Mark Eilers

CS 405: Secure Coding

2-2 Activity: SQL Injection Coding

March 17, 2024

2-2 Activity: SQL Injection Coding Write Up

Unfortunately for my programing, Visual Studios was not allowing me to debug the program. The first screen shows that the debugging tool was inactive. I do not know why this is the case. For my screenshots, I included the code, which I believe to be correct. I will also discuss my process here. The process of writing the code was to have a clear understanding of the objective. For this assignment, I needed to create a C++ function capable of executing SQL queries on an SQLite database. It’s crucial to understand the significance of preventing SQL injection vulnerabilities, a common security concern when dealing with SQL queries.

The design of the run\_query function revolves around several key components. Firstly, it's necessary to define the function's parameters, including the SQLite database connection (sqlite3\* db), the SQL query to execute (const std::string& sql), and a vector to store the query results (std::vector<user\_record>& records). The function is also designed to return a boolean value indicating the success or failure of the query execution.

To address the risk of SQL injection, the function implements a method to identify specific patterns within the SQL query string. If the query matches a predefined pattern indicating a potential SQL injection attempt, the function takes precautionary measures to mitigate this risk. This involves preparing a modified SQL statement with parameter placeholders and binding parameters securely using the SQLite API. Throughout the code, error handling methods are implemented to handle potential errors that may occur during the execution of SQL queries. Error messages are displayed to provide feedback in case of query failures, aiding in debugging and troubleshooting.

Overall, the process of writing this code involves careful consideration of security concerns, effective use of the SQLite API, and diligent error handling to ensure the reliability and security of the database operations. By following these principles, the resulting run\_query function provides a safe and efficient means of executing SQL queries on a SQLite database in a C++ environment. For screenshot, please see the next page.

A computer screen shot of a black screen

Description automatically generatedA screenshot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated