A Review on SQL Injection Attacks & Prevention System Technology

Ankit Srivastava (212IS003)

Department of Computer Science and Engineering
National Institute of Technology Karnataka
Surathkal, Mangaluru, India

Abstract. SQL injection attacks constitute a serious security risk to Web applications because they offer attackers free access to the databases supporting the apps and the highly sensitive information stored in such databases. Despite the fact that researchers and practitioners have presented many techniques to address the SQL injection problem, existing approaches either fail to address the entire breadth of the problem or have constraints that preclude their use and acceptance. In this article, we will look at PHP strategies and other ways for preventing SQL from injection, methods for detecting SQL assaults, forms of SQL injection, causes of SQL injection via receiving and posting, and SQL vulnerability protection technologies.

Keywords: SQL Injection, PHP, Database Security

1 Introduction

In today's technology-driven society, website apps play an important role in daily living. Websites are used for a variety of activities, including online shopping, banking, and talking with friends. Databases are frequently used on the backend of websites to store user data. Because such files include sensitive information such as passwords, credit card numbers, and social security numbers, hostile hackers frequently target them. SQL injection flaws have been identified as one of the most severe risks to Web applications. [1] SQL injection vulnerabilities in web applications may allow an attacker to obtain total access to its supporting databases. Since these databases usually contain sensitive client or user information, the associated security violations can include identity theft, confidential information loss, and fraud. In rare situations, a SQL injection vulnerability can be used by attackers to seize power of and harm the system that serves the Web application.

1.1 Problem Description

SQL injections lead to the theft of critical information. [2] Furthermore, it has the potential to be disastrous for any business, government, or institution. Such occurrences can jeopardise the company's operations and image, as well as result

in large fines imposed by data protection regulations. In order to prevent such kind of highly devastating attack, we have discussed some crucial prevention and detection methodology in the upcoming section.

1.2 Motivation

SQL injection is a form of insertion attack that may be defined as a harmful tactic that attacks the website's SQL-based application software by injecting malicious SQL statements or exploiting faulty input. It does not need high-level implementation, resulting in perhaps one of the most deadly assaults. SQL injection attacks are constantly at the top of the attackers' priority list. SQL injection attacks accounted for nearly two-thirds (65.1 percent) of all attacks against software applications between 2017 and 2019. Looking at the above records, the prevention of such deadly activity should be at the top priority for us, thus we have tried to provide some way out to get rid of it.

1.3 Scope

SQL Injection is a methodology for trying to persuade an application to execute Sql statements which was not anticipated. The attacker's goal is to get around the security screening and get illegal access using the results of the modified SQL query. This type of attacks are devastating the majority of online applications. Such dangers can be avoided with cautious planning and execution. Now that we are aware of such flaws, we are implementing the appropriate validation tests to avoid any form of SQL injection. Once we will be able to prevent such attacks, most of the web applications would be less prone to attack, atleast from their database respective. There can be more such methods to prevent such attacks, so we can dive deep into this to find the best suited and strong protection wall against all such injection attacks.

1.4 Objectives

This proposal's goal is rather obvious. To begin, we discovered a common cause and vulnerability in the system, such as insufficient validation of user input. To address this issue, we suggested a set of coding principles that encourage defensive coding approaches such as encoding user input and validation. A thorough and systematic implementation of these strategies is an excellent solution for preventing SQL injection issues. However, in practise, the execution of such strategies is human-based and hence prone to mistakes. Furthermore, repairing older code bases that may include SQL injection vulnerabilities may be a time-consuming and labor-intensive operation.

1.5 Organization of the Report

The remainder of this work is structured as follows: Section 2 provides a detailed literature review of prior work in this topic. Section 3 identifies and provides the

most significant aspect of this work, which includes several ways for dealing with SQL injection attacks. Section 4 outlines all of the experimental results, which are then followed by the Conclusion and Reference.

2 Related Work

According to an assessment of countless hacking incidents, as operating security system enhances and security protection software solutions become more widely available, security breaches directly triggered by operating system vulnerabilities reduce every year, while WEB application system vulnerabilities increase in usage. PHP has become the dominant language for constructing all types of portal websites and Web application programs due to its simplicity and excellent development performance.

In 2009, 15th IEEE Pacific Rim International Symposium on Dependable Computing paper, Nuno Antunes and Marco Vieira from University of Coimbra proposed a comparison between two major techniques Penetration Testing and Static Code Analysis for SQL Injection Detection. [1] In order to understand the strengths and limitations of these tactics, they used a variety of commercial and open source tools to uncover vulnerabilities in a collection of susceptible services. Static code analyzers, according to the research, reveal more SQL Injection vulnerabilities than penetration testing methodologies. Another significant discovery is that tools that utilise the same detection method frequently detect distinct vulnerabilities. Finally, many tools have insufficient coverage and have a large rate of false positives, making them inappropriate for programmers.

At IEEE international symposium on secure software engineering,2006, William G.J. Halfond, Jeremy Viegas, and Alessandro Orso from College of Computing, Georgia Institute of Technology proposed that an in-depth analysis of the many forms of SQL injection attacks known to date. [2] Their paper contain descriptions and illustrations of how each form of attack may be carried out. They also provided an assessment of existing SQL injection detection and prevention methodologies, as well as the benefits and drawbacks of each strategy in dealing with a wide spectrum of SQL injection threats.

At International Journal on Computer Science and Engineering (IJCSE), Nikita Patel, Fahim Mohammed and Santosh Soni proposed a paper titled "SQL Injection Attacks: Techniques and Protection Mechanisms". [3] This document presents challenges relating to information leaking via SQL injection attacks, as well as protection methods.

During 3rd International Conference on Computer Science and Information Technology in 2010, Lambert Ntagwabira, from Central South University (CHINA) P.O.BOX 410083, Department of Information Science and Engineering, China and Song Lin Kang from Central South University (CHINA) P.O.BOX 410083, Department of Information Science and Engineering, China proposed a paper titled "Use of Query tokenization to detect and prevent SQL injection attacks". [4] Their study objective was to develop a mechanism for identifying and preventing SQL injection attacks by analysing whether user in-

puts impact the intended output of the query. They demonstrated a technique for detecting SQL injection attacks that makes use of query tokenization through the QueryParser function. In most cases, when performing SQL injection, the attacker should use a space, single quotes, or double dashes in his input. Tokenization is accomplished by detecting a space, single quotation mark, or double dash, and all strings before each symbol are combined to form a token. Following the creation of tokens, they are all combined to form an array, with each token being a member of the array.

In 2nd International Conference on Computer Science and Application Engineering, October 2018 Article No.: 187, Haiyan Zhang, Agricultural informatics, Hebei North University, Zhangjiakou, China and Xiao Zhang, Medical informatics, Hebei North University, Zhangjiakou, China proposed a paper titled "SQL Injection Attack Principles and Preventive Techniques for PHP Site". [5] Their study examines the motivations for SQL injection and does comprehensive research on common SQL injection attack techniques, using PHP as an example. This article presents SQL injection detection tools and how to avoid SQL injection vulnerabilities while developing WEB software code based on real penetration testing experience. This article provides in-depