APDS7311

POE Task 1

GROUP MEMBERS

Mbali Banda ST10057677

Olwethu Ntobela ST10175937

Mosebyadi Legodi ST10192784

Palesa Sibuyi ST10115462

David Ledwaba ST10118368

Contents

Question 1

a. **Securing Input Information:**

Encryption is a technique for converting data into a code that prevents unwanted access. This entails transforming plain text data, such as credit card information, into a jumbled version known as ciphertext using encryption keys. To restore the data to its original form, a decryption procedure employs the associated decryption key. (Stripe, 2023)

When a consumer inputs payment information online, the data is encrypted before transmission. This manner, even if data is intercepted, it is illegible without the decryption key. Payment data, such as credit cards kept on ecommerce sites, can also be encrypted to improve security. (Stripe, 2023)
Core components: The gateway encrypts payment information to ensure data security during transmission.
Bank verification: The encrypted data is transferred to the customer's bank to ensure the availability of funds and the accuracy of payment information.
Transaction approval or denial: The bank responds to the company and the client, approving or disapproving the transaction.
Data Encryption: Use TLS (Transport Layer Security) for all data sent between the customer's browser and the bank server.
Input Validation: Use strong server-side input validation to protect against SQL injection, cross-site scripting (XSS), and other common vulnerabilities.
Password Hashing: Use powerful hashing algorithms such as bcrypt or Argon2 with a unique salt for each password.

b. **Securing Data in Transit:**

Protocols for Securing Data in Transit Secure protocols are used to encrypt data transfer files safely over the internet. Always ensure these protocols to securely transmit your data without any danger. There are four typical secure protocols for file transfer:

- Transport Layer Security (TLS)
- Secure File Transfer Protocol (SFTP)
- Secure Shell (SSH) File Transfer Protocol (SCP)
- Hypertext Transfer Protocol Secure (HTTPS)

TLS/SSL: TLS is a cryptographic protocol used to provide secure network communication. It is commonly used to encrypt data in transit, particularly when sensitive information, such as passwords, financial data, or personal information, must be communicated securely. (TitanFile, 2023)

API Security: Protect APIs using OAuth2.0, API keys, and rigorous access controls.

c.

i. **Portal Hardening:** (Nova, 2024)

- **Session Jacking:**

- **Enabling Two-factor Authentication**
  Enabling two-factor authentication has become the gold standard for avoiding session hijacking. If a user's credentials are hacked and hackers attempt to get into a site with two-factor authentication enabled, the attack will be blocked. This additional layer of security might be a one-time PIN number delivered by SMS or email. This extra authentication might be a code generated by your smartphone's Microsoft or Google authenticator.

- **Use HTTPS Everywhere**

  Enabling SSL and HTTPS instead of merely HTTP helps to prevent hackers from submitting XSS to your website. With HTTPS enabled, hackers will be unable to get the session ID. Enabling HTTPS also helps to prevent unauthorized access to stored cookies.

- Use session tokens with short expiration times and regenerate them after login.
- Implement multi-factor authentication (MFA) for both customer and staff logins.

- **Clickjacking Prevention:** (Chiarelli, 2020)
- To prevent the site from being embedded in an iFrame, use the X-Frame-Options HTTP header.
- Use Content Security Policy (CSP) headers to further restrict framing and embedding.

ii. Narayanan (2017) argues that to prevent Clickjacking attacks is by providing confirmation window for the click. If it is a different component the user can deny the engagement and report it. Another defence against clickjacking is frame busting, which prevents elements of an iFrame from appearing on a webpage, JavaScript can be used to accomplish this.
At page load time it will check whether the active page is the top-level in the browser window or not. Every authenticated request has a new HTTP called XFRAME-OPTIONS. The server should run in an HTML5 sandbox implementation, and it prevents any JavaScript from running on a server.

Tools that will be used to ensure that the web application is not vulnerable to Clickjacking attacks:

- Helmet.js: A Node.js middleware that can be used for setting several HTTP headers. For example: X-Frame Options.

iii. Steps on how to harden the portal against SQL Injected attacks:
Make use of stored procedures:

- Dynamic SQL should be replaced with stored procedures where possible.
- Implement Input Validation and Sanitization:
- Implement strict input validation to make sure only valid data is accepted.

Use parameterized queries:

- Make sure all SQL queries are parameterized to avoid direct

Tools to be used ensure that the web application is not vulnerable to SQL Injected attacks:

- SQLMap: An automated tool for identifying and taking advantage of SQL injection vulnerabilities (Indusface, 2024).

iv. Gurvinder (2014) argues that preventing Cross-Site Scripting consists of the following:
Content Security Policy:

- To lessen the impact of any XSS vulnerabilities that still occur.

Upon arrival, filter the input:

- When the user input is received, filter as strictly as possible based on expected or valid input.

Implement data on output:

- Encrypt the output to avoid it being interpreted as an active content. Depending on the output, this may require applying a combination of HTML, URL, JavaScript, and CSS encoding.

Tools to be used ensure that the web application is not vulnerable to Cross-Site Scripting attacks:

- ESAPI (OWASP Enterprise Security API)

v. Jared (2024) surmises that to prevent Man in the Middle attacks
Make use of strong TLS:

- Configure the server to make use of strong encryption protocols and ciphers.

Make use of HTTPS:

- Use SSL/TLS certificates to implement HTTPS throughout the entire application.

Make use of HTTP Strict Security Transport (HSTS):

- Configure HSTS so that the browsers can only connect to the website using HTTPS

Verify SSL certificates:

- To ensure that the connection secure make use of SSL certificates to validate the authenticity of the website. The data transmitted between the user and the server will be encrypted.

Tools to be used ensure that the web application is not vulnerable to Man in the Middle attacks:

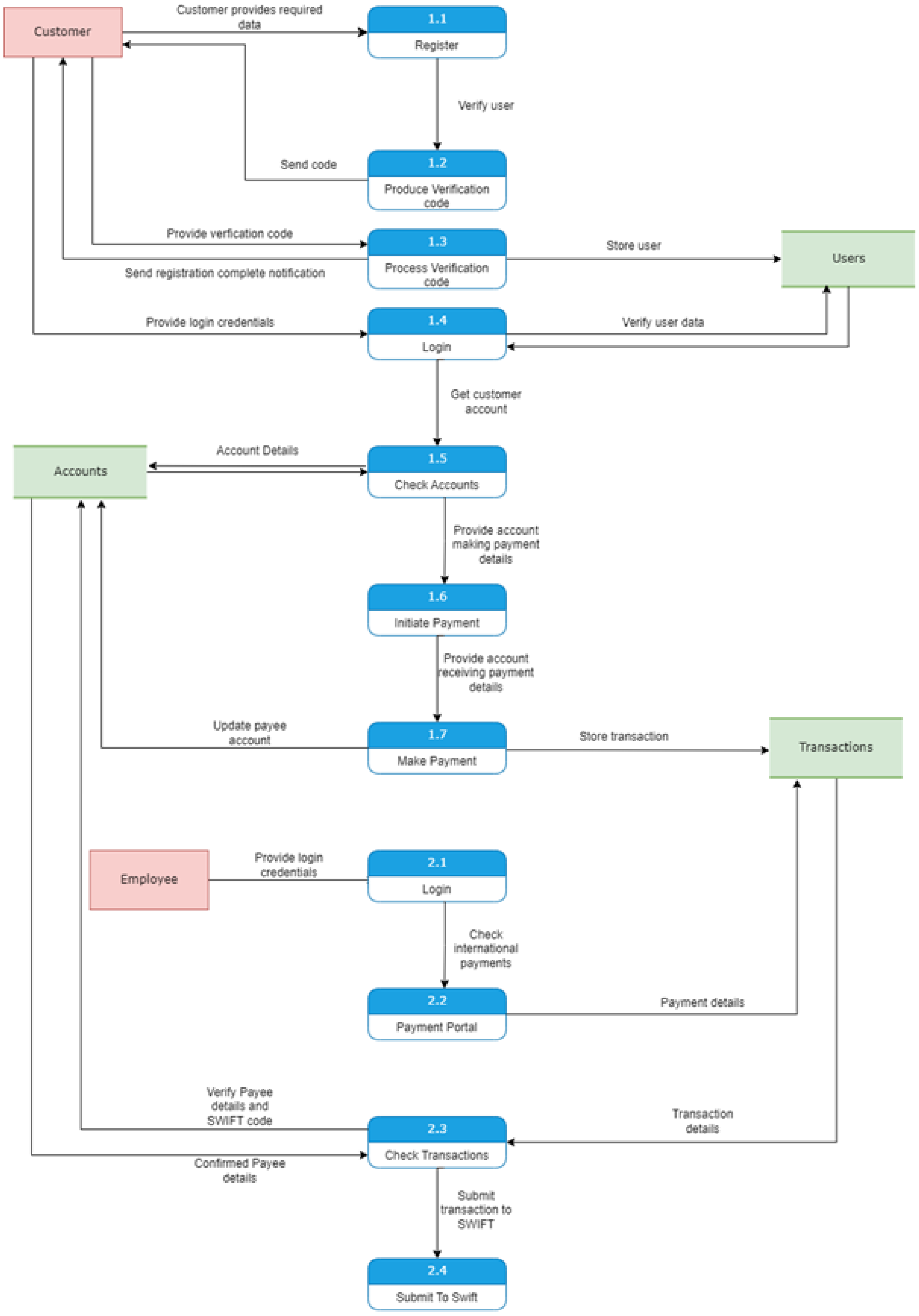
- OpenSSL: It will be used to generate and manage the SSL/TLS certificates

(Forbes Expert Panel, 2024)

vi. According to Kime, (2023) DDoS attacks can be prevented by doing the following:

- Harden against attacks: Deploy a firewall to filter out malicious traffic before it reaches the server**.**
- Implement Rate Limiting into Practice: Configure rate limiting to limit the number of requests from a single IP address.
- Monitor Traffic and Set Alerts: Continuously monitor traffic patterns and set alerts for unusual spikes

Tools to be used to ensure that the web application is not vulnerable to DDoS attacks.



Customer
Customer provides required data
1.1
Register
Verify user
Send code
1.2
Produce Verification code
Provide verfication code
1.3
Process Verification code
Send registration complete notification
Store user
Users
Provide login credentials
1.4
Login
Verify user data
Get customer account
Accounts
Account Details
1.5
Check Accounts
Provide account making payment details
1.6
Initiate Payment
Provide account receiving payment details
Update payee account
1.7
Make Payment
Store transaction
Transactions
Employee
Provide login credentials
2.1
Login
Check international payments
2.2
Payment Portal
Payment details
Verify Payee details and SWIFT code
2.3
Check Transactions
Transaction details
Confirmed Payee details
Submit transaction to SWIFT
2.4
Submit To Swift

Question 2

2A) **Report on the Use of Mobile Security Framework (MobSF)**

Introduction:

The Mobile Security Framework (MobSF) is a comprehensive tool designed to analyse the security of mobile applications, including Android, iOS, and Windows apps. Our team has been tasked with evaluating this tool to determine its suitability for integration into our hosting environment and future mobile application security protocols.

Configuration and Usage:

To assess MobSF's effectiveness, the tool was downloaded and configured following the official documentation provided at MobSF Documentation. The setup process was straightforward, and the tool was successfully integrated into our existing testing environment. For this evaluation, the *My Study Life* app was selected for analysis to evaluate the tool's capabilities in identifying potential security vulnerabilities.

Key Findings:

1. **Ease of Use:**
    - MobSF offers an intuitive web interface, making it accessible for both developers and security analysts. The tool supports drag-and-drop functionality for uploading mobile app packages (APKs, IPAs) for analysis, streamlining the initial steps of security testing.
2. **Comprehensive Analysis:**
    - The tool provides a wide range of security assessments, including static analysis, dynamic analysis, and malware analysis. For the *My Study Life* app, MobSF identified critical security issues such as insecure data storage, improper use of permissions, and potential vulnerabilities in third-party libraries.
3. **Detailed Reporting:**
    - MobSF generates detailed reports that include vulnerability descriptions, severity levels, and recommendations for remediation. This feature is particularly useful for developers who may need specific guidance on addressing the identified security issues in the *My Study Life* app.
4. **Integration Capabilities:**
    - MobSF supports integration with CI/CD pipelines, which is crucial for automating security testing in the development lifecycle. This feature

aligns with our organization's goal of maintaining continuous security checks as part of our DevSecOps practices.

5. **Community Support and Updates:**
    - As an open-source tool, MobSF benefits from active community support and regular updates. This ensures that the tool remains relevant in addressing new and emerging security threats.

Limitations:

1. **Resource Intensity:**
    - MobSF can be resource-intensive, particularly during dynamic analysis. This may impact the performance of the hosting environment, especially when analyzing large or complex applications like *My Study Life*.
2. **Learning Curve:**
    - While the tool is user-friendly, understanding the full extent of its capabilities requires some learning. Security teams may need additional training to maximize the tool’s potential.
3. **False Positives:**
    - In some cases, MobSF may report false positives, which could lead to unnecessary remediation efforts. It’s essential to validate the findings with additional tools or manual analysis.

Recommendation:

Based on the evaluation, MobSF demonstrates significant potential in enhancing our mobile application security practices. The tool’s ability to perform comprehensive security assessments, generate detailed reports, and integrate with CI/CD pipelines makes it an asset for our organization. However, it is essential to address the limitations, particularly the resource requirements and the potential for false positives, by complementing MobSF with additional security tools and practices.

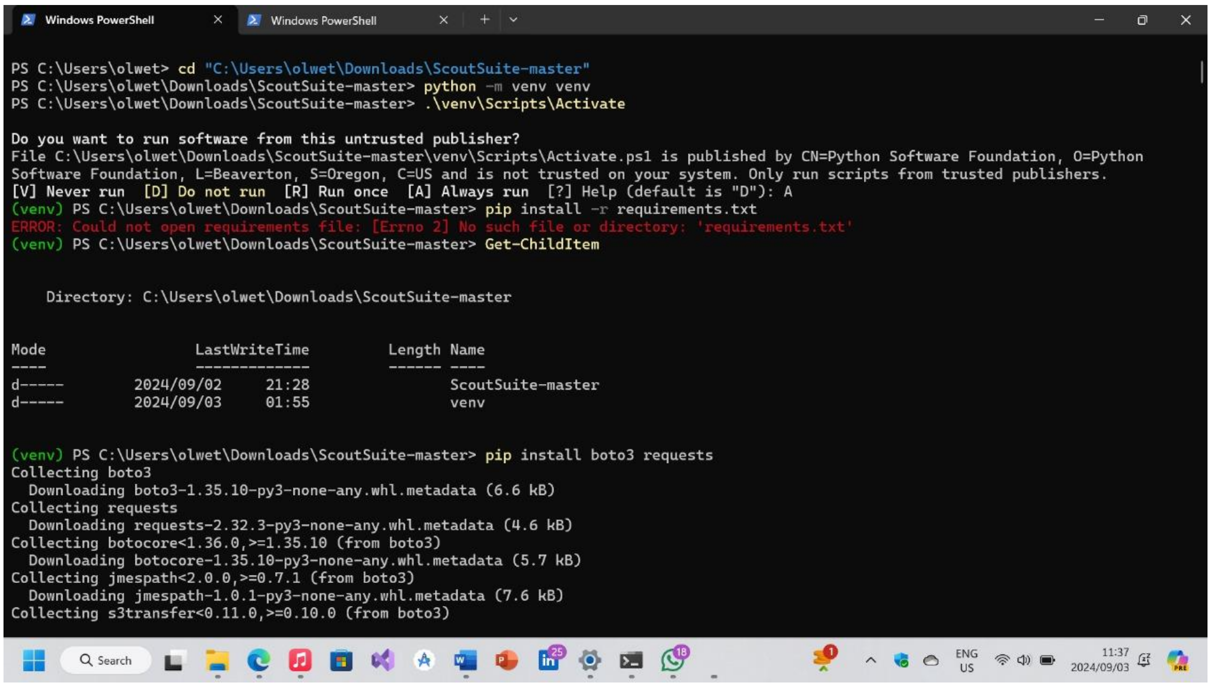
Conclusion:

I recommend adopting MobSF as part of our security toolkit, with the understanding that it should be used in conjunction with other security measures. The tool's strengths in identifying vulnerabilities early in the development process can help us achieve a higher standard of security for our mobile applications, including *My Study Life*, ultimately protecting our users and our infrastructure from potential threats.

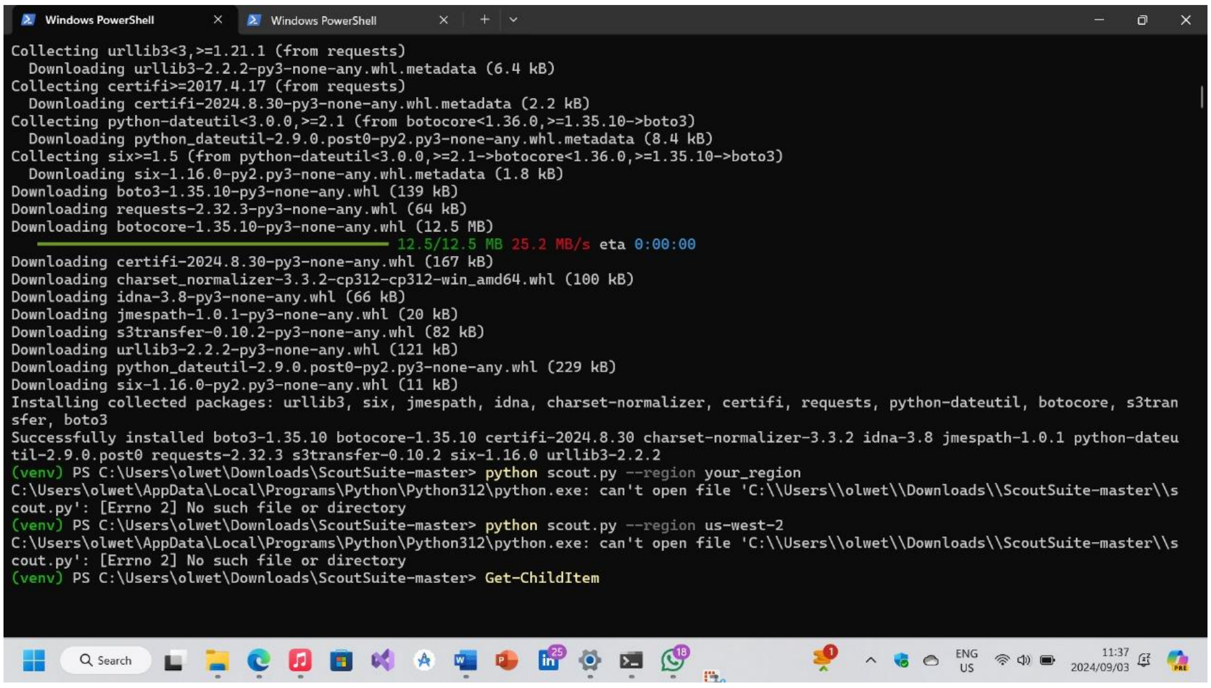
Reference:

This report was generated with the assistance of ChatGPT, an AI language model developed by OpenAI, which provided insights and guidance in analysing the findings related to the MobSF tool and formulating the recommendations.

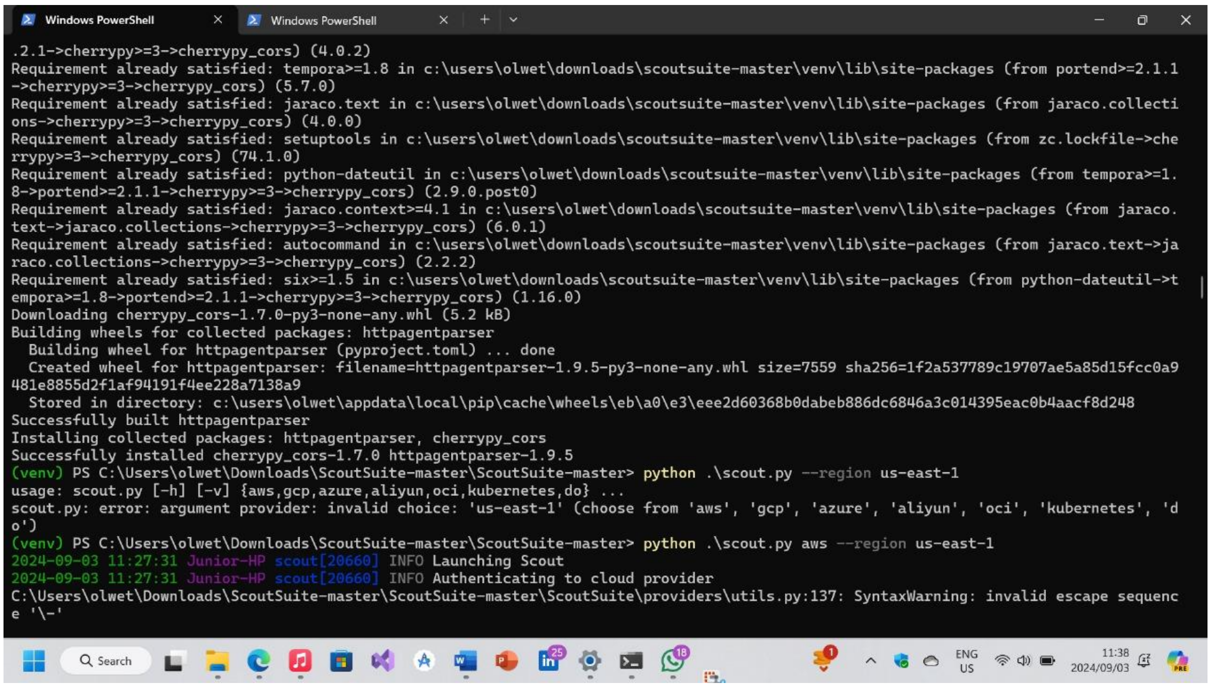
2b)



```
PS C:\Users\olwet> cd "C:\Users\olwet\Downloads\ScoutSuite-master"
PS C:\Users\olwet\Downloads\ScoutSuite-master> python -m venv venv
PS C:\Users\olwet\Downloads\ScoutSuite-master> .\venv\Scripts\Activate

Do you want to run software from this untrusted publisher?
File C:\Users\olwet\Downloads\ScoutSuite-master\venv\Scripts\Activate.ps1 is published by CN=Python Software Foundation, O=Python
Software Foundation, L=Beaverton, S=Oregon, C=US and is not trusted on your system. Only run scripts from trusted publishers.
[V] Never run  [D] Do not run  [R] Run once  [A] Always run  [?] Help (default is "D"): A
(venv) PS C:\Users\olwet\Downloads\ScoutSuite-master> pip install -r requirements.txt
ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.txt'
(venv) PS C:\Users\olwet\Downloads\ScoutSuite-master> Get-ChildItem


    Directory: C:\Users\olwet\Downloads\ScoutSuite-master


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2024/09/02     21:28                ScoutSuite-master
d-----        2024/09/03     01:55                venv


(venv) PS C:\Users\olwet\Downloads\ScoutSuite-master> pip install boto3 requests
Collecting boto3
  Downloading boto3-1.35.10-py3-none-any.whl.metadata (6.6 kB)
Collecting requests
  Downloading requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Collecting botocore<1.36.0,>=1.35.10 (from boto3)
  Downloading botocore-1.35.10-py3-none-any.whl.metadata (5.7 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from boto3)
  Downloading jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Collecting s3transfer<0.11.0,>=0.10.0 (from boto3)
```

