Task 2 Report

We measured the likelihood of a query being malicious by scanning it for potentially malicious keywords and characters. Our program errs on the side of being overly cautious, as we deemed the potential cost of a malicious query being falsely accepted to be greater than the potential cost of a benign query being falsely labeled as malicious.

The scanner checks for tautologies by checking for "=", "<”, “>”, “and”, and “or”. It checks for union attacks for checking for “union”. Then it checks for piggyback attacks by checking for ";". Then it checks for timing attacks by checking for "waitfor". After that if checks for alternate encodings by checking for the presence of “char(“ and “exec(”, and finally checks for illegal/logically incorrect queries by checking for the presence of " (unclosed quotations) and “convert(“. For all of these checks, the scanner only checks for the presence of these characters/substrings in the query in locations where it wouldn't naturally be in query 1 and 2. For instance, query 1 already contains one instance of “AND", so this instance will not trigger the security scanner. However, any other instances of “AND” will trigger the security scanner. For each instance of a suspicious character or substring, the “score" of the query is incremented by 1. The score is a measure of the likelihood of a query being malicious and is capped at 5, so the score is a measure from 0 to 5 of the likelihood of a malicious query.

The scope of improving the SQLI security scanner could include having a security professional attempt to write malicious SQL queries which could get past the security scanner. By identifying such queries, we could modify our scanner in such a way as to make it able to detect the malicious queries which would initially get past it. By doing this, our scanner would be improved and would become more effective at detecting malicious queries.