LOGICAL OPERATORS AND ITS APPLICATION IN DETERMINING VULNERABLE WEBSITES CAUSED BY SQL INJECTION AMONG UTM FACULTY WEBSITES

NURUL FARIHA BINTI MOKHTER

UNIVERSITI TEKNOLOGI MALAYSIA

LOGICAL OPERATORS AND ITS APPLICATION IN DETERMINING VULNERABLE WEBSITES CAUSED BY SQL INJECTION AMONG UTM FACULTY WEBSITES

NURUL FARIHA BINTI MOKHTER

A disertation submitted in partial fulfillment of the
requirements for the award of the
Master of Science Mathematics

Faculty of Science
Universiti Teknologi Malaysia

JANUARY 2013

To the endless list of people who are meant very much to me

Mokhter bin Abdullah

Sobariah binti Sulong

Muhammad Farhan

Muhammad Fakhrul Aiman

Muhammad Fathul Muin

Muhammad Fariq Irfan

Thank you very much for the endless support

The one who inspire me and give the strength

My beloved fiancé,

Muhamad Nur ‘Azim bin Muhaimin

Lecturers and friends

Thank you very much

"It is not how much you do, but how much love you put in the doing."

ACKNOWLEDGEMENT

In The Name of Allah, The Most Gracious And The most Merciful first and foremost, I would like to extend my heartfelt gratitude to my supervisor, Prof Madya Dr. Jamalludin Bin Talib, who helped me a lot throughout the duration of the research. His efforts in guiding, supporting and giving constructive suggestions are much appreciated. My sincere appreciation also extends to all my course mate and other who have provided assistance at various occasions. Their views and opinions are helpful indeed. Unfortunately, it is impossible to list all of them in this limited space. Moreover, I would like to thank to my family members who have given their undying support. Last but not least, I would like to take this opportunity to express my heartiest appreciation to all those involved in helping me to complete this project. Thank you for their tolerance in doing all this things.

ABSTRACT

This research identifies the problems caused by SQL Injection Bypassing Login among Universiti Teknologi Malaysia (UTM), Johor Bahru faculty's websites by applying Logical Operators. Structured Query Language, SQL is some kind of language used to allow users to work on the data stored in a database. SQL Injection is a technique to gain illegal access through the vulnerability of a website. Vulnerability of a website means poorly designed website in performing operation on the database. In order to test the vulnerability of websites, the SQL Injection rule will be created and will be test on Mutillidae website to see the effectiveness of the rule. If the rule is effective enough, those rule will be combined by using logical operators. The combination rule will be tested on UTM faculty's websites to observe and determine whether the website is vulnerable to the combination rule of SQL Injection or not. If the website can be accessed using this combination rule, conclusion can be made that the website is vulnerable and needs to improve on their website security. Hence, the way to avoid SQL Injection will be recommended.

ABSTRAK

Kajian ini dijalankan untuk mengenal pasti laman web yang lemah disebabkan oleh Bahasa Pertanyaan Berstruktur, SQL antara laman web fakulti di Universiti Teknologi Malaysia (UTM), Johor Bahru dengan mengaplikasikan Operator logik. Bahasa Pertanyaan Berstruktur, SQL adalah antara bahasa yang digunakan untuk membenarkan pengguna untuk berurusan dengan data yang disimpan dalam pangkalan data. Suntikan SQL adalah satu teknik untuk mendapatkan akses haram melalui kelemahan laman web. Dalam usaha untuk menguji kelemahan laman web, peraturan Suntikan SQL akan diwujudkan dan akan diuji ke atas laman web Mutillidae untuk melihat keberkesanan kaedah. Jika peraturan cukup berkesan, peraturan akan digabungkan dengan menggunakan operator logikal. Kombinasi peraturan akan diuji di laman web fakulti UTM untuk melihat dan menentukan sama ada laman web ini adalah terdedah kepada kombinasi peraturan Suntikan SQL atau tidak. Jika laman web tersebut boleh diakses menggunakan kombinasi peraturan ini, kesimpulan yang boleh dibuat ialah laman web ini adalah lemah dan perlu mempertingkatkan keselamatan laman web mereka. Oleh itu, cara untuk mengelakkan Suntikan SQL akan disyorkan.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Introduction

In recent years, internet has become an important part in our daily life as well as facilitating the users in many ways. The rapid growth of the internet offers a variety of web applications which can be accessed using any web browser that runs on any architecture and operating system. Furthermore, internet has become one of the platforms to spread any information as well as exchange any story or news and, sometimes, confidential information.

Users also utilize the internet as a platform in doing online business, encouraging people to shop online, make reservations and also pay bills in which all of these activities are related to money-transfer. A lot of web application users are not aware or do not have any education on their security and privacy on using this kind of service. Lack of education and knowledge on security and privacy of users can expose them to a variety of security threats presented in those websites.