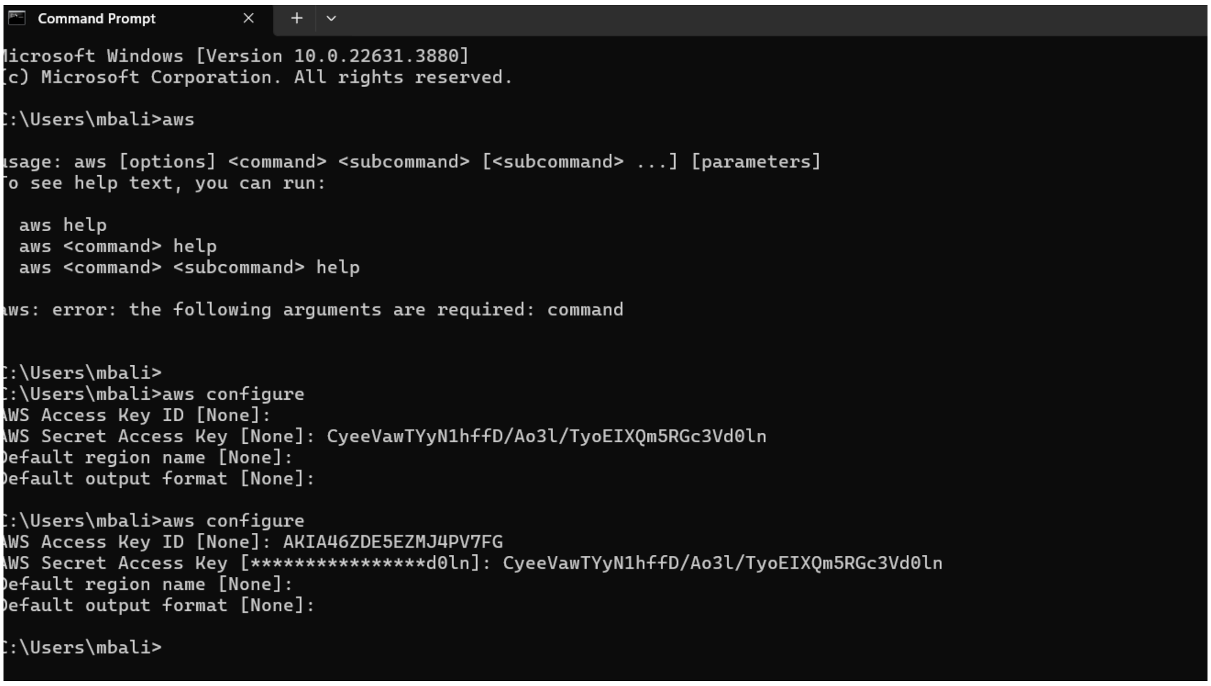


```
Collecting urllib3<3,>=1.21.1 (from requests)
  Downloading urllib3-2.2.2-py3-none-any.whl.metadata (6.4 kB)
Collecting certifi>=2017.4.17 (from requests)
  Downloading certifi-2024.8.30-py3-none-any.whl.metadata (2.2 kB)
Collecting python-dateutil<3.0.0,>=2.1 (from botocore<1.36.0,>=1.35.10->boto3)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->botocore<1.36.0,>=1.35.10->boto3)
  Downloading six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
Downloading boto3-1.35.10-py3-none-any.whl (139 kB)
Downloading requests-2.32.3-py3-none-any.whl (64 kB)
Downloading botocore-1.35.10-py3-none-any.whl (12.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.5/12.5 MB 25.2 MB/s eta 0:00:00
Downloading certifi-2024.8.30-py3-none-any.whl (167 kB)
Downloading charset_normalizer-3.3.2-cp312-cp312-win_amd64.whl (100 kB)
Downloading idna-3.8-py3-none-any.whl (66 kB)
Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Downloading s3transfer-0.10.2-py3-none-any.whl (82 kB)
Downloading urllib3-2.2.2-py3-none-any.whl (121 kB)
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: urllib3, six, jmespath, idna, charset-normalizer, certifi, requests, python-dateutil, botocore, s3tran
sfer, boto3
Successfully installed boto3-1.35.10 botocore-1.35.10 certifi-2024.8.30 charset-normalizer-3.3.2 idna-3.8 jmespath-1.0.1 python-dateu
til-2.9.0.post0 requests-2.32.3 s3transfer-0.10.2 six-1.16.0 urllib3-2.2.2
(venv) PS C:\Users\olwet\Downloads\ScoutSuite-master> python scout.py --region your_region
C:\Users\olwet\AppData\Local\Programs\Python\Python312\python.exe: can't open file 'C:\\Users\\olwet\\Downloads\\ScoutSuite-master\\s
cout.py': [Errno 2] No such file or directory
(venv) PS C:\Users\olwet\Downloads\ScoutSuite-master> python scout.py --region us-west-2
C:\Users\olwet\AppData\Local\Programs\Python\Python312\python.exe: can't open file 'C:\\Users\\olwet\\Downloads\\ScoutSuite-master\\s
cout.py': [Errno 2] No such file or directory
(venv) PS C:\Users\olwet\Downloads\ScoutSuite-master> Get-ChildItem
```



