Edwin Jones

CS-405: Secure Coding

May 17, 2025

Process Summary

In this activity, I modified the run\_query() function to detect SQL injection attempts using generalized logic-based patterns. Specifically, I checked for the presence of logical operators (OR, AND), unbalanced quotes (to catch incomplete string literals), SQL comment syntax (--, /\*, #), and semicolons (;) that could indicate statement chaining. These elements are commonly used in injection attacks such as OR 1=1; or 'a' OR 'a'='a';. Since SQL is case-insensitive, I converted the query to lowercase before performing these checks to ensure consistent detection regardless of capitalization.

If an injection is suspected, the program prints a message to the console and blocks execution of the query. Otherwise, the query is allowed to run as intended.

I tested valid queries (e.g., WHERE NAME='Fred') and confirmed that they execute normally, while injected queries (such as those generated by run\_query\_injection) are correctly detected and rejected. This approach avoids false positives and ensures that only malicious patterns are intercepted. The final implementation prevents SQL injection reliably while preserving normal functionality.