

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

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

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

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

Contents

1. SECURITY SOLUTIONS	6
1.1 User Flow	6
1.2 Protecting input data in transit	7
1.3 Session Jacking attack.....	8
1.4. Click Jacking Attack.....	9
1.5 SQL Injection Attack	10
1.6 Cross Site Scripting Attack	11
1.7 Man in the Middle Attack	12
1.8 DDoS Attack	13
Question 2	14
2.a)	14
2.b)	17
2.c)	18
2.d)	20
AWS Security Assessment Report	22
References	24

Figures

Figure 1: Data Flow (Mwema, 2024)	6
Figure 2: Data Protection (Mwema, 2024)	7
Figure 3 Session Jacking	8
Figure 4 Click Jacking	9
Figure 5 SQL Injection (Imperva, 2019)	10
Figure 6 Cross Site Scripting	11
Figure 7 Man In the Middle (Imperva, 2019)	12
Figure 8 DDOS Attack (Imperva, 2019)	13
Figure 9 MobSF Running in Docker	14
Figure 10 MobSF Report	14
Figure 11 MobSF Grade for OPSC7311 Mobile Application	15
Figure 12 MobSF Findings	15
<i>Figure 13 Downloading and configuring ScoutSuite 1</i>	<i>17</i>
<i>Figure 14 Downloading and configuring ScoutSuite 2</i>	<i>17</i>
<i>Figure 15 Configuring AWS CLI 1</i>	<i>18</i>
<i>Figure 16 Configuring AWS CLI 2</i>	<i>18</i>
<i>Figure 17 Configuring AWS CLI 3</i>	<i>19</i>
<i>Figure 18 Generating ScoutSuite Report</i>	<i>20</i>
<i>Figure 19 ScoutSuite Report</i>	<i>21</i>

1. SECURITY SOLUTIONS

1.1 User Flow

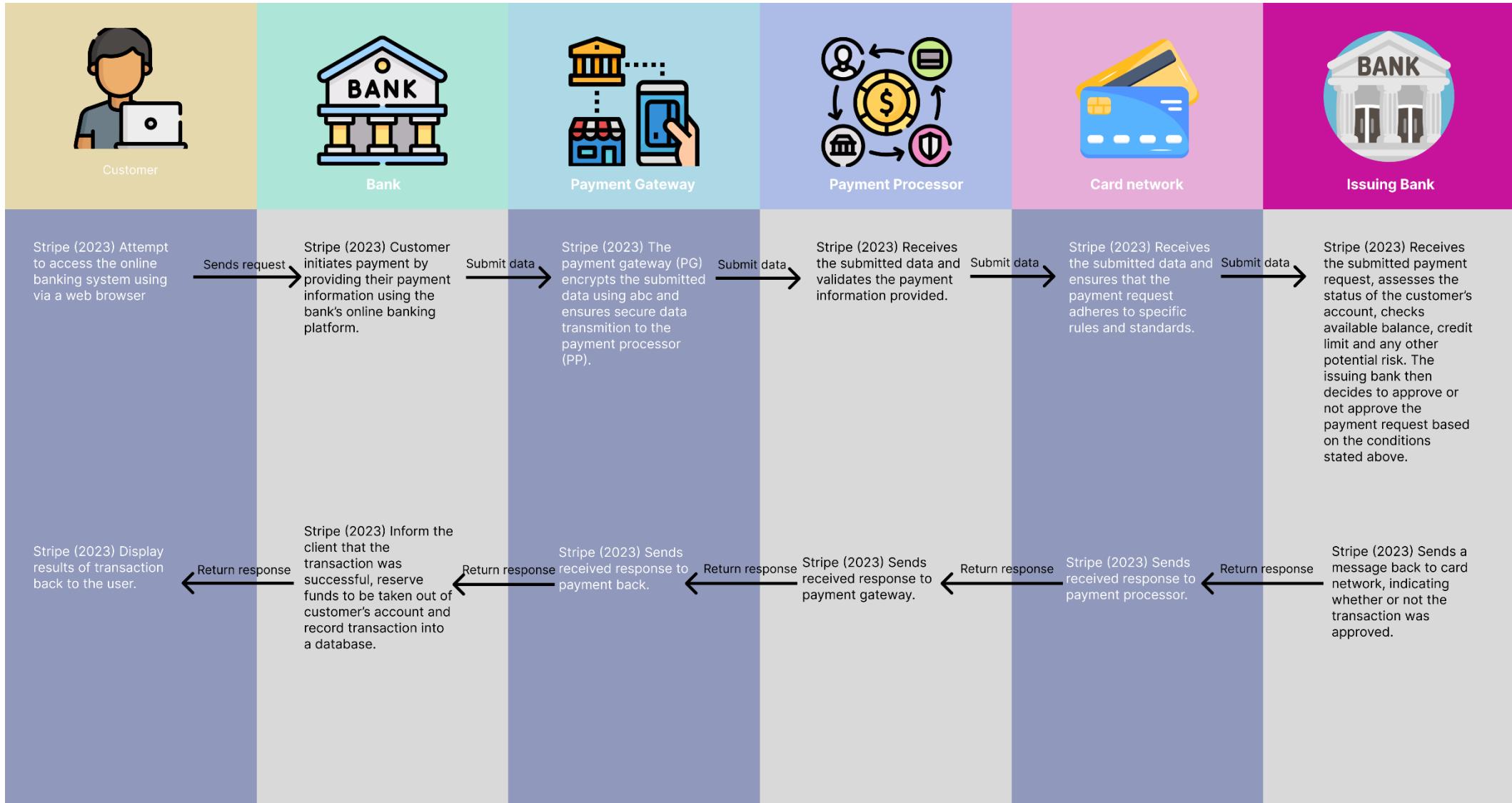


Figure 1: Data Flow (Mwema, 2024)

1.2 Protecting input data in transit



Figure 2: Data Protection (Eben, 2024)