Basically, web browser is the most insecure platform channel in exchanging any information detail because of the rapid growth of web applications which are not secure enough. Users are vulnerable to lots of web security threats which attack sometimes without their knowledge. One of the gravest security threats which is becoming 'notorious' at present is the Structured Query Language (SQL) Injection. SQL Injection is a technique to gain illegal access through the vulnerability of the web application. In other words, the malicious user, or nowadays called Hacker, will inject a dynamically constructed SQL query which contains the input data, and hence the malicious query will be treated as a SQL code (William J. V., Halfond G. J., and Orso A., 2006).

By injecting the malicious SQL Injection or code injection, the web application could be affected as it is very threatening for all components of the web application. By inserting SQL Injection to the web application, it will result in illegal access to the database, modification and extraction of the database content as well as dissemination of confidential information. As of recent, SQL Injection Attacks (SQLIAs) are becoming one of the most serious threats to web applications (Partington V. *et al*, 2005; Ulmer C. *et al*, 2010; Anchila A. *et al*, 2010)

1.2 Background of study

Database Management System (DBMS) is a program or a software system of the database which operates in managing data transfer, organizing the data as well as providing ways for users or other programs to modify and extract information in the database (Healey R. G., 2007). As the DBMS was build with its own modeling language, there are five types of modeling languages used in DBMS; one of them being the SQL which can examine and process the data in the database structure.

The modeling language is the structure of the database used in DBMS and the relational database structure is the type of model structure used by the SQL which is the one of the language designed to manage all the data in the database structure. Edgar F. Codd's study in his paper "A Relational Model of Data for Large Shared Data Banks" in 1970 described that for his relational model, the Relational Database Management System (RDBMS), the SQL was one of the first commercial language (Codd E. F., 1970).

In order to access and work with the data stored in RDBMS, one way is to develop or create a language to allow users to get into a database. The Structured Query Language (SQL) allows its users to work on the data stored in a database on the computer in a relational database system (Chappel D., and Trimble J. H. Jr., 2002). It is called query when users are trying to access a database and the language users use in order to access to the database is called the query language. When the SQL query are manipulated, and hence inserted into the web application, it will perform an unauthorized database operation. As this illegal operations works, the malicious user would illegally access the database and affect the security of the information contained in it as the website totally depends on database.

More than 50,000 websites, including the US Department of Homeland Security, the United Nations and also the UK government websites had been attacked and hacked in 2008 (Khaleel Ahmad, Shekhar J., Yadav K. P., 2011). Ironically, on March 27 2011, the MySQL website and the Sun/Oracle website had also been hacked by one of the SQLIAs, Blind SQL Injection. As the website stores and manages lots of confidential and important information and data, the web application should be well protected from malicious users who try to gain illegal access on the remote machines through the web application's vulnerability.

1.3 Rational of the Study

Vulnerability of a website in the computering context means poorly designed website in performing operation on the database, or in other words, the weaknesses of the website. Most websites nowadays are exposed to vulnerability due to the lack of proper validation or conformation when the users enter the data to the web application and also the lack of understanding of this kind of attack when the programmer is building the website. In 2005, the Gartner Group conducted a study on more than 300 websites that are exposed to the vulnerability of SQL Injection and, unexpectedly, 97 percent of the tested websites were vulnerable to SQL Injection attacks (Buehrer G. T., Weide B. W., and Sivilotti P. A. G., 2005).

As websites are vulnerable to security threats such as SQL Injection attacks, they are exposed to modification of the content by the malicious users in the database including insertion, updating and deleting, shutting down the Database Management System (DBMS) and sometimes, modification of the command of the operating system (Anchila A. *et al*, 2010; Skaruz J. *et al*, 2010; Ruzhi X. *et al*, 2010; Yaashuwanth C. *et al*, 2010; Bisht P. *et al*, 2010; Sushila M. *et al*, 2010; Hossain Shahriar *et al*, 2010).

The rational of this study is to obtain the malicious SQL Injection rule which is called bypass strings that can be used to test vulnerability of websites. The rule will be test on a vulnerable website to see the effectiveness of the rule. Then, those rules will be combined by using Boolean Logic and Boolean Operators. The combination rule will be tested on real world websites to observe and determine whether the website is vulnerable to the combination rules of SQL Injection or not. If the website can be access using this combination rules, conclusion can be made that the website is vulnerable and needs to improve on their website security. Hence, the way to avoid SQL Injection and the problems of the vulnerability of a website caused by SQL Injection will be recommended.

1.4 Statement of Problem

Mass media has reported a lot of issues on websites being hacked by the irresponsible users. These hacking activities can affect the website if the website is not well protected and has vulnerability. The rapid growth of the Structured Query Language Injection Attacks (SQLIAs) nowadays concerns all web application users. There have been a lot of complaints and reports due to these problems because by inserting the SQL code or the malicious query of SQL, the attackers can have legitimate access to the database, worrying other users about their security and privacy which lie under the web application.