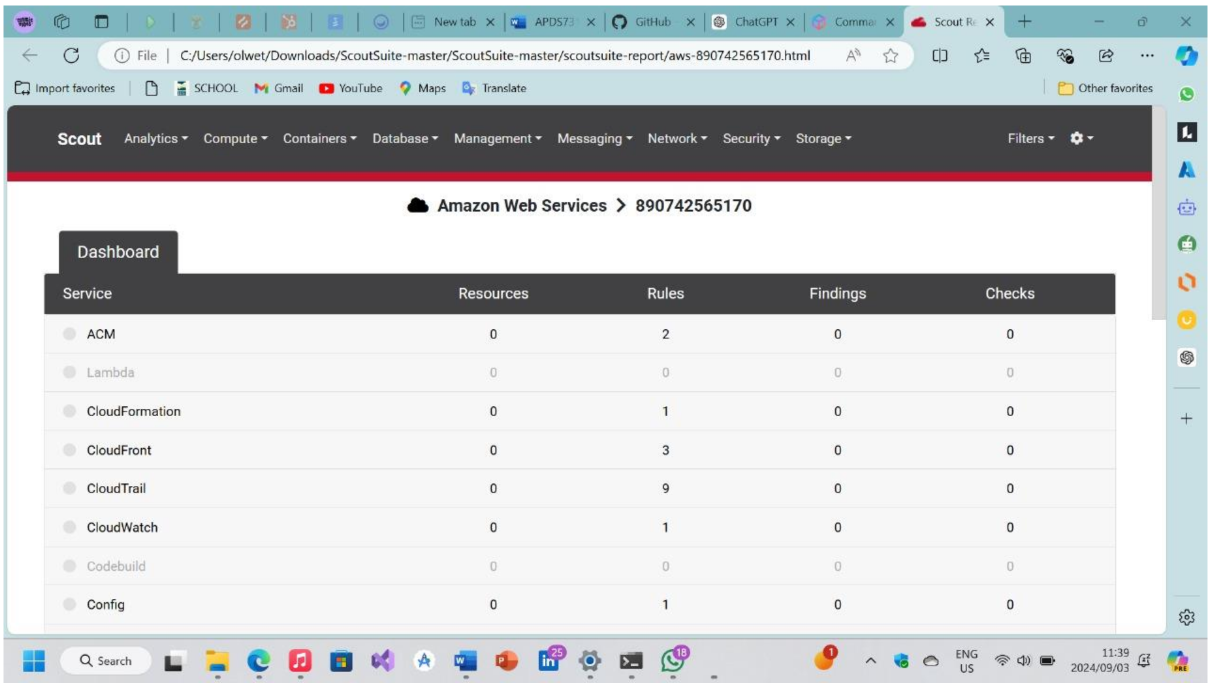
```
.2.1->cherrypy>=3->cherrypy_cors) (4.0.2)
Requirement already satisfied: tempora>=1.8 in c:\users\olwet\downloads\scoutsuite-master\venv\lib\site-packages (from portend>=2.1.1
->cherrypy>=3->cherrypy_cors) (5.7.0)
Requirement already satisfied: jaraco.text in c:\users\olwet\downloads\scoutsuite-master\venv\lib\site-packages (from jaraco.collecti
ons->cherrypy>=3->cherrypy_cors) (4.0.0)
Requirement already satisfied: setuptools in c:\users\olwet\downloads\scoutsuite-master\venv\lib\site-packages (from zc.lockfile->che
rrypy>=3->cherrypy_cors) (74.1.0)
Requirement already satisfied: python-dateutil in c:\users\olwet\downloads\scoutsuite-master\venv\lib\site-packages (from tempora>=1.
8->portend>=2.1.1->cherrypy>=3->cherrypy_cors) (2.9.0.post0)
Requirement already satisfied: jaraco.context>=4.1 in c:\users\olwet\downloads\scoutsuite-master\venv\lib\site-packages (from jaraco.
text->jaraco.collections->cherrypy>=3->cherrypy_cors) (6.0.1)
Requirement already satisfied: autocommand in c:\users\olwet\downloads\scoutsuite-master\venv\lib\site-packages (from jaraco.text->ja
raco.collections->cherrypy>=3->cherrypy_cors) (2.2.2)
Requirement already satisfied: six>=1.5 in c:\users\olwet\downloads\scoutsuite-master\venv\lib\site-packages (from python-dateutil->t
empora>=1.8->portend>=2.1.1->cherrypy>=3->cherrypy_cors) (1.16.0)
Downloading cherrypy_cors-1.7.0-py3-none-any.whl (5.2 kB)
Building wheels for collected packages: httpagentparser
  Building wheel for httpagentparser (pyproject.toml) ... done
  Created wheel for httpagentparser: filename=httpagentparser-1.9.5-py3-none-any.whl size=7559 sha256=1f2a537789c19707ae5a85d15fcc0a9
481e8855d2f1af94191f4ee228a7138a9
  Stored in directory: c:\users\olwet\appdata\local\pip\cache\wheels\eb\a0\e3\eee2d60368b0dabeb886dc6846a3c014395eac0b4aacf8d248
Successfully built httpagentparser
Installing collected packages: httpagentparser, cherrypy_cors
Successfully installed cherrypy_cors-1.7.0 httpagentparser-1.9.5
(venv) PS C:\Users\olwet\Downloads\ScoutSuite-master\ScoutSuite-master> python .\scout.py --region us-east-1
usage: scout.py [-h] [-v] {aws,gcp,azure,aliyun,oci,kubernetes,do} ...
scout.py: error: argument provider: invalid choice: 'us-east-1' (choose from 'aws', 'gcp', 'azure', 'aliyun', 'oci', 'kubernetes', 'd
o')
(venv) PS C:\Users\olwet\Downloads\ScoutSuite-master\ScoutSuite-master> python .\scout.py aws --region us-east-1
2024-09-03 11:27:31 Junior-HP scout[20660] INFO Launching Scout
2024-09-03 11:27:31 Junior-HP scout[20660] INFO Authenticating to cloud provider
C:\Users\olwet\Downloads\ScoutSuite-master\ScoutSuite-master\ScoutSuite\providers\utils.py:137: SyntaxWarning: invalid escape sequenc
e '\-'
```

