
- Regular expressions (RegEx) allows us to construct a precise pattern that input data has to correspond to. This keeps attackers from introducing malicious data or instructions into entry fields.

PARAMETERISED QUERIES

- Parameterised queries guarantee that user input is regarded as data rather than executable software.
- By utilising placeholders for user data and assigning the actual data at runtime, a database engine recognises input as a text or integer instead of a SQL query. This protects against database injection threats by splitting user input from query logic.

ORM (OBJECT-RELATIONAL MAPPING) TOOLS

- ORMs encapsulate the SQL layer, enabling developers to interface with databases through high-level programming techniques instead of raw SQL.
- This abstraction barrier helps to avoid SQL injection by automating input sanitisation and guaranteeing user inputs aren't included directly in query results.

THE APPLICATION RUNS HARMFUL SQL CODE

The backend program incorrectly interprets the malicious inputs as genuine SQL and runs it on the database system. Because the input was not adequately sanitised, the inserted SQL code changed the original query.

03

AN ATTACKER GAINS UNAUTHORISED ACCESS

The attacker may get unauthorised entry to confidential information or possibly take possession of the database, subject to the characteristics of the malicious SQL query.

05

DATABASE COMPROMISED

The database performs the inserted SQL command, which might result in unauthorised access to information, modification, or deletion. This may result in a breach of security or the destruction of crucial information.

04

Figure 5 SQL Injection

1.6 Cross Site Scripting Attack

Figure 6 Cross Site Scripting

1.7 Man in the Middle Attack

Man-in-the-Middle (MITM) Attack Flow

Shows the flow where an attacker intercepts communication between the client and server.

1

Client initiates connection

The consumer logs onto the bank's web portal, enters their login details, and sends the details to a server.

2

Attacker intercepts transmission

An attacker intercepts messages sent between the client's device and the server, possibly obtaining confidential details like as login passwords or payment data.

3

Attacker Relays and Alters Data

The attacker transfers acquired data involving the client and the server, sometimes modifying it before sending it back to the server.

4

Client receives response

A client obtains a response from the server side, unaware that its contents was accessed and perhaps manipulated. The activity is finished although with deliberately altered information.

5

Server Processes Malicious Data

The server interprets the altered data as if it were real, resulting in unauthorised operations or theft of information.

Countermeasures

01

HTTPS (TLS/SSL) to secure every message in transit.

- HTTPS secures content sent between the client and the server through Transport Layer Security (TLS). This assures that if a hacker intercepts the interaction, the data stays encoded and inaccessible.
- All pages, particularly those containing confidential data such as login or payment credentials, must be delivered over HTTPS.

02

Reliable authentication and manage sessions.

- Strong authentication techniques (e.g., multi-factor authentication) render it more difficult for attackers to get unauthorised access, regardless of how they acquire user credentials.
- Secure session control measures, including resetting session tokens after verification and limiting session lifetimes, help to restrict the area of attack for MITM assaults.

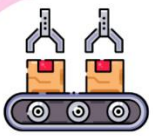
03

Certificate pinning

- This is the process of pairing a certain public key with a specific server in order for clients to only accept certificates from the real server.
- This blocks attackers against using forged certifications to mimic the server during an MITM attack.

04

HSTS (HTTP Strict Transport Security)



- HSTS is a network safety protocol that requires browsers to exclusively connect with the server via HTTPS, regardless if the user tries to visit the page using HTTP.
- This helps to avoid downgraded attacks, in which an attacker attempts to reroute data to an unverified HTTP instance of the site.

Figure 7 Man in the Middle (fortinet, 2024)

1.8 DDoS Attack

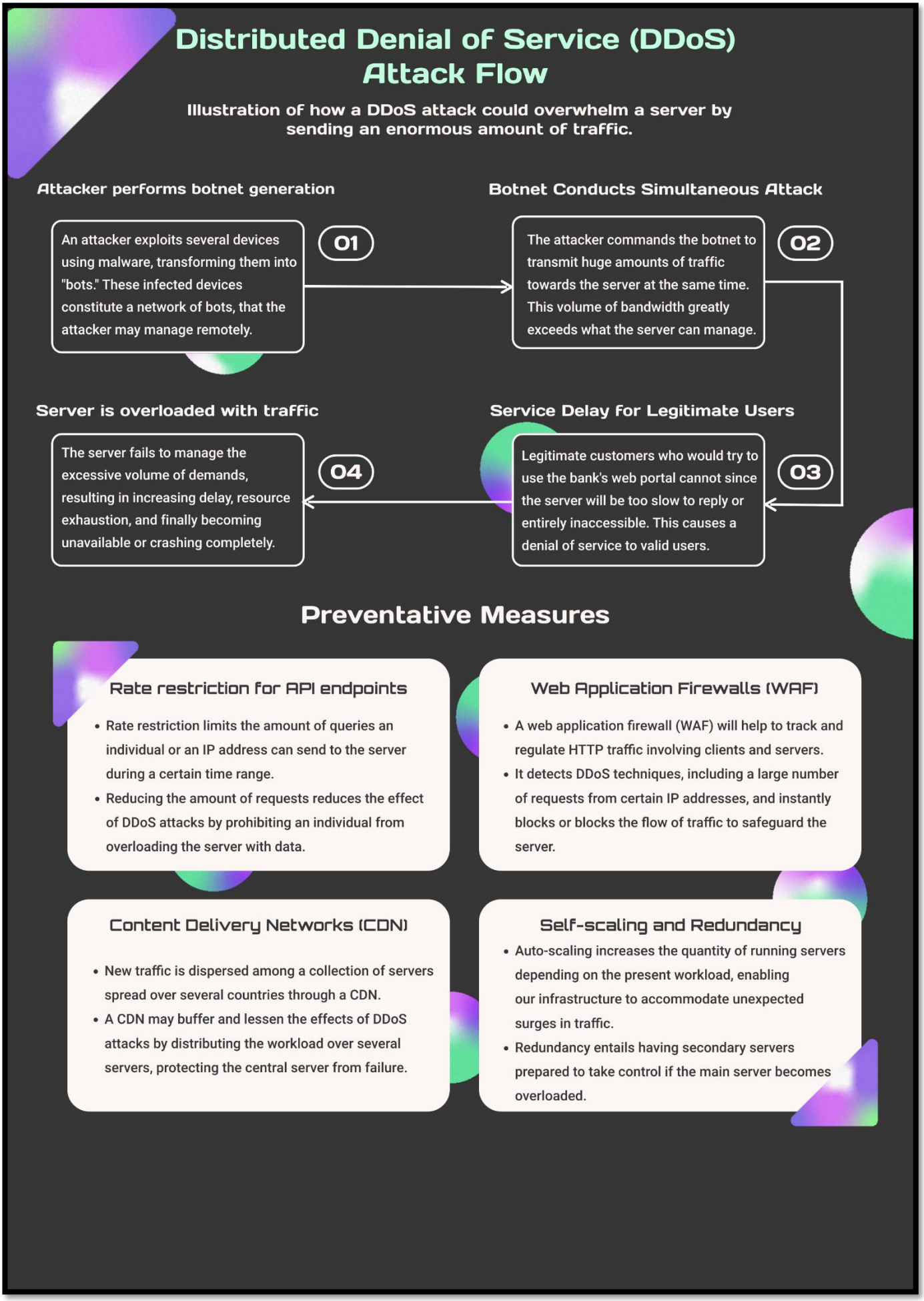


Figure 8 DDoS (Hasan et al., 2023)

Question 2

2.a)

INTRODUCTION

Mobile Security Framework (MobSF) is an open-source, automated mobile application security testing tool that can perform static and dynamic analysis of mobile applications for Android and iOS (Das, 2023).