As reported on February 2009, the percentage of websites that have been hacked by special crafted SQL query, the SQL Injection attacks, is more than 30% (Breach Security Inc. Annual Report, Feb. 2009). Therefore, it is hope that the issues raised from this research can provide beneficial inside especially to the website developer so that they will concern to apply more security checks in their websites.

This project tries to answer some questions; the first is on how to know that websites are vulnerable to SQL Injection Bypassing Login or not? Secondly, is there any UTM faculty websites which are vulnerable to SQL Injection Bypassing Login? Last but not least, how are the vulnerable websites can be protected from SQL Injection attack?

1.5 Objective of the Study

The objectives of the research are to:

a) Explore and learn Boolean Logic and Boolean Operator in application on SQL Injection.
b) Create and combine SQL Injection rules or bypass strings that can be used to test websites by using SQL Injection Bypassing Login.
c) Determine the vulnerable websites caused by SQL Injection Bypassing Login among Universiti Teknologi Malaysia (UTM) faculty websites.
d) Recommend solutions to the problems of vulnerability of websites cause by SQL Injection.

1.6 Significance of the Study

The significance of this study is to upgrade level of protection among vulnerable website cause by SQL Injection among Universiti Teknologi Malaysia (UTM) faculty websites. The vulnerability of a website happens due to less protection of the website. Usually, most of the website developer are too lazy in considering and applies proper security checks in their website. This thought could expose their website to the malicious users. The website could be attacked by SQL injection. By creating the combination of SQL Injection rule and test it on the website, we can observe and determine whether the website is vulnerable to SQL Injection or not. If the website can be access by using those combination rules, the websites are vulnerable to the SQL Injection. Hence, this will facilitate us to determine solutions to the problems so that the website can upgrade their protection and security.

1.7 Scope of the Study

In this research, we are mainly concerned with the Boolean Logic and Boolean Operator in the application of SQL Injection. The SQL Injection rule or in other name bypass string will be create by using the application of Boolean logic and Boolean operator in SQL Injection. The rules will be tested on the vulnerable test website, Mutillidae to observe the effectiveness of the rules. Those effectives' rules will be combined by Logical Operator in order to get the combination rule that will be used to test websites. Then, the combination rule will be injected into the real world website which will be selected among Universiti Teknologi Malaysia (UTM) faculty websites. Recommendations on ways to avoid SQL Injection will be suggested.

1.8 Report Outline

This report begins with Chapter 1 which is the introduction, rational of the study, problem statement, objectives, significance and scope of study. Chapter 2 discusses some literature review on the basic concept of Database Management System (DBMS), Structured Query Language (SQL), SQL Injection and Boolean Algebra concentrated on the Boolean Operator and Boolean Logic. The research methodology, operational framework and research planning are discussed in Chapter 3, while analysis and discussions will be explained in Chapter 4 and Chapter 5 will include the result and recommendations.

REFERENCES

Anchila A. and Jain S. (2010). A Novel Injection Aware Approach for the Testing of Database Applications. *IEEE International Conference on Recent Trends in Information, Telecommunication and Computing*. IEEE Computer Society Washington, DC, USA. 311- 312.

Bisht P., Sistla A. P., and Venkatakrishnan V. N. (2010). Automatically preparing Safe SQL Queries. *14th Financial Cryptography and Data Security Conference (FC'2010)*. 25 – 28 January. Canary Islands, Spain. 272-288.

Blum J. (2010). *Using AND, OR and NOT (Boolean Operators)*. Mathewson-IGT Knowledge Center.

Brian H. (2011). Third Base. *American Scientist (SigmaXi, the Scientific Research Society)*. 490-494.

Buehrer G. T., Weide B. W., and Sivilotti P. A. G., (2005). Using Parse Tree Validation to Prevent SQL Injection Attacks. *SEM '05 Proceedings of The 5th International Workshop On Software Engineering And Middleware*. New York, USA. Sept 2005. 106-113.

Chamberlin D. D., and Boyce R. F., (1974). SEQUEL : A Structured English Query Language. *Proceedings of the 1974 ACM SIGFIDENT Workshop on Data Description, Access and Control (Association for Computing Machinery).* September 6, 2007. 64-249.

Chappel D., and Trimble J. H. Jr., (2002). *A Visual Introduction to SQL*. (2nd ed.) New York. John Wiley & Sons, Inc.

Christiansen S., and Route M. (Ed.), (2005). Database Management System (DBMS). Retrieved September, 2005. from Tech Target Website : http://searchsqlserver.techtarget.com/definition/database-management-system

Codd E. F., (1970). A Relational Model of Data for Large Shared Data Banks. Magazine Communication of the ACM – Special 25th Anniversary Issues. Vol. 26 (Issue 1). 64-69.

