



Tribhuvan University
Institute of Science and Technology

**Blocking SQL Injection in Database Stored
Procedures**

Dissertation

Submitted to

Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements
for the Master's Degree in Computer Science and Information
Technology

by

Sanu Manandhar

(December, 2010)

Supervisor

Dr. Subarna Shakya

(Associate Professor)

Department of Electronics and
Computer Engineering,
Pulchowk Campus



Tribhuvan University
Institute of Science and Technology
Central Department of Computer Science and Information
Technology

Date: _____

Recommendation

I hereby recommend that the dissertation prepared under my supervision by **Mr. Sanu Manandhar** entitled “**Blocking SQL Injection in Database Stored Procedures**” be accepted as fulfilling in part requirements for the degree of masters of science. In my best knowledge this is an original work in computer science.

Dr. Subarna Shakya

Associate Professor

Department of Electronics and Computer Engineering,

Pulchowk Campus,

Institute of Engineering

(Supervisor)



Tribhuvan University
Institute of Science and Technology
Central Department of Computer Science and Information
Technology

We certify that we have read this dissertation work and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Master of Science in Computer Science and Information Technology.

Evaluation Committee

Prof. Dr. Jeevan Jyoti Nakarmi
Act. Head, Central Department of Computer
Science and Information Technology
Tribhuvan University

Dr. Subarna Shakya
Associate Professor
Department of Electronics and Computer
Engineering,
Pulchowk Campus, IOE
(Supervisor)

(External Examiner)

(Internal Examiner)

Date: _____

ABSTRACT

Web application is described as an application accessible by the web through a network. SQL injection is an attack method used by hackers to retrieve, manipulate, fabricate or delete information in organizations' relational databases through web applications. Information processed by web applications has become critical to corporations, customers, organizations, and countries.

Several research papers in literature have proposed ways to prevent SQL injection attacks in the application layer by examining dynamic SQL query semantics at runtime. However, very little emphasis is laid on securing stored procedures which could also suffer from SQL injection attacks. Some research papers in literature even refer to stored procedures as a remedy against SQL injection attacks. As stored procedures reside on the database front, the methods proposed by them cannot be applied to secure stored procedures themselves.

In this research paper, we propose a technique to defend against the attacks targeted at stored procedures.

ACKNOWLEDGEMENT

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

I am heartily thankful to my supervisor, **Associate Professor Dr. Subarna Shakya** , whose encouragement, guidance and support from the initial to the final level enabled me to develop and understanding of the subject. I want to thank my co-supervisor **Mr. Jagdish Bhatta** for giving me continuous support, inspiration and guidance throughout the study period of my thesis work. I am grateful to Dr. Shashidhar Ram Joshi, Mr. Min Bahadur Khati of Central Department of Computer Science and Information Technology who, while not being directly involved in my thesis work, nevertheless influenced me greatly.

Many thanks go to my friends Mr. Madhav Dhakal, Mr. Sachin Kumar Shrestha and Mr. Rajesh Gurubacharya for their interest, cooperation, worries and complain.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

CONTENTS

ABSTRACT.....	IV
ACKNOWLEDGEMENT.....	V
ABBREVIATONS.....	VI
CONTENTS.....	VII
LIST OF FIGURES.....	IX
LIST OF TABLES.....	X
1. Introduction.....	1-3
1.1 Background.....	1
1.2 Problem.....	1
1.3 Objective and Outline.....	2
1.4 Literature Review.....	3
2. Web Application.....	4-9
2.1 Introduction.....	4
2.2 Architecture of Web application.....	5
2.3 Input Validation-Based Vulnerabilities.....	5
2.4 Web application and SQL injection.....	8
3. SQL.....	10-11
3.1 Introduction.....	10
3.1.1 DML.....	10
3.1.2 DDL.....	11
3.2 Query Techniques.....	11
3.2.1 Dynamic SQL.....	11
3.2.2 Static SQL.....	11
4. SQL Injection.....	12-28
4.1 SQL Injection Attacks.....	12
4.2 Attack Intention.....	13
4.3 How It Happens.....	15
4.3.1 Dynamic String Building.....	16
4.3.2 Insecure Database Configuration.....	23
4.4 Existing Technologies to stop SQL Injection.....	26
4.4.1 Defensive Programming.....	26

4.4.2	Anomaly Detection.....	28
5.	Existing Prevention/Detection Models.....	29-38
5.1	Secure SQL Processing.....	29
5.1.1	Proposed architecture.....	32
5.1.2	Performance.....	32
5.1.3	Shortcomings of the technique.....	33
5.2	Weight-based Symptom Correlation Approach.....	33
5.2.1	Detection Approach.....	37
5.2.2	Performance.....	38
6.	Stored Procedures.....	39-54
6.1	Introduction.....	39
6.2	Why Stored Procedures?.....	40
6.3	SQL Injection in stored procedures.....	41
6.4	Types of Attacks.....	45
6.5	Preventing SQLIA's Stored Procedures.....	50
6.5.1	Validating Input.....	50
6.5.2	Encoding Output.....	53
7.	Analysis Implementation and Results.....	55-60
7.1	Implementation	55
7.2	Results.....	56
7.3	Performance Analysis	58
8.	Conclusion and Future work.....	61
8.1	Concluding remarks.....	61
8.2	Future work.....	62
	References.....	63

LIST OF FIGURES

FIGURE	TITLE	PAGE NO.
Figure 1	Architecture of a typical web based system	5
Figure 2	Vulnerability Type	6
Figure 3	Reported Web Attacks in 2008	7
Figure 4	Number of incident by number of records leaking in 2007	8
Figure 5	Informative error messages	27
Figure 6	Node structure of the main doubly linked list	30
Figure 7	Node structure of singly linked list for storing a valid individual query structure	30
Figure 8	Correlation Process for weight-based approach	34
Figure 9	User Login Form	42
Figure 10	Flow chart for intrusion detection for SQL-injection	56
Figure 11	SQL-Injection intrusion detection	57
Figure 12	Number of input versus execution time	59
Figure 12	Number of users versus execution time	60

LIST OF TABLES

TABLE	TITLE	PAGE NO.
Table 1	Data Returned After Successful Injection	21
Table 2	Summary of testing results	47
Table 3	Execution time for different no. of input	48
Table 4	Execution time for different no. of users	50

ABBREVIATIONS

SQL	Structured Query Language
SQLIA's	SQL-Injection Attacks
SP	Stored Procedure
API	Application Program Interfaces
RDBMS	Relational Database Management System
DML	Data Manipulation Language
DDL	Data Definition Language
HTML	HyperText Markup Language
ID	Intrusion Detection