1.3 Session Jacking attack

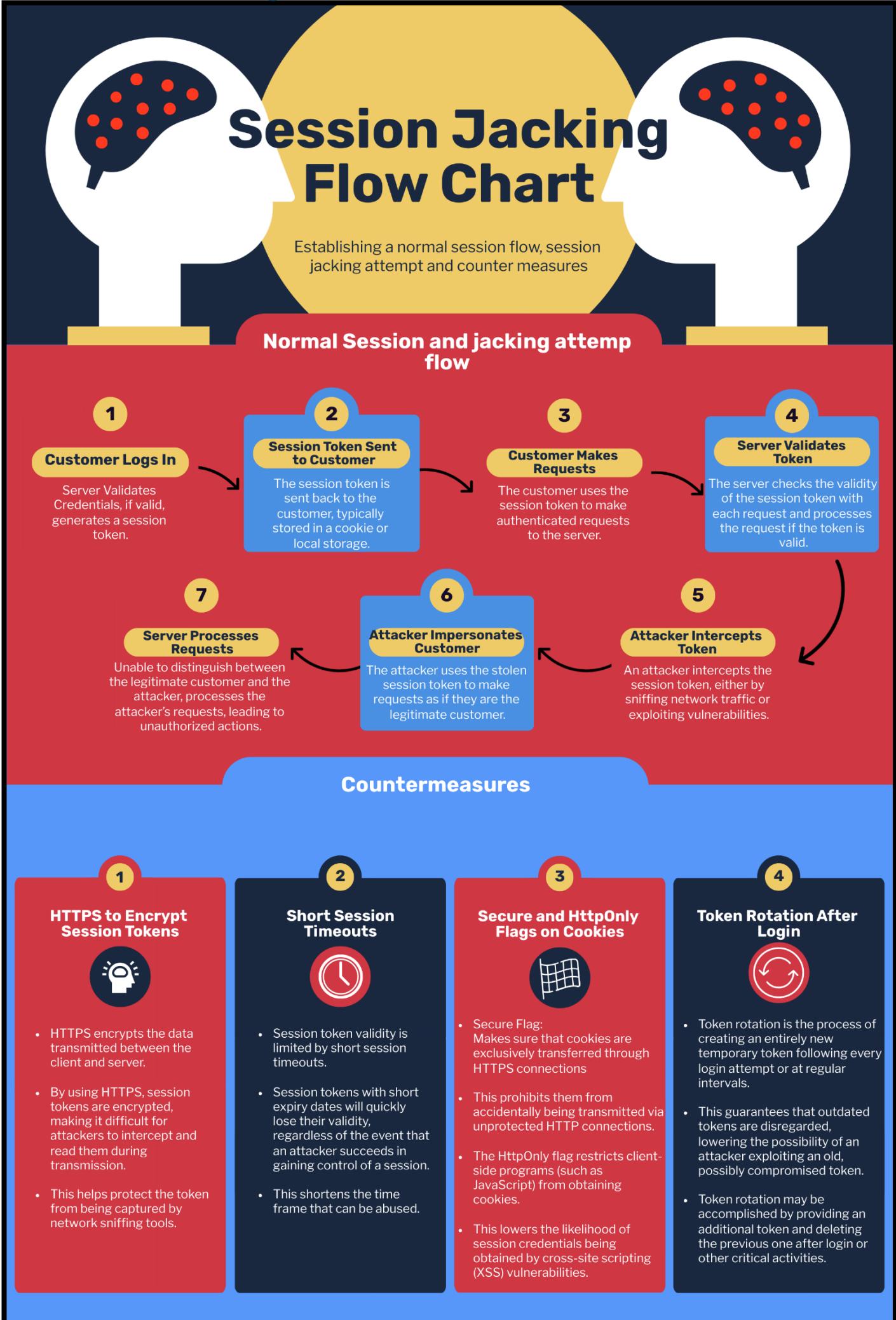


Figure 3 Session Jacking

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

2

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.

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.



3

4

User Visits Malicious Website

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

User visits malicious website

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



5

6

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.

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

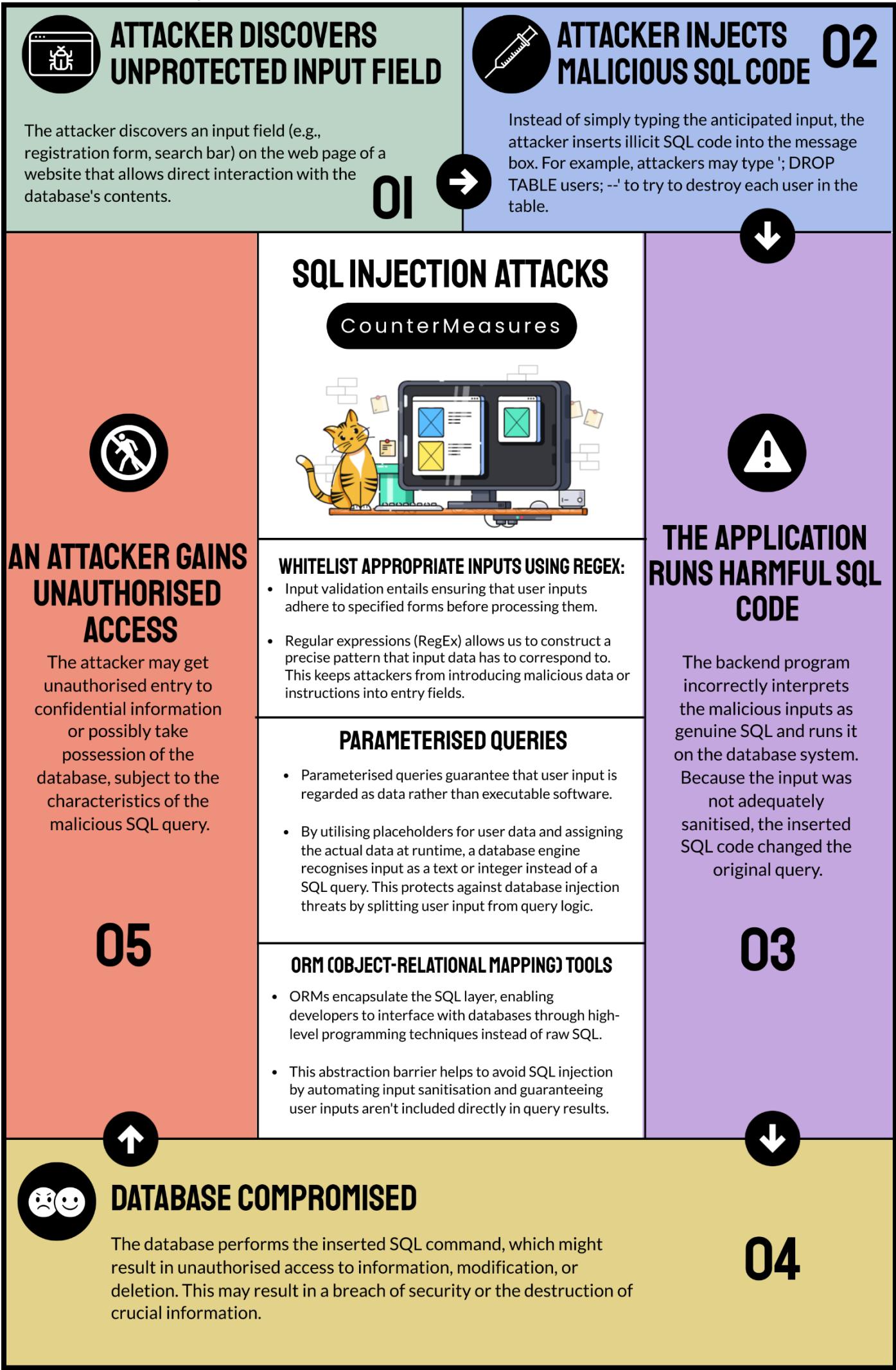


Figure 5 SQL Injection

1.6 Cross Site Scripting Attack

Cross-Site Scripting (XSS) attacks

XSS attacks allow attackers to insert client-side executable scripts into trustworthy web sites visited by other individuals.

Four Strategies to Countering this Attack



01

Output encoding

- Confirms that every user input presented on the website is correctly encrypted prior to rendering.
- This stops the web page from treating data entered by users as software that can be executed.
- Characters like <, >, &, and " require being translated to their respective HTML-encoded counterparts (<, >, &, ").

02

Content Security Policy (CSP)

- CSP is a strong security element that assists in preventing XSS by designating what material providers can be performed on our web page.
- For example, we may use CSP to limit the operation of JavaScript to scripts fetched from reliable sources while disallowing embedded scripts.

03

Sanitise and verify all user inputs.

- Before analysing any user input, completely sanitise and verify it.
- This involves removing or bypassing any possibly dangerous elements or scripts.
- For example, utilising a library like DOMPurify to sanitise HTML input can help avoid XSS attacks.

04

HTTPOnly and Encrypted Variables in Cookies:

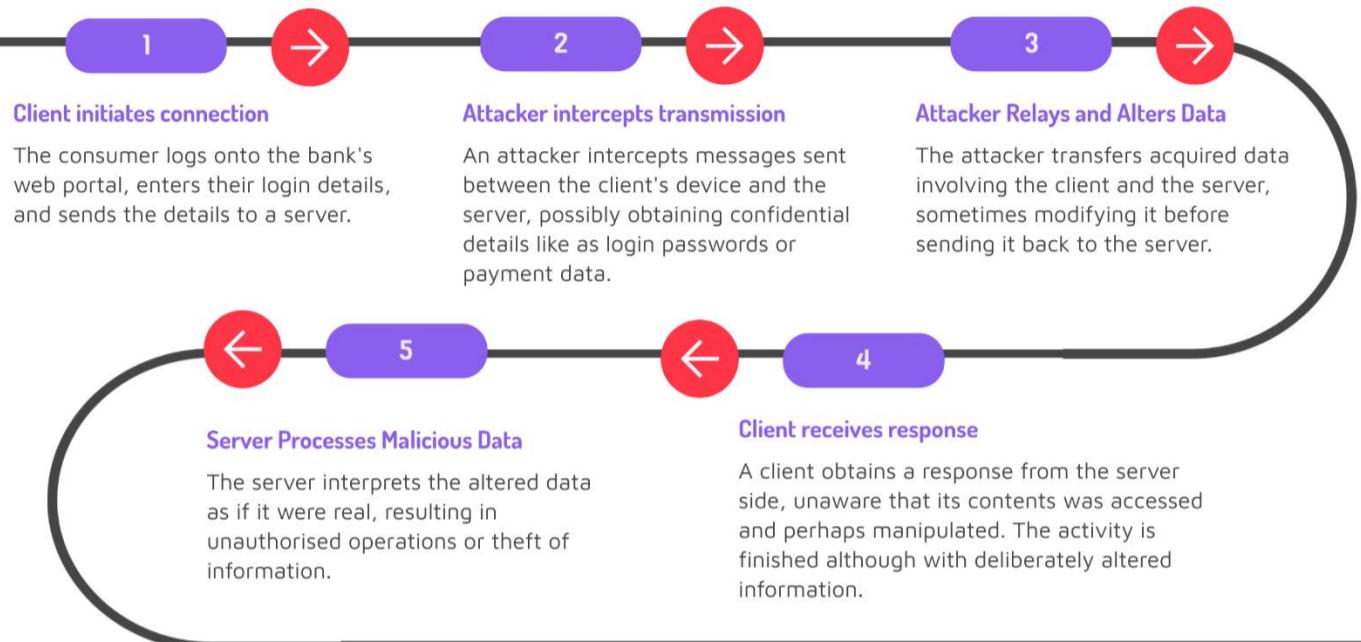
- Set the HttpOnly flag for cookies to avoid scripts running on the client from retrieving them.
- This reduces the danger of cookies being removed via XSS.
- Furthermore, using the Secure option guarantees that cookies are transmitted solely via HTTPS, preventing it from getting stolen during transit.

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.



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.



01

04

Countermeasures

02

03

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.

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.



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.

Figure 7 Man in the Middle (fortinet, 2024)

