Form3 – Technical challenge.

Vulnerable service runs on port TCP 8000

URI Path is vulnerable to Boolean/Blind SQL injection /v1/report?customer?=””

<http://127.0.0.1:8000/v1/report?customer=>

Text

Description automatically generated

Figure 1 - Corroborates SQL injection the initial payload response returns “NULL” whilst the second CURL GET request with a valid “1111 = ‘11111 returns all the customer data.

Text

Description automatically generated

Figure 2 - SQLMAP was used to corroborate the backend SQLite database in use.

Text

Description automatically generated

Figure 3 – listing tables on the SQLite\_masterdb.

Text

Description automatically generated

Figure 4 extracting the user table username/base64 encoded password.

Armed with the manager credential it was possible to make a CURL POST request to obtain a valid token.

Text

Description automatically generated

Figure 5 obtained a valid token using manager credential.

Text

Description automatically generated

Figure 6 – armed with a valid token it was possible to obtain the secret flag.

Next step - automation with Python

**See attached exploit.py**

Requirements to run the exploit: 1) SQLMAP – Link to the SQLMAP repository: <https://github.com/sqlmapproject/sqlmap.git> 2) pip install requests.

How to run the exploit?

Python3 exploit.py 127.0.0.1:8000

Text

Description automatically generated

Figure 7 - Python exploit code was used to automate the attack obtained the flag.

Text

Description automatically generated

Figure – Secret flag obtained.