Universiti Teknologi MARA

SQL INJECTION: COMPARISON OF PREVENTION STRATEGIES FOR PHP

MOHD FAIRUZ BIN ABDUL JALIL
2005616484

Thesis submitted in fulfillment of the requirements for
Bachelor of Science (Hons) Information System Engineering
Faculty of Information Technology And
Quantitative Science

MAY 2008

ACKNOWLEDGEMENT

In the name of Allah, the Most Merciful and the Most Compassionate.

Alhamdulilah and peace upon Prophet Muhammad S.A.W., this thesis is finally finished according to time and objectives required.

This study would not have been possible without the assistance and support from those who guided the author in his course of graduate work. First, the author would like to thank Allah SWT for His grace and mercy throughout this research. It is by His hands and wisdom in guiding the author to finish his work within the study.

The author would like to extend his thanks to honorable supervisor, Cik Ruhaila Bte Maskat, for academic guidance, support, encouragement, and help during the study. Furthermore, the author would like to specially thank her for patience and tolerance, in which her diligence, dedication and working attitudes are good examples to the author.

And the most thanks to my family for their encouragement in finishing this course gloriously. Last but not least, for my friends who have given advices and suggestions to make this thesis achieves it goals.

Thanks.

TABLE OF CONTENTS

CONTENT *PAGE*

ABSTRACT

Since 2002, over 10% of total cyber vulnerabilities were SQL injection vulnerabilities. Since most developers are not experienced software security practitioners, a solution for correctly fixing SQL injection vulnerabilities that does not require security expertise is desirable. By using SQL injection attacks, an attacker could thus obtain and/or modify confidential/sensitive information. SQL injection attacks take advantage of code that does not filter input that is being entered directly into a form. Susceptible applications are applications that take direct user input and then generate dynamic SQL that is executed via back-end code. Objectives of the research are to indentify weakness in current website, identify the prevention strategies, applying malicious code to PHP framework which is Joomla 1.0.15 and eZ Publish 4.0.0 and finally identify whether SQL prevention strategies have been applied. As for the results, both framework have applied the SQL injections prevention strategies and not allowing SQL injection to occur.