1.8 DDoS Attack

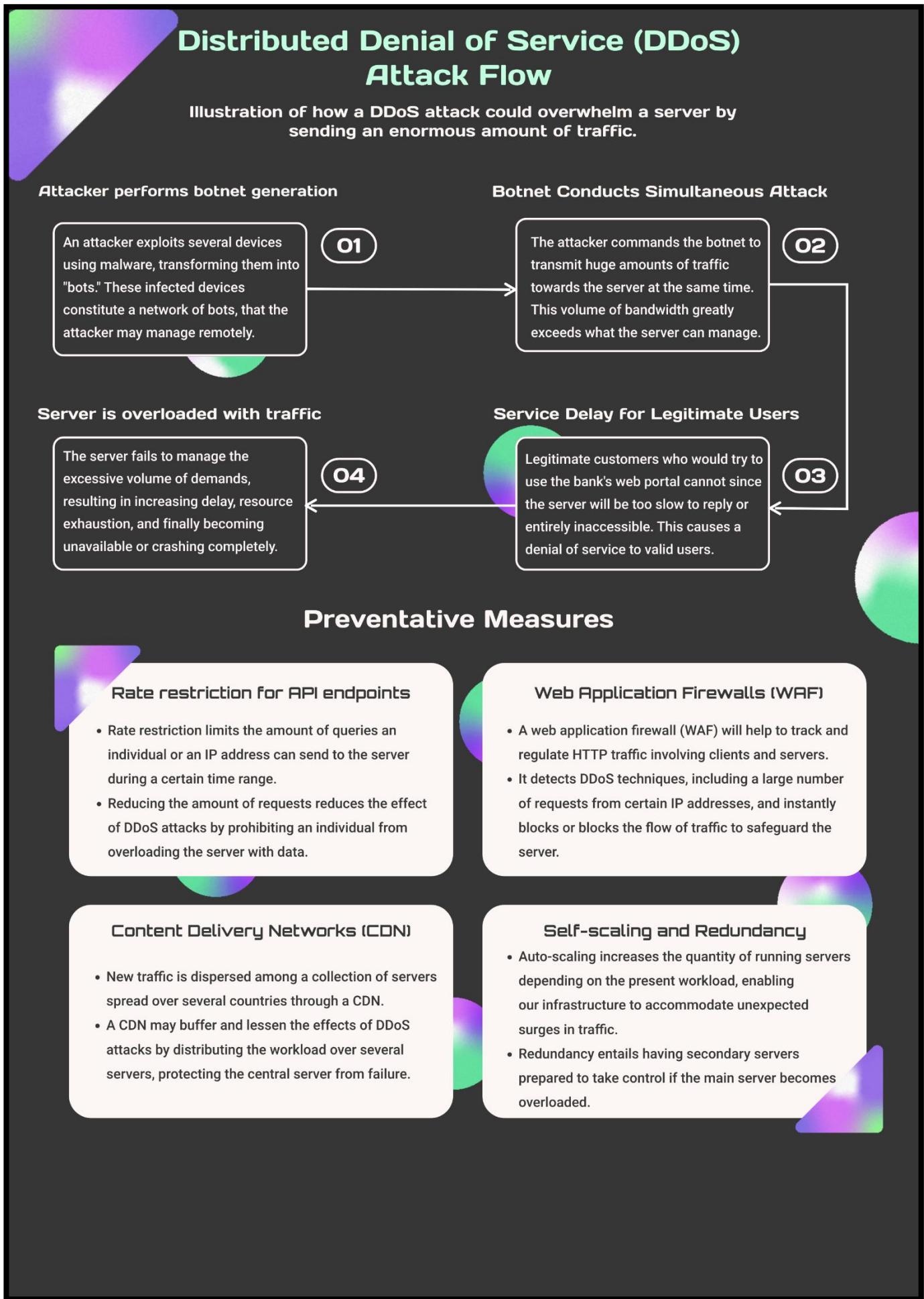


Figure 8 DDoS (Hasan et al., 2023)

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

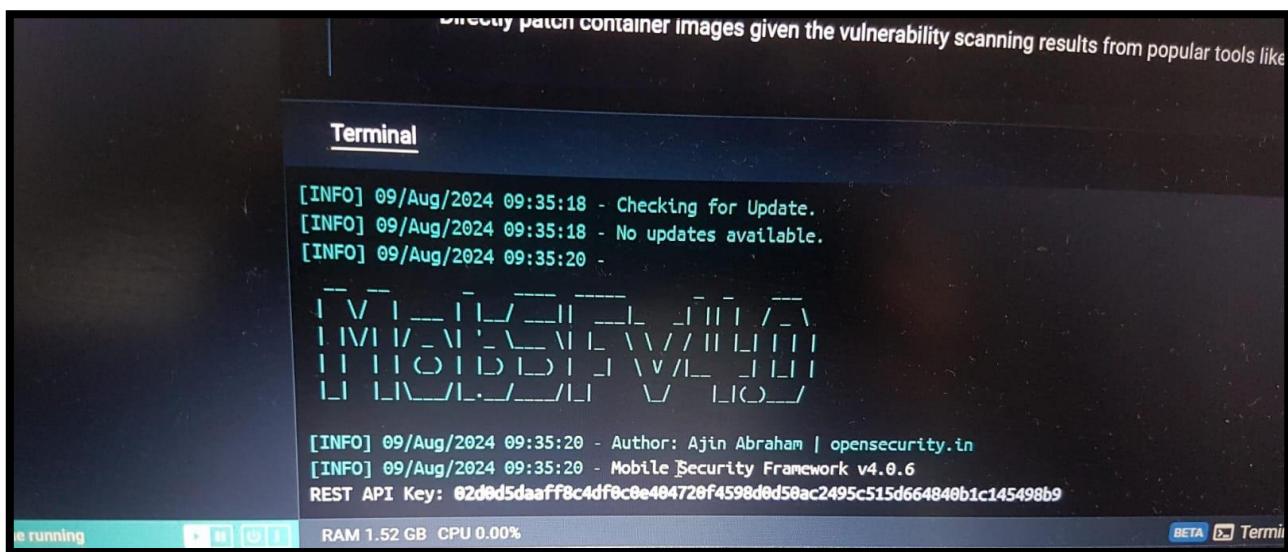


Figure 9 MobSF running in Docker

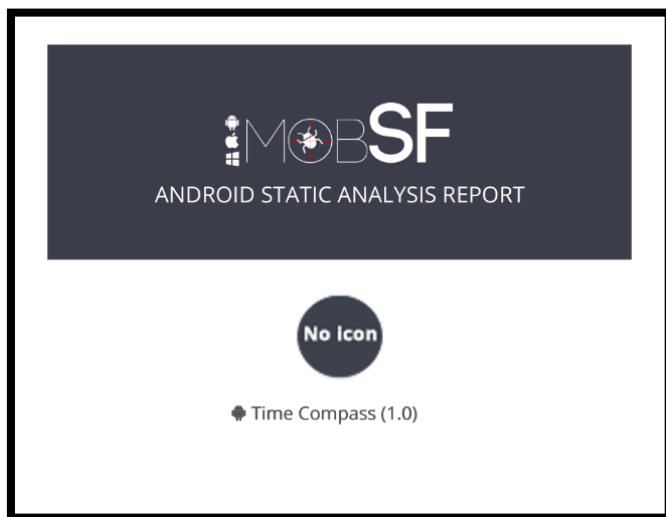


Figure 10 MobSF Report

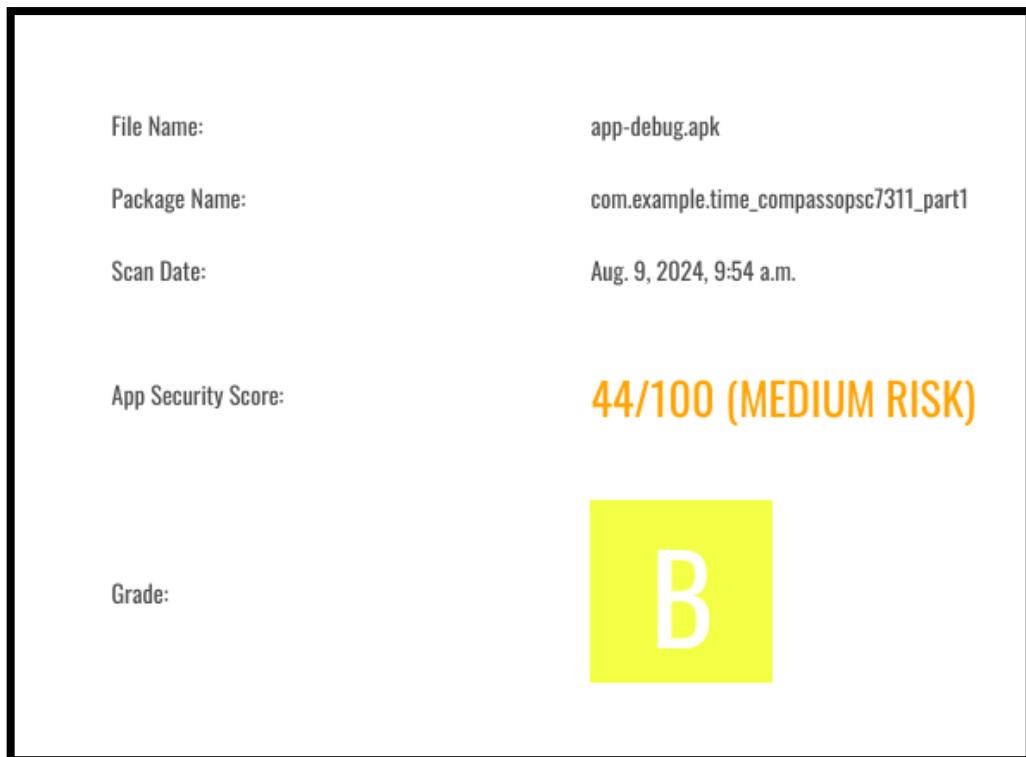


Figure 11 MobSF Grade for OPSC7311 Mobile Application

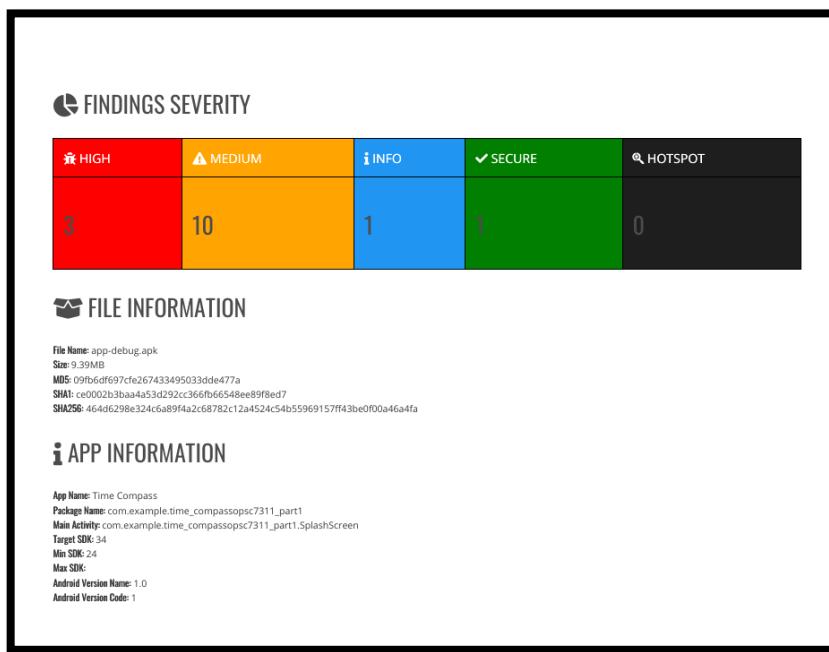


Figure 12 MobSF Findings

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)

```

Command Prompt - pip Insta  +  ▾

Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Avish>git clone https://github.com/nccgroup/ScoutSuite
fatal: destination path 'ScoutSuite' already exists and is not an empty directory.

C:\Users\Avish>cd ScoutSuite

C:\Users\Avish\ScoutSuite>python -m venv venv

C:\Users\Avish\ScoutSuite>venv\Scripts\activate

(venv) C:\Users\Avish\ScoutSuite>pip install -r requirements.txt
Collecting python-dateutil<2.8.1,>=2.1 (from -r requirements.txt (line 2))
  Using cached python_dateutil-2.8.0-py2.py3-none-any.whl.metadata (7.6 kB)
Collecting netaddr<=0.8.0 (from -r requirements.txt (line 3))
  Using cached netaddr-1.3.0-py3-none-any.whl.metadata (5.0 kB)
Collecting sqlitedict<=1.6.0 (from -r requirements.txt (line 4))
  Using cached sqlitedict-2.1.0-py3-none-any.whl
Collecting cherrypy<=18.1.0 (from -r requirements.txt (line 5))
  Using cached CherryPy-18.10.0-py3-none-any.whl.metadata (8.7 kB)
Collecting cherrypy-cors<=1.6 (from -r requirements.txt (line 6))
  Using cached cherrypy_cors-1.7.0-py3-none-any.whl.metadata (3.1 kB)
Collecting coloredlogs<=10.0 (from -r requirements.txt (line 7))
  Using cached coloredlogs-10.0-py2.py3-none-any.whl.metadata (11 kB)
Collecting asyncio-throttle<=0.1.1 (from -r requirements.txt (line 8))
  Using cached asyncio_throttle-0.1.1-py3-none-any.whl.metadata (3.2 kB)
Collecting botocore<=1.20.21 (from -r requirements.txt (line 11))
  Using cached botocore-1.35.7-py3-none-any.whl.metadata (5.7 kB)
Collecting boto3<=1.9.210 (from -r requirements.txt (line 12))
  Using cached boto3-1.35.7-py3-none-any.whl.metadata (6.6 kB)
Collecting policyuniverse<=1.3.2.0 (from -r requirements.txt (line 13))
  Using cached policyuniverse-1.5.1.20231109-py2.py3-none-any.whl.metadata (9.0 kB)
Collecting grpcio<=1.18.0 (from -r requirements.txt (line 16))
  Using cached grpcio-1.66.0-cp312-cp312-win_amd64.whl.metadata (4.0 kB)
Collecting google-cloud-container<=2.1.0 (from -r requirements.txt (line 18))
  Using cached google_cloud_container-2.50.0-py2.py3-none-any.whl.metadata (5.2 kB)
Collecting google-cloud-core<=0.29.1 (from -r requirements.txt (line 19))
  Using cached google_cloud_core-2.4.1-py2.py3-none-any.whl.metadata (2.7 kB)
Collecting google-cloud-iam<=0.1.0 (from -r requirements.txt (line 20))
  Using cached google_cloud_iam-2.15.2-py2.py3-none-any.whl.metadata (5.4 kB)

```

Figure 13 Downloading and configuring ScoutSuite 1.

```

Command Prompt  -  □  △  ▾

azurerm-mgmt-redis, azurerm-mgmt-rdbms, azurerm-mgmt-network, azurerm-mgmt-monitor, azurerm-mgmt-keyvault, azurerm-mgmt-compute, azurerm-mgmt-authorization, aliyun-python-n-sdk-vpc, aliyun-python-sdk-sts, aliyun-python-sdk-rds, aliyun-python-sdk-ram, aliyun-python-sdk-ocs, aliyun-python-sdk-kms, aliyun-python-sdk-ecs, aliyun-python-sdk-actiontrail, oss2, msal-extensions, google-cloud-storage, google-cloud-resource-manager, google-cloud-monitoring, google-cloud-kms, google-cloud-iam, google-cloud-container, google-cloud-appengine-logging, cherrypy-cors, google-cloud-logging, azure-identity, pyo
Successfully installed PyJWT-2.9.0 aliyun-python-sdk-actiontrail-2.2.0 aliyun-python-sdk-core-2.15.2 aliyun-python-sdk-ecs-4.24.73 aliyun-python-sdk-kms-2.1
6.4 aliyun-python-sdk-ocs-0.0.4 aliyun-python-sdk-ram-3.3.0 aliyun-python-sdk-rds-2.7.49 aliyun-python-sdk-sts-3.1.2 aliyun-python-sdk-vpc-3.0.45
asyncio-th
rottle-0.1.1
autocommand-2.2.2
azure-common-1.1.28
azure-core-1.35.2
azurerm-mgmt-compute-18.2.0
azurerm-mgmt-identity-1.4.0
azurerm-mgmt-keyvault-8.0.0
azurerm-mgmt-monitor-2.0.0
azurerm-mgmt-network-17.1.0
azurerm-mgmt-rdbms-8.0.0
azurerm-mgmt-redis-12.0.0
azurerm-mgmt-resource-15.0.0
azurerm-mgmt-security-1.0.0
azurerm-mgmt-sql-1.0.0
azurerm-mgmt-storage-17.0.0
azurerm-mgmt-web-1.0.0
boto3-1.35.7
botocore-1.35.7
cachetools-5.5.0
certifi-2024.7.4
cffi-1.17.0
charset-normalizer-3.3.2
cheroot-10.0.1
cherrypy-18.10.0
cherrypy-cors-1.7.0
circuitbreaker-2.0.0
colorama-0.4.6
coloredlogs-10.0
crcmod-1.7
cryptography-42.0.8
deprecated-1.2.14
google-api-core-1.34.1
google-api-python-client-2.142.0
google-auth-2.34.0
google-auth-httplib2-0.2.0
google-cloud-appengine-logging-1.4.5
google-cloud-audit-log-0.3.0
google-cloud-container-2.50.0
google-cloud-core-2.4.1
google-cloud-iam-2.15.2
google-cloud-kms-1.3.0
google-cloud-logging-3.1.2
google-cloud-monitoring-1.1.0
google-cloud-resource-manager-1.12.5
google-cloud-storage-2.14.0
google-crc32c-1.5.0
google-resumable-media-2.7.2
googleapis-common-protos-1.65.0
grpc-google-iam-v1-0.12.7
grpcio-1.66.0
grpcio-status-1.48.2
httpagentparser-1.9.5
httplib2-0.22.0
httpplib2-shim-0.3.0
humanfriendly-10.0
idna-3.8
importlib-metadata-8.0.0
isodate-0.6.1
jaraco.collections-5.1.0
jaraco.context-6.0.1
jaraco.functools-4.0.2
jaraco.te
xt-4.0.0
jmespath-0.10.0
kubernetes-30.1.0
more-itertools-10.4.0
msal-1.30.0
msal-extensions-0.3.1
msgraph-core-0.2.2
msrest-0.7.1
netaddr-1.3.0
oauth2client-4.1.3
oci-2.3.2
oci-2.133.0
opentelemetry-api-1.26.0
oss2-2.18.6
policyuniverse-1.5.1.20231109
portalocker-2.10.1
portend-3.2.0
proto-plus-1.24.0
protobuf-32.0.3
pyOpenSSL-24.2.1
pyasn1-0.6.0
pyasn1-modules-0.4.0
pycparser-2.22
pycryptodom-3.20.0
pydo-0.4.0
pyparsin-3.1.4
pyreadline3-3.4.1
python-dateutil-2.8.0
pytz-2024.1
pywin32-306
pyyaml-6.0.2
requests-oauthlib-2.0.0
rsa-4.9
s3transfer-0.10.2
setupools-74.0.0
six-1.16.0
sqlitedict-2.1.0
tempora-5.7.0
typing-extensions-4.12.2
uritemplate-4.1.1
urllib3-2.2.2
websocket-client-1.8.0
wrapt-1.16.0
zc.lockfile-3.0
postl-3.20.1
zipp-3.20.1

(venv) C:\Users\Avish\ScoutSuite>python scout.py --help
usage: scout.py [-h] [-v] {aws,gcp,azure,aliyun,oci,kubernetes,do} ...

options:
  -h, --help            show this help message and exit
  -v, --version          show program's version number and exit

The provider you want to run scout against:
{aws,gcp,azure,aliyun,oci,kubernetes,do}
  aws                  Run Scout against an Amazon Web Services account
  gcp                  Run Scout against a Google Cloud Platform account
  azure                Run Scout against a Microsoft Azure account
  aliyun               Run Scout against an Alibaba Cloud account
  oci                  Run Scout against an Oracle Cloud Infrastructure account
  kubernetes           Run Scout against a Kubernetes cluster
  do                   Run Scout against a DigitalOcean account

To get additional help on a specific provider run: scout.py {provider} -h
(venv) C:\Users\Avish\ScoutSuite>

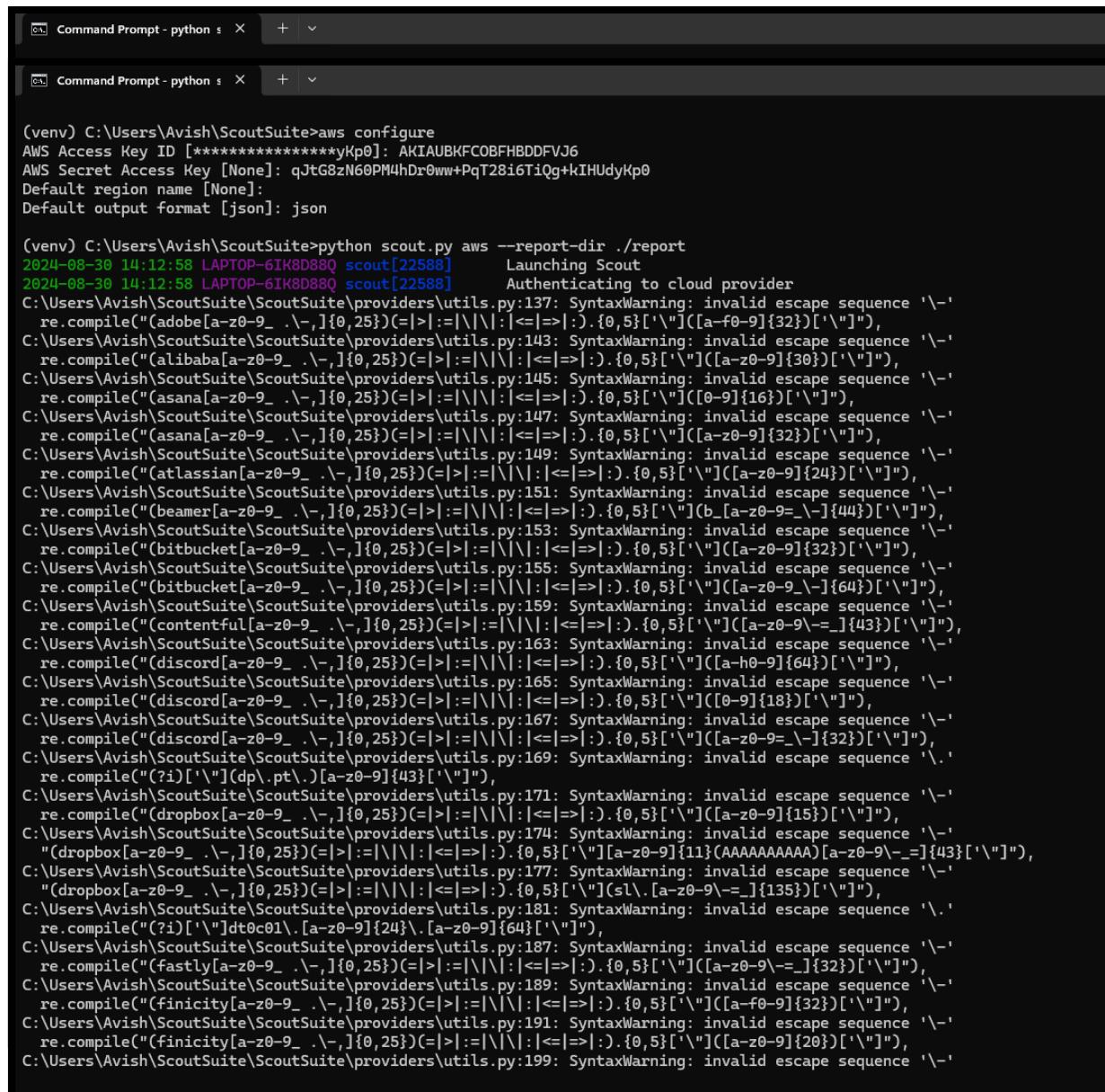
```

Figure 14 Downloading and configuring ScoutSuite 2.

2.c)

```
(venv) C:\Users\Avish\ScoutSuite>aws configure
AWS Access Key ID [*****yKp0]: AKIAUBKFCOBFBDDFVJ6
AWS Secret Access Key [None]: qJtG8zN60PM4hDr0ww+PqT28i6TiQg+kIHUdyKp0
Default region name [None]:
Default output format [json]: json
```

Figure 15 Configuring AWS CLI 1.



```
(venv) C:\Users\Avish\ScoutSuite>aws configure
AWS Access Key ID [*****yKp0]: AKIAUBKFCOBFBDDFVJ6
AWS Secret Access Key [None]: qJtG8zN60PM4hDr0ww+PqT28i6TiQg+kIHUdyKp0
Default region name [None]:
Default output format [json]: json

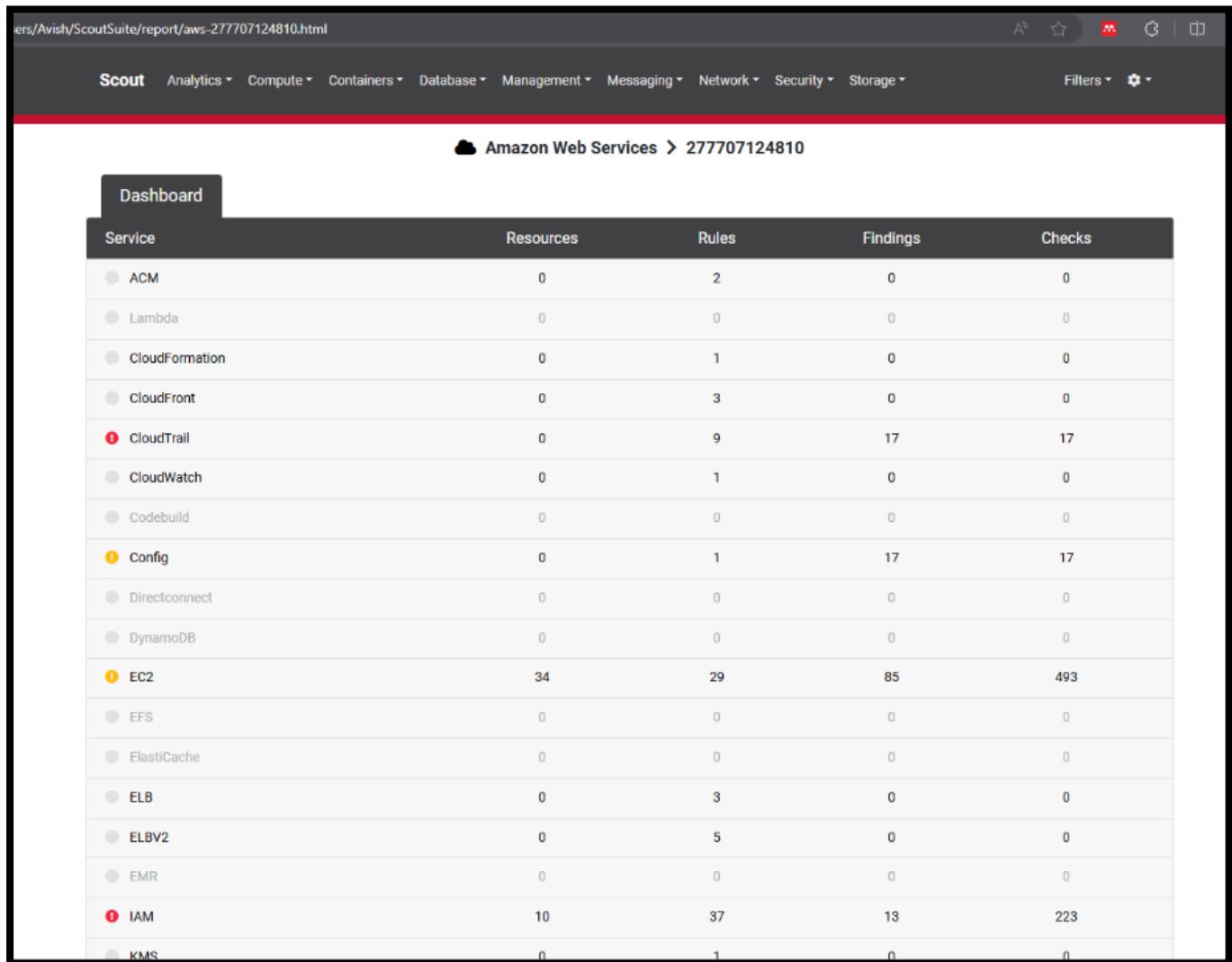
(venv) C:\Users\Avish\ScoutSuite>python scout.py aws --report-dir ./report
2024-08-30 14:12:58 LAPTOP-6IK8D88Q scout[22588]      Launching Scout
2024-08-30 14:12:58 LAPTOP-6IK8D88Q scout[22588]      Authenticating to cloud provider
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:137: SyntaxWarning: invalid escape sequence '\-'
re.compile("(adobe[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-f0-9]{32})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:143: SyntaxWarning: invalid escape sequence '\-'
re.compile("(alibaba[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9]{30})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:149: SyntaxWarning: invalid escape sequence '\-'
re.compile("(asana[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([0-9]{16})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:155: SyntaxWarning: invalid escape sequence '\-'
re.compile("(asana[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9]{32})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:159: SyntaxWarning: invalid escape sequence '\-'
re.compile("(atlassian[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9]{24})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:165: SyntaxWarning: invalid escape sequence '\-'
re.compile("(beamer[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([b-zA-Z0-9_]{44})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:171: SyntaxWarning: invalid escape sequence '\-'
re.compile("(bitbucket[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9]{32})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:177: SyntaxWarning: invalid escape sequence '\-'
re.compile("(bitbucket[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9_]{64})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:183: SyntaxWarning: invalid escape sequence '\-'
re.compile("(contentful[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9_-]{43})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:189: SyntaxWarning: invalid escape sequence '\-'
re.compile("(discord[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-h0-9]{64})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:195: SyntaxWarning: invalid escape sequence '\-'
re.compile("(discord[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([0-9]{18})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:201: SyntaxWarning: invalid escape sequence '\-'
re.compile("(discord[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9_]{32})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:207: SyntaxWarning: invalid escape sequence '\-'
re.compile("(?i)[\\"](\dp\\.pt\\ )[a-zA-Z0-9]{43}[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:213: SyntaxWarning: invalid escape sequence '\-'
re.compile("(dropbox[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9]{15})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:219: SyntaxWarning: invalid escape sequence '\-'
"(dropbox[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9]{11}(AAAAAAAAAA)[a-zA-Z0-9_-]{43})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:225: SyntaxWarning: invalid escape sequence '\-'
"(dropbox[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([s1\\.a-zA-Z0-9_-]{135})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:231: SyntaxWarning: invalid escape sequence '\-'
re.compile("(?i)[\\"](\dt0c01\\.a-zA-Z0-9){24}\\.[a-zA-Z0-9]{64}[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:237: SyntaxWarning: invalid escape sequence '\-'
re.compile("(fastly[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9_-]{32})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:243: SyntaxWarning: invalid escape sequence '\-'
re.compile("(finicity[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-f0-9]{32})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:249: SyntaxWarning: invalid escape sequence '\-'
re.compile("(finicity[a-zA-Z_][0-25])(=|>|=|\\"|:<|=|>|.).{0,5}[\\"]([a-zA-Z0-9]{20})[\\"]'),
C:\Users\Avish\ScoutSuite\ScoutSuite\providers\utils.py:255: SyntaxWarning: invalid escape sequence '\-
```

Figure 16 Configuring AWS CLI 2.

Figure 17 Configuring AWS CLI 3.

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

2.d)



The screenshot shows the ScoutSuite AWS report dashboard. At the top, there's a navigation bar with links for Analytics, Compute, Containers, Database, Management, Messaging, Network, Security, and Storage, along with filters and settings options. Below the navigation bar is a breadcrumb trail: Cloud Amazon Web Services > 277707124810. The main area is a table titled "Dashboard" with columns: Service, Resources, Rules, Findings, and Checks. The table lists various AWS services with their respective counts of resources, rules, findings, and checks. Notable entries include EC2 with 34 resources, 29 rules, 85 findings, and 493 checks, and IAM with 10 resources, 37 rules, 13 findings, and 223 checks.

Service	Resources	Rules	Findings	Checks
ACM	0	2	0	0
Lambda	0	0	0	0
CloudFormation	0	1	0	0
CloudFront	0	3	0	0
CloudTrail	0	9	17	17
CloudWatch	0	1	0	0
Codebuild	0	0	0	0
Config	0	1	17	17
Directconnect	0	0	0	0
DynamoDB	0	0	0	0
EC2	34	29	85	493
EFS	0	0	0	0
ElastiCache	0	0	0	0
ELB	0	3	0	0
ELBV2	0	5	0	0
EMR	0	0	0	0
IAM	10	37	13	223
KMS	0	1	0	0

Figure 18 Generating ScoutSuite Report.

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

AWS Service Report				
	Config	Compute	Network	Storage
Config	0	1	17	17
Directconnect	0	0	0	0
DynamoDB	0	0	0	0
EC2	34	29	85	493
EFS	0	0	0	0
ElastiCache	0	0	0	0
ELB	0	3	0	0
ELBV2	0	5	0	0
EMR	0	0	0	0
IAM	10	37	13	223
KMS	0	1	0	0
RDS	0	9	0	0
RedShift	0	6	0	0
Route53	0	3	0	0
S3	0	18	0	0
Secrets Manager	0	0	0	0
SES	0	4	0	0
SNS	0	8	0	0
SQS	0	8	0	0
VPC	0	9	199	250

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.