BODY

Findings from MobSF

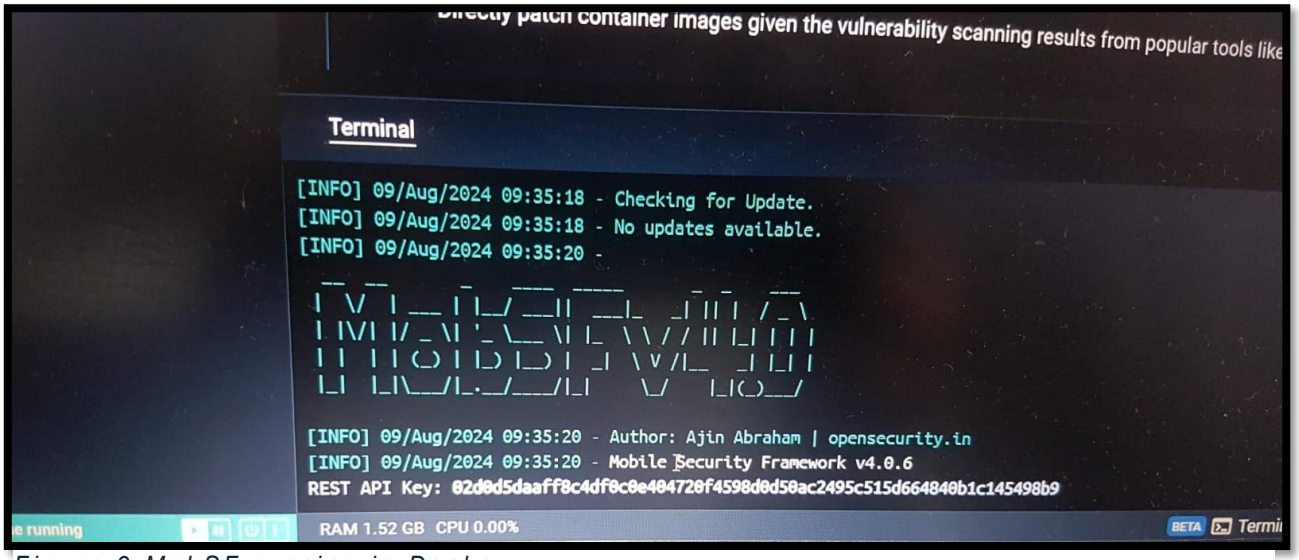


Figure 9 MobSF running in Docker

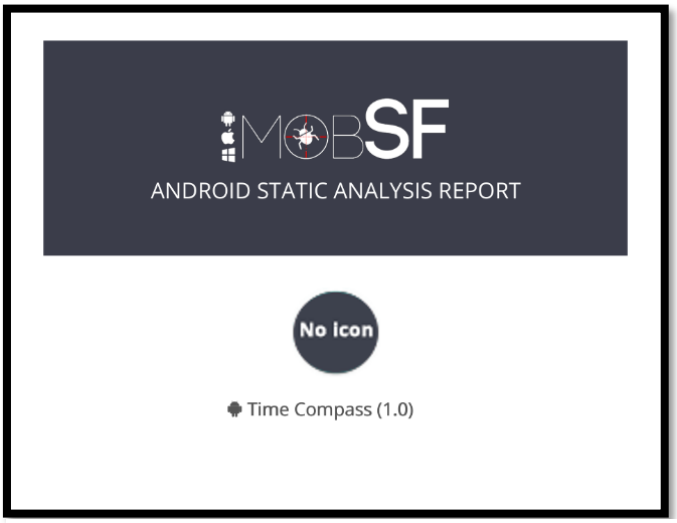


Figure 10 MobSF Report

Figure 11 MobSF Grade for OPSC7311 Mobile Application

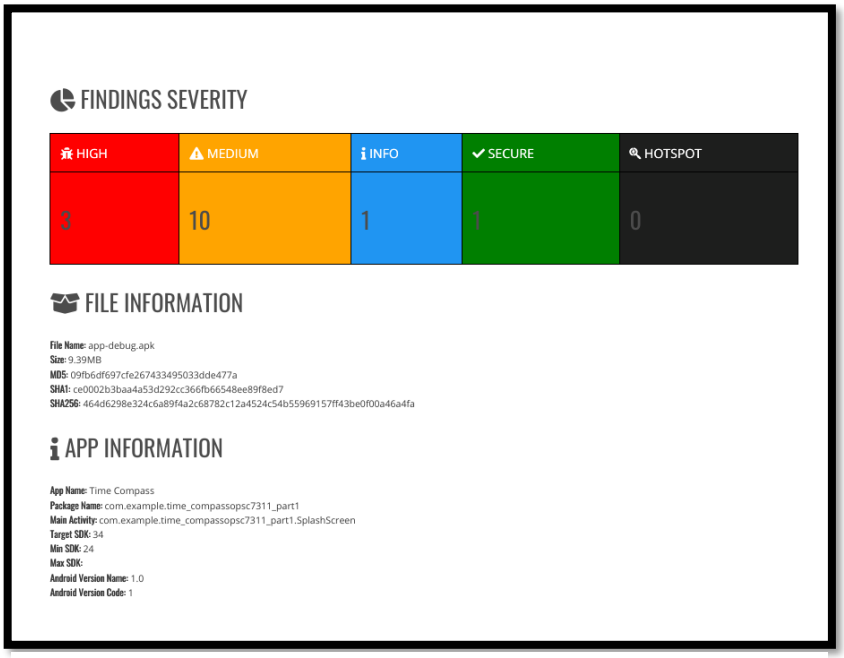


Figure 12 MobSF Findings

From scanning our OPSC7311 mobile application, Time Compass, MobSF has created a report explaining our application and its security. We received a 44/100 for the app security score with a medium risk.

Why do we argue for using the tool?

- We argue that MobSF is a great tool that should be used more often, especially since it is a great free-to-use open-source tool. According to Das (2023), this tool can easily be customized to fit specific requirements.
- MobSF is easy to use, this tool is great because you can easily install and use it, with such a simple user-friendly interface which makes it accessible to security analysts and developers with different levels of experience.
- MobSF can be used to help developers improve the security of their mobile applications especially since MobSF can perform a deep analysis of the source code or APK files if no source code is available, which can help developers identify any potential vulnerabilities in the application and then easily try and improve the security of their applications.
- MobSF allows for static and dynamic analysis, as this tool covers a wide range of vulnerabilities and security issues such as data storage, insecure communication, and code vulnerabilities.
- MobSF receives regular updates with new features, security checks, and improvements, ensuring that it remains current with the latest mobile security threats (OpenAI, 2024).
- This tool can save so much time for developers, instead of finding these vulnerabilities, this tool's automated testing can be done within a matter of minutes various vulnerabilities and issues will be identified with ease, which can save time, and it ensures important security checks are not overlooked.
- According to (OpenAI, 2024) MobSF allows developers to easily automate security assessments. Integrating MobSF's features into their system will allow for an enhancement in the usability of automated workflows.

Conclusion

In conclusion, based on the research we have done on MobSF we believe that it will be a great tool to use to test mobile security especially since it can help not only speed up identifying weaknesses but can also expose areas which the security team was not aware of.

2.b)

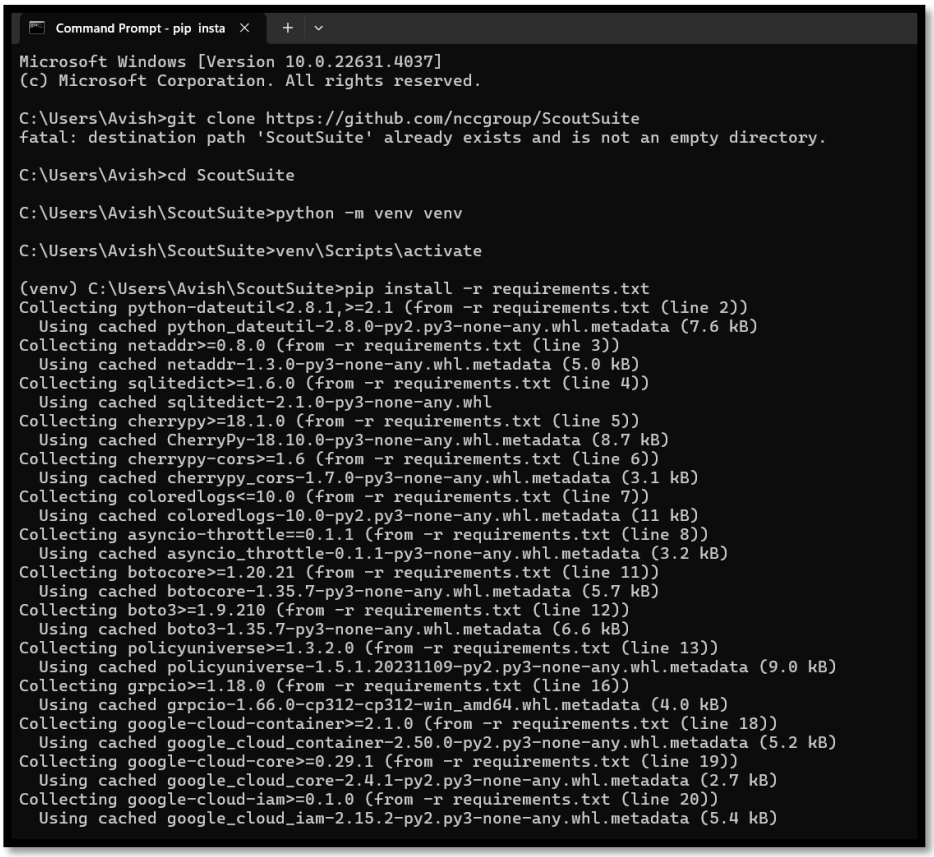


Figure 13 Downloading and configuring ScoutSuite 1.

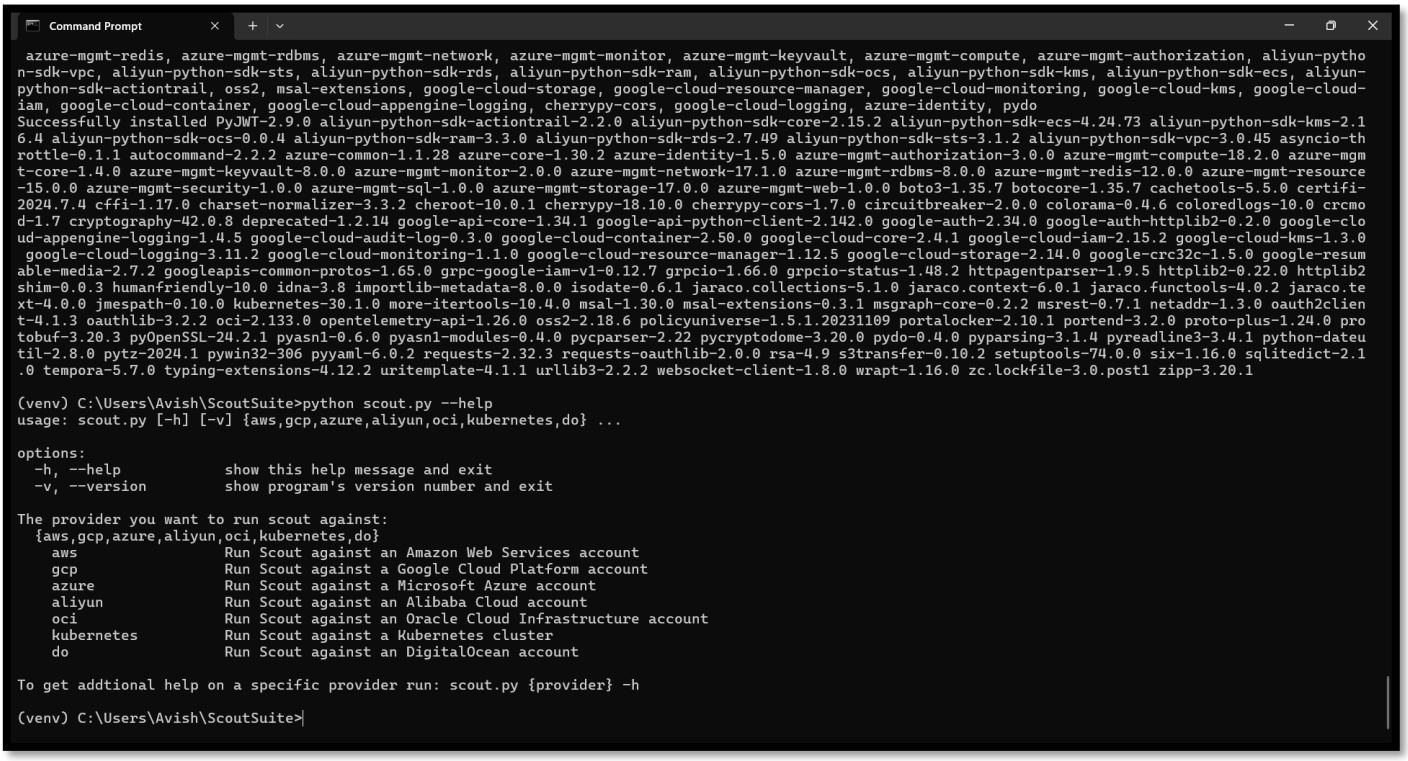


Figure 14 Downloading and configuring ScoutSuite 2.

2.c)

```
(venv) C:\Users\Avish\ScoutSuite>aws configure
AWS Access Key ID [****************yKp0]: AKIAUBKFCOBFHBDDFVJ6
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [json]: json
```

Figure 15 Configuring AWS CLI 1.

