**CPSC 4660 – Project Proposal**

**SQL Injection Attack Prevention**

**By: Amanda Munden**

**#001148507**

**Due: October 4, 2019**

**Background:**

SQL injection attacks are some of the most common attacks on databases. There are multiple SQL injection strategies that are used to accomplish different goals when attacking a database. Some of the most common types of attacks are using tautologies which can allow you to bypass authentication and retrieve data you would not normally have access to, using a Union query which allows you to return additional data from a query, and using Piggy-Backed queries which allows you to alter the database by removing tables or tuple, changing values or adding values. To test my different preventative techniques that I will be implementing in this project, I will be using attacks from all three strategies to see how accurately each of the prevention methods stops the different kinds of attacks.

**Summary of Component Being Implemented:**

For this project, I am going to be implementing three separate preventative techniques that can be used to reduce and try to eliminate SQL injection attacks. Two of the techniques are quite basic. The first one is “sensitive key word filtering” which is where the program will check to see if there are sensitive key words in the submitted data and if there are then the program execution will stop and return an error message. The second one is “type or length detection” which is where the program will check to make sure that submitted data is of the correct data type and if there is a fixed length for the data, that the input is of the correct length. Finally, I will implement “tokenization” which is done by using a “query parser method” to break down the expected input into tokens and then the actual input into tokens and compare the two sets of tokens. The goal will be to determine if a more complex strategy like “tokenization” better prevents SQL injection attacks than more basic strategies like “sensitive key word filtering” or “type length detection”.

**Evaluation Strategy:**

My evaluation strategy is to apply one of the prevention strategies to my test database and then try multiple kinds of attacks against the database and see how many of the attacks the strategy can prevent and how accurately it prevents the attacks. I will repeat this for all three prevention strategies to see which one can prevent the most attacks and if there is a difference in how well a simple strategy prevents attacks compared to a more advanced and complex strategy.

**Plan to Produce Test Data for Evaluation:**

To be able to implement my strategies and test their accuracy, I will be creating a test mySQL database with dummy data.

**References**

Yeole, A. S., Meshram, B. B. (2011). Analysis of different technique for detection of SQL injection. *ICWET '11: Proceedings of the International Conference & Workshop on Emerging Trends in Technology*. Publisher ACM.

Zhang, Haiyan, Zhang, Xiao. (2018). SQL Injection Attack Principles and Preventive Techniques for PHP Site. *CSAE '18: Proceedings of the 2nd International Conference on Computer Science and Application Engineering*. Publisher ACM.