2c)



```
icrosoft Windows [Version 10.0.22631.3880]
c) Microsoft Corporation. All rights reserved.

:\Users\mbali>aws

sage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
o see help text, you can run:

 aws help
 aws <command> help
 aws <command> <subcommand> help

ws: error: the following arguments are required: command


:\Users\mbali>
:\Users\mbali>aws configure
WS Access Key ID [None]:
WS Secret Access Key [None]: CyeeVawTYyN1hffD/Ao3l/TyoEIXQm5RGc3Vd0ln
efault region name [None]:
efault output format [None]:

:\Users\mbali>aws configure
WS Access Key ID [None]: AKIA46ZDE5EZMJ4PV7FG
WS Secret Access Key [****************d0ln]: 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
efault region name [None]:
efault output format [None]:

:\Users\mbali>
```

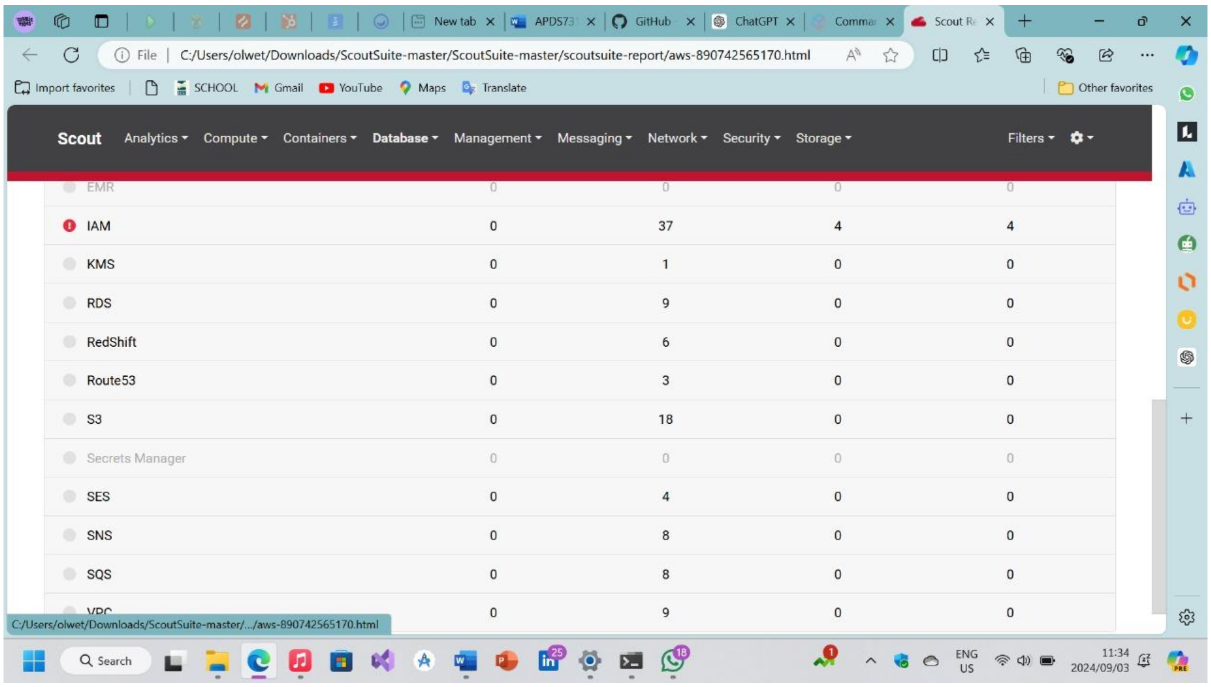
2d)


C:/Users/olwet/Downloads/ScoutSuite-master/ScoutSuite-master/scoutsuite-report/aws-890742565170.html
Scout
Analytics
Compute
Containers
Database
Management
Messaging
Network
Security
Storage
Filters

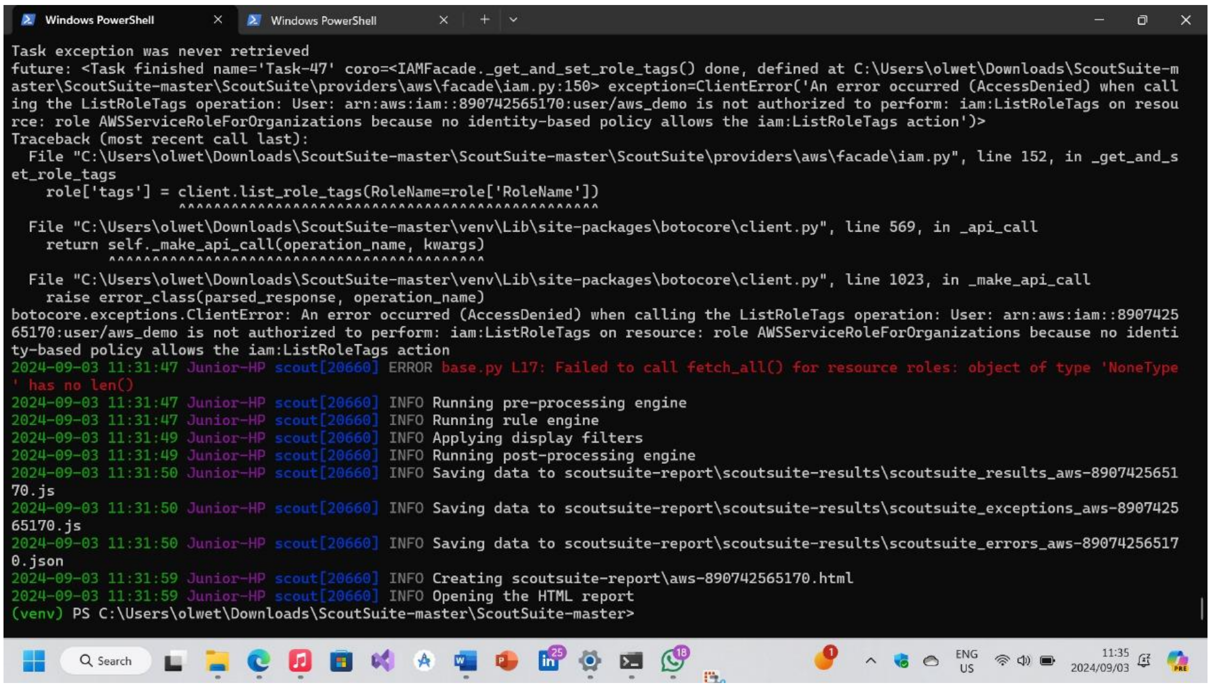
Amazon Web Services > 890742565170

Dashboard

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	0	0
CloudWatch	0	1	0	0
Codebuild	0	0	0	0
Config	0	1	0	0