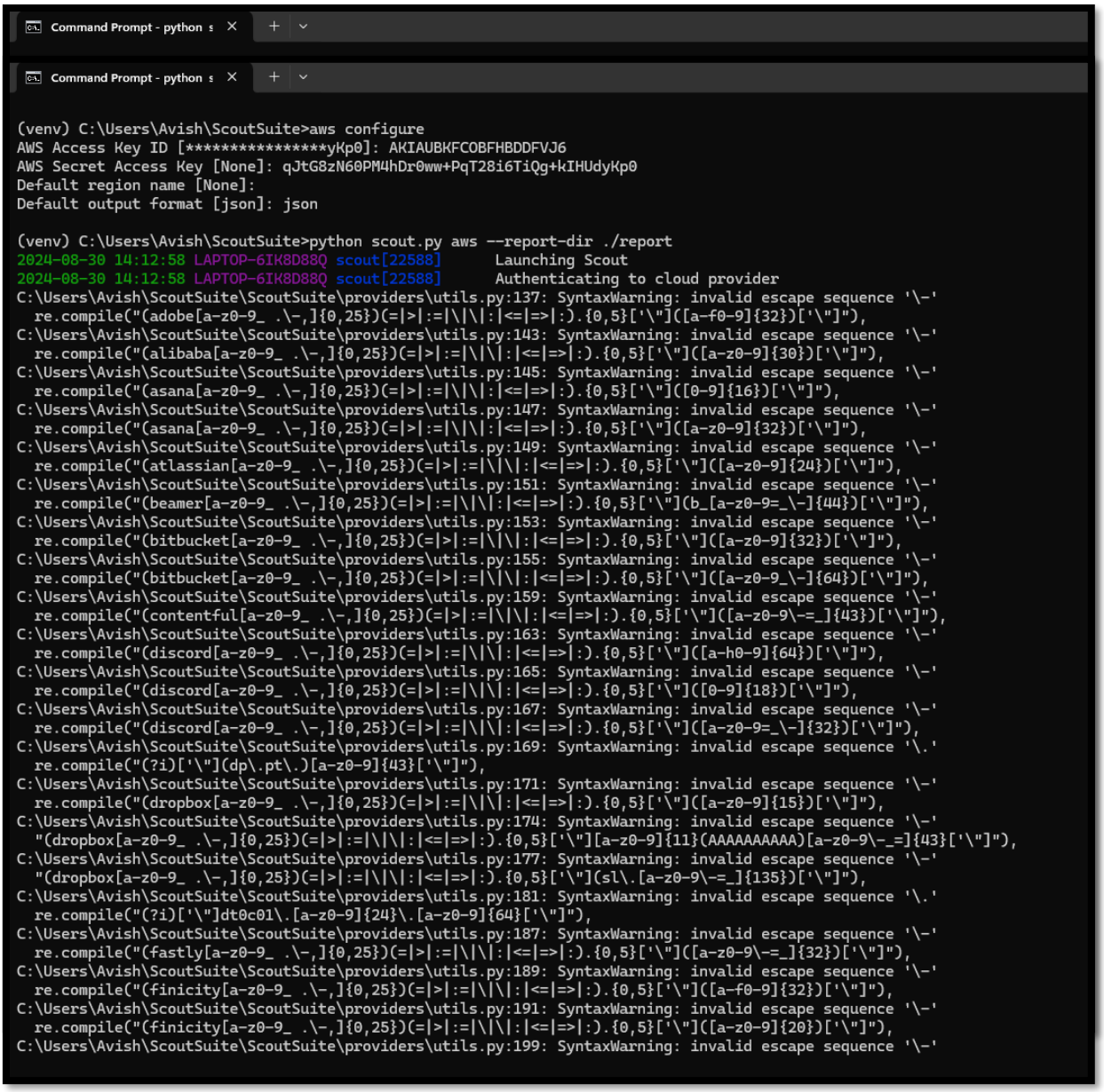


Figure 16 Configuring AWS CLI 2.

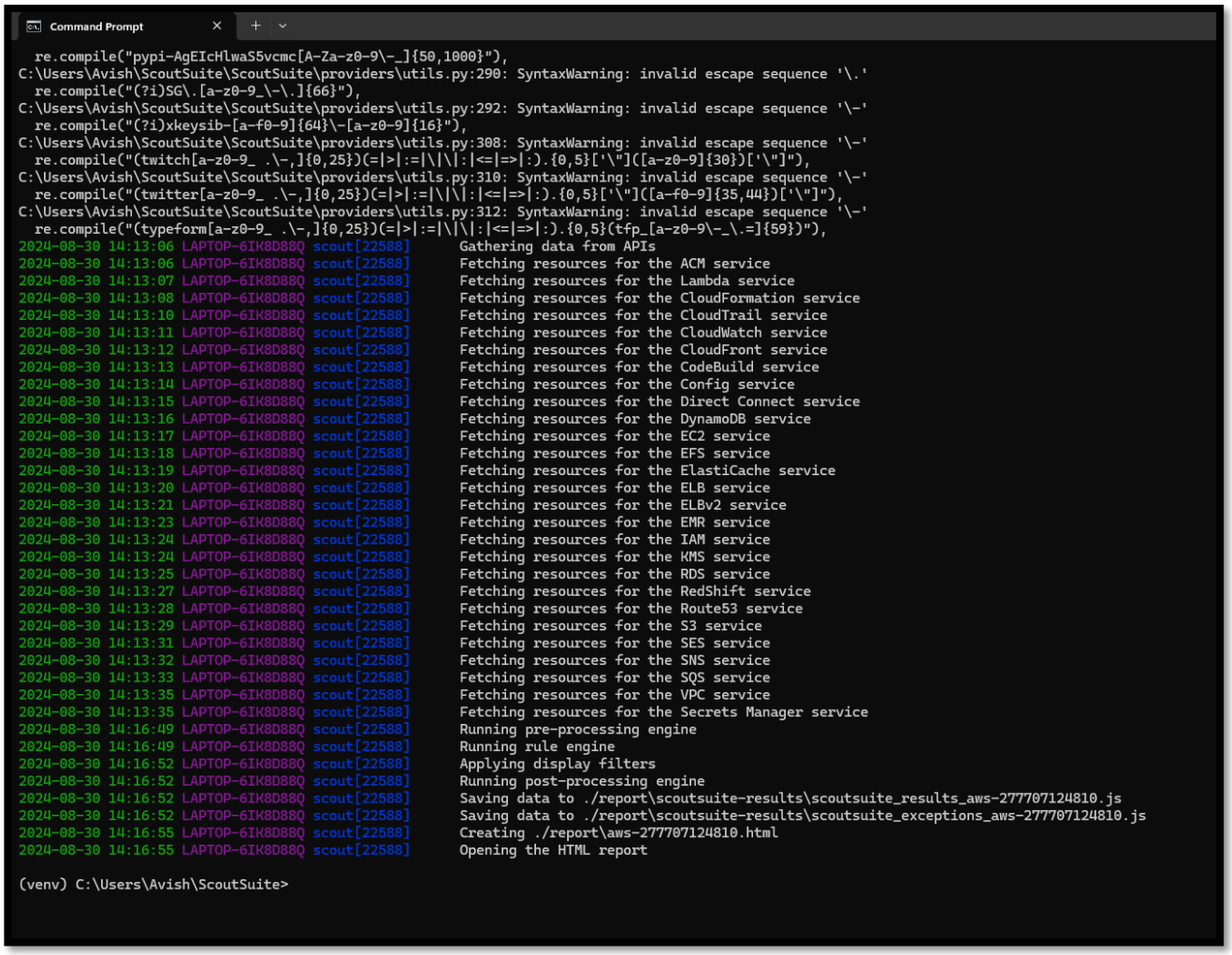


Figure 17 Configuring AWS CLI 3.

Generating the report in the “./report” directory.

2.d)

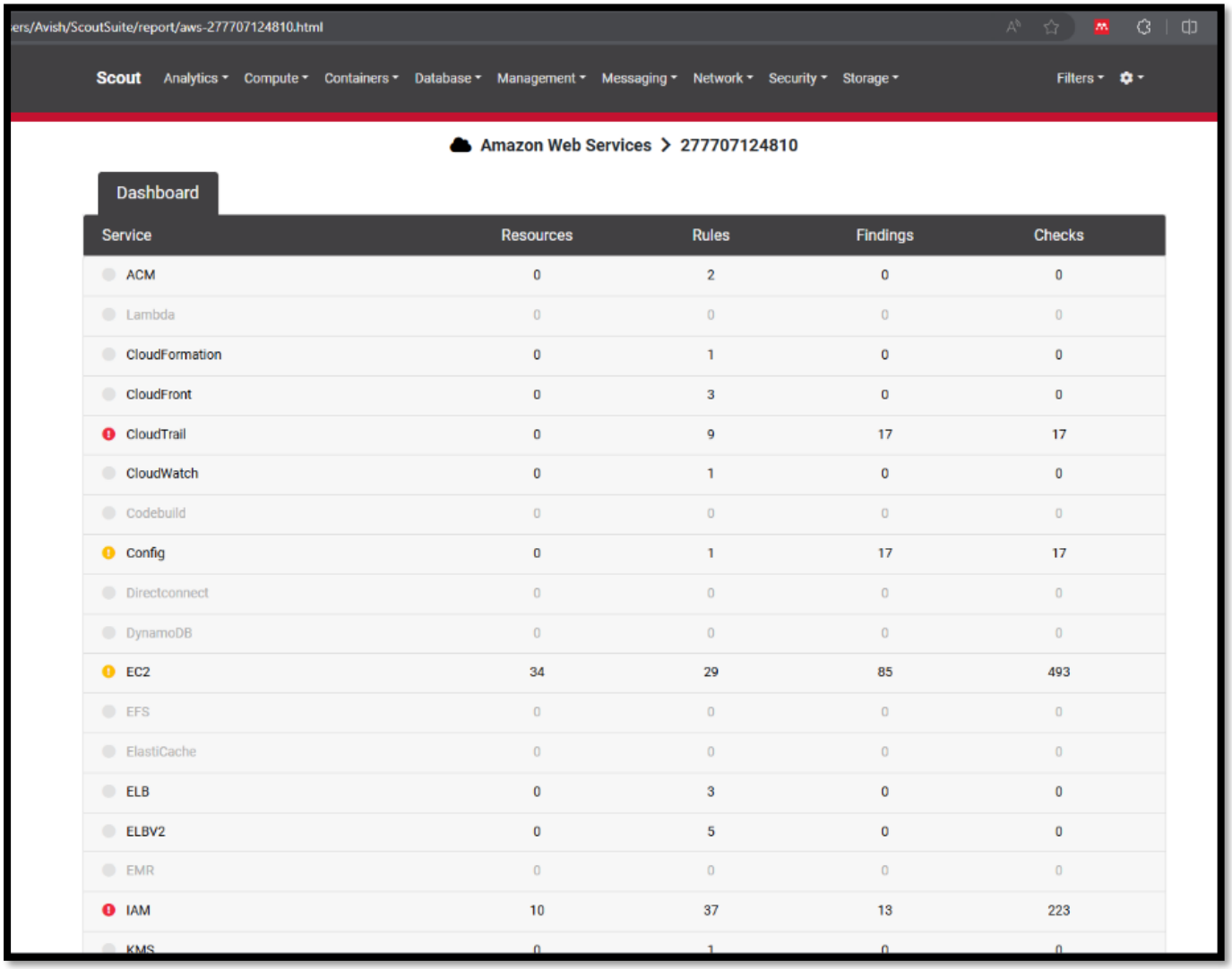


Figure 18 Generating ScoutSuite Report.

Running ScoutSuite against the provided AWS instance generated the above report.

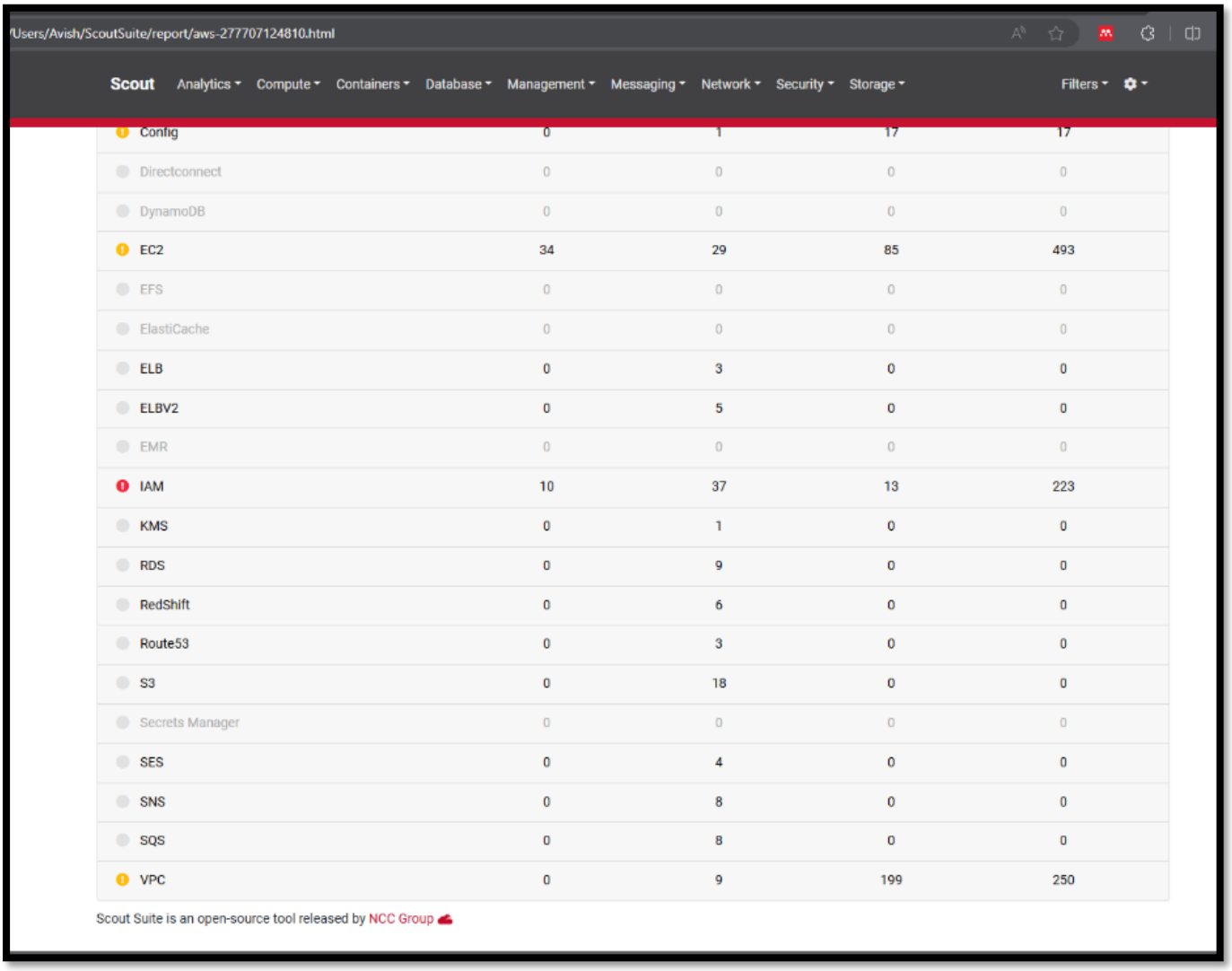


Figure 19 ScoutSuite Report.

AWS Security Assessment Report

INTRODUCTION

We conducted a security assessment of an Amazon Web Services (AWS) environment using ScoutSuite, an open-source security auditing tool. The primary focus of this assessment is to identify potential security risks within the AWS environment by evaluating various AWS services. An extensive analysis of the results, emphasising important security concerns and providing suggestions for reducing these risks, may be found in the document that follows.

BODY

ScoutSuite is a powerful security auditing tool that provides a detailed view of the security posture of cloud environments. It supports various cloud platforms, including AWS, Azure, and Google Cloud Platform. By scanning cloud configurations, ScoutSuite helps identify security misconfigurations, vulnerabilities, and compliance violations. We used ScoutSuite to scan and generate an in-depth report of the AWS environment. The report includes detailed information on resources, rules, findings, and checks performed for various AWS services such as IAM, EC2, CloudTrail, and VPC.

Summary of Findings

IAM (Identity and Access Management)

Resources: 10
Rules: 37
Findings: 13
Checks: 223
For controlling access to AWS resources, IAM is essential. The results point to possible security issues with IAM roles, policies, and permissions that, if not effectively controlled, could result in unauthorised access.

EC2

Resources: 34
Rules: 29
Findings: 85
Checks: 493
The core of cloud computing resources is EC2 instances. Potential weaknesses in EC2 instance configurations, such as exposed instances, open security groups, and unencrypted volumes, are indicated by the substantial number of discoveries.

CloudTrail

Resources: 0
Rules: 9

Findings: 17
Checks: 17
CloudTrail is essential for auditing and monitoring AWS account activities. The results draw attention to problems with incomplete or incorrectly set trails, which may affect the capacity to identify and react to suspicious activity.

Config

Resources: 0
Rules: 1
Findings: 17
Checks: 17
AWS Config helps track changes to resources and ensures compliance with desired configurations. The results imply that there may be compliance problems because some configurations are not being observed or followed.

VPC (Virtual Private Cloud)

Resources: 0
Rules: 9
Findings: 199
Checks: 250
VPCs form the network boundary for AWS resources. The results point to potential VPC configuration errors, including open ports, insufficient subnet security, and problems with network access rules.

Conclusion

Several serious security flaws in the AWS system were found by the ScoutSuite evaluation, mostly pertaining to the IAM, EC2, CloudTrail, Config, and VPC services. The AWS environment's overall security posture can be greatly strengthened by taking these results into consideration. To guarantee continued security and compliance, it is crucial to heed the advice given in this study and to regularly monitor and audit the AWS environment with ScoutSuite as it a useful tool to developers.

References

Das, D. 2023. Secure Your Mobile Apps with MobSF: A Comprehensive Guide to Android and iOS Security Analysis. [Online]. Medium. Available at: https://medium.com/@debasishkumardas5/secure-your-mobile-apps-with-mobsf-a-comprehensive-guide-to-android-and-ios-security-analysis-2ae7c928bf1d#:~:text=With%20MobSF%2C%20you%20can%20perform [Accessed 28 August 2024].

Fortinet. 2024. *What Is a Man-in-the Middle (MITM) Attack?* [Online] Fortinet. Available at: https://www.fortinet.com/resources/cyberglossary/man-in-the-middle-attack [Accessed 13 Aug. 2024].

Hasan, M.K., Habib, A.K.M.A., Islam, S., Safie, N., Abdullah, S.N.H.S. and Pandey, B. 2023. DDoS: Distributed denial of service attack in communication standard vulnerabilities in smart grid applications and cyber security with recent developments. *Energy Reports*, [online] 9(9), pp.1318–1326. doi:https://doi.org/10.1016/j.egyr.2023.05.184. [Accessed 13 Aug. 2024].

Imperva 2019. *What is Clickjacking | Attack Example | X-Frame-Options Pros & Cons | Imperva*. [Online] Learning Center. Available at: https://www.imperva.com/learn/application-security/clickjacking/ [Accessed 13 Aug. 2024].

OpenAI. 2024. Chat-GPT (Version 3.5). [Large language model]. Available at: https://chat.openai.com/ [Accessed: 28 August 2024].

Piktochart. n.d. *Log in and start using Piktochart's templates and drag-and-drop editor to create stunning visuals*. [Online] Available at: https://create.piktochart.com [Accessed: 13 August 2024].

Ajedi32. 2014. What role do hashes play in TLS/SSL certificate validation? Stack Exchange, 17 September 2014. Available at: https://security.stackexchange.com/questions/67512/what-role-do-hashes-play-in-tls-ssl-certificate-validation [Accessed 01 September 2024].

Cloudflare. 2024. What is TLS (Transport Layer Security)? Cloudflare, [n.d.]. Available at: https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/ [Accessed 01 September 2024].

Shivani7081. 2024. Transport Layer Security (TLS), GeeksForGeeks, 23 May 2024. Available at: https://www.geeksforgeeks.org/transport-layer-security-tls/ [Accessed 01 September 2024].

Stripe. 2023. Payment processing explained: What it is and how it works, Stripe, 10 July 2023. Available at:

https://stripe.com/resources/more/payment-processing-explained [Accessed 31 August 2024].

Cloudflare. 2024. What is a session key? Session keys and TLS handshakes, Cloudflare, [n.d.]. Available at: https://www.cloudflare.com/learning/ssl/what-is-a-session-key/ [Accessed 02 September 2024].

Linkedin. 2024. How do you secure user input in your code? Linkedin, [n.d.]. Available at: https://www.linkedin.com/advice/0/how-do-you-secure-user-input-your-code-skills-programming [Accessed 03 September 2023].

Mwema, E. 2024. Data Flow diagram. [Personal drawing]. Midrand: Unpublished.

Mwema, E. 2024. Data protection diagram. [Personal drawing]. Midrand: Unpublished.