Dolson J. (2004). Working With Boolean Queries in MySQL/PHP. *Practical eCommerce Magazine*.

Elmasri R., and Navathe S. B., (1994). Chapter 11 : The Hierarchical Data Model and the IMS Syatems. *Fundamentals of Database Systems*. Benjamin / Cummings. (2nd ed.) 343-389.

Elmasri R., and Navathe S. B., (1994). Chapter 8 : SQL - A Relational Database Language. *Fundamentals of Database Systems*. Benjamin / Cummings. (2nd ed.) 343-389.

Elmasri R., and Navathe S. B., (1994). Chapter 9 : A Relational Database Management System - DB2. *Fundamentals of Database Systems*. Benjamin / Cummings. (2nd ed.) 343-389.

Healey R. G., (2007). *Database Management Systems*. In Goodchild M. F., Rhind D. W., and Maguire D. J. (Ed.). *Geographical Information Systems : overview, principles and applications*. (pp. 251-267). Londan : Longman.

Heberling G. (2008). *Lesson 5 : Types of Database Management System*. Retrieved September 1, 2009. from Penn State New Kensington University Web site : http://www.personal.psu.edu/glh10/ist110/topic/topic07/topic07_06.html

Hossain Shahriar and Mohammad Zulkernine. (2010). Taxonomy and Classification of Automatic Monitoring of Program Security Vulnerability Exploitations. *The Journal of Systems and Software*. Vol. 84 (Issue 2). 250-269.

Irving C. (1918). A Survey of Symbolic Logic. *Berkeley : University of California Press*. 157.

Khaleel Ahmad, Shekhar J., Yadav K. P., (2011). Coalesce Techniques to Secure Web Applications and Databases against SQL Injection Attacks. Electronic Journal of Computer Science and Information Technology (eJCSIT). Vol. 3 (No. 1). 26-30.

Knuth, Donal E., (1981). The Art of Computer Programming. *Reading Mass : Addison-Wesley Publishing Company*. Vol. 2. 190.

Lemon S. (2008, May 19). Mass SQL Injection Attack Targets Chinese Web Sites. PCWorld Communications, Inc. *IDG News*.

Los R. (2011). *MySQL Website Hacked (Ironically) by Blind SQL Injection*. Infosec Island, LLC. March, 29.

Oppel A. (2004). Database Demystified. *CA : McGraw Hill Osborne Media*. February 2007. 1-90.

Partington V. and Xebia E. K. (2005). Top Ten Most Critical Web Application Vulnerabilities. *Open Web Application Security (OWASP) Foundation*.

Patel N., Fahim Mohammed, and Soni S. (2011). SQL Injection Attacks : Techniques and Protection Mechanism. *International Journal on Computer Science and Engineering (IJCSE)*. Vol. 3. 199-203.

Ruzhi X., Jian G., and Liwu D. (2010). A Database Security Gateway to the Detection of SQL Attacks. *IEEE 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. 20-22 August. Univ. Beijing, China. 537-540.

Shah H. (2006). *Understanding SQL Injection*. Infys Systems.

Silveria C., Eloy L., and Monteiro J. M., (2010). A Query Language for Data Access in Ubiquitous Environments. Clei Electronic Journal, Vol. 13. no. 3, paper 2.

Skaruz J., Nowacki J. P., and Drabik A. (2010). Soft Computing Techniques for Intrusion Detection of SQL-Based Attacks. *ACIIDS'10 Proceedings of the Second international conference on Intelligent information and database systems: Part I*. Springer-Verlag Berlin Heidelberg. 33-42.

Sushila M. and Supriya M. (2010). Security Standards Perspective to Fortify Web Database Applications From Code Injection Attack. *IEEE International Conference on Intelligent Systems, Modeling and Simulation*. 27-29 January. Univ. of Delhi, India. 226-230.

The Penguin Dictionary of Mathematics. (1998). *London, England : Penguin Books*. (2nd ed.). 417.

The Web Hacking Incidents Database 2008. *Breach Security Inc. Annual Repo*rt. February 2009.

Ulmer C., Gokhale M., and Gallagher B. (2010). Massively Parallel Acceleration of a Document-Similarity Classifier to Detect Web Attacks. *Journal of Parallel and Distributed Computing,* 225-235.

William J. V., Halfond G. J., and Orso A., (2006). A Classification of SQL Injection Attacks and Countermeasures. *Proceedings of the IEEE International Symposium on Secure Software Engineering*. March 2006. Arlington, VA, USA.

Yaashuwanth C. and Ramesh R. (2010). Attacks in WEB Based Embedded Applications. *International Journal of Computer Sciences Issues (IJCSI)*. Vol. 7 (Issue 6). 116-119.