C:/Users/olwet/Downloads/ScoutSuite-master/ScoutSuite-master/scoutsuite-report/aws-890742565170.html
Scout
Analytics
Compute
Containers
Database
Management
Messaging
Network
Security
Storage
Filters
C:/Users/olwet/Downloads/ScoutSuite-master/.../aws-890742565170.html

EMR	0	0	0	0
IAM	0	37	4	4
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	0	0



```
Task exception was never retrieved
future: <Task finished name='Task-47' coro=<IAMFacade._get_and_set_role_tags() done, defined at C:\Users\olwet\Downloads\ScoutSuite-m
aster\ScoutSuite-master\ScoutSuite\providers\aws\facade\iam.py:150> exception=ClientError('An error occurred (AccessDenied) when call
ing the ListRoleTags operation: User: arn:aws:iam::890742565170:user/aws_demo is not authorized to perform: iam:ListRoleTags on resou
rce: role AWSServiceRoleForOrganizations because no identity-based policy allows the iam:ListRoleTags action')>
Traceback (most recent call last):
  File "C:\Users\olwet\Downloads\ScoutSuite-master\ScoutSuite-master\ScoutSuite\providers\aws\facade\iam.py", line 152, in _get_and_s
et_role_tags
    role['tags'] = client.list_role_tags(RoleName=role['RoleName'])
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\olwet\Downloads\ScoutSuite-master\venv\Lib\site-packages\botocore\client.py", line 569, in _api_call
    return self._make_api_call(operation_name, kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\olwet\Downloads\ScoutSuite-master\venv\Lib\site-packages\botocore\client.py", line 1023, in _make_api_call
    raise error_class(parsed_response, operation_name)
botocore.exceptions.ClientError: An error occurred (AccessDenied) when calling the ListRoleTags operation: User: arn:aws:iam::8907425
65170:user/aws_demo is not authorized to perform: iam:ListRoleTags on resource: role AWSServiceRoleForOrganizations because no identi
ty-based policy allows the iam:ListRoleTags action
2024-09-03 11:31:47 Junior-HP scout[20660] ERROR base.py L17: Failed to call fetch_all() for resource roles: object of type 'NoneType
' has no len()
2024-09-03 11:31:47 Junior-HP scout[20660] INFO Running pre-processing engine
2024-09-03 11:31:47 Junior-HP scout[20660] INFO Running rule engine
2024-09-03 11:31:49 Junior-HP scout[20660] INFO Applying display filters
2024-09-03 11:31:49 Junior-HP scout[20660] INFO Running post-processing engine
2024-09-03 11:31:50 Junior-HP scout[20660] INFO Saving data to scoutsuite-report\scoutsuite-results\scoutsuite_results_aws-8907425651
70.js
2024-09-03 11:31:50 Junior-HP scout[20660] INFO Saving data to scoutsuite-report\scoutsuite-results\scoutsuite_exceptions_aws-8907425
65170.js
2024-09-03 11:31:50 Junior-HP scout[20660] INFO Saving data to scoutsuite-report\scoutsuite-results\scoutsuite_errors_aws-89074256517
0.json
2024-09-03 11:31:59 Junior-HP scout[20660] INFO Creating scoutsuite-report\aws-890742565170.html
2024-09-03 11:31:59 Junior-HP scout[20660] INFO Opening the HTML report
(venv) PS C:\Users\olwet\Downloads\ScoutSuite-master\ScoutSuite-master>
```

References

Chiarelli, A., 2020. *Auth0ByOkta.* [Online]
Available at: https://auth0.com/blog/preventing-clickjacking-attacks/
[Accessed 31 08 2024].

Forbes Expert Panel, 2024. *Council Post: 19 Keys To Detecting And Preventing Man-In-The-Middle Attacks.* [Online]
Available at: https://www.forbes.com/councils/forbestechcouncil/2024/03/07/19-keys-to-detecting-and-preventing-man-in-the-middle-attacks/
[Accessed 28 August 2024].

Gurvinder, K., 2014. Study of Cross-Site Scripting Attacks and their Countermeasures 0975 (8887). *International Journal of Computer Applications,* 170(9).

Indusface, 2024. *How to prevent SQL injection, 2 April 2024. [Online].* [Online]
Available at: https://www.indusface.com/blog/how-to-stop-sql-injection/
[Accessed 27 August 2024].

Jared, T., 2024. 19 Keys To Detecting And Preventing Man-In-The-Middle Attacks. *Forbes*, 12 August.

Kime, C., 2023. *How to Prevent DDoS Attacks: 5 Steps for DDoS Prevention. eSecurity Planet.* [Online]
Available at: https://www.esecurityplanet.com/networks/how-to-prevent-ddos-attacks/
[Accessed 28 August 2024].

Narayanan, S., 2017. Clickjacking vulnerability and countermeasures. *International Journal of Computer Applications 0975 (8887),* 170(9).

Nova, F., 2024. *ForeNova.* [Online]
Available at: https://www.forenova.com/blog/how-to-prevent-browser-session-hijacking-in-2024
[Accessed 31 08 2024].

Stripe, 2023. *Stripe.* [Online]
Available at: https://stripe.com/resources/more/secure-payment-systems-explained
[Accessed 31 08 2024].

TitanFile, 2023. *TitanFile.* [Online]
Available at: https://www.titanfile.com/blog/data-in-transit-encryption/
[Accessed 31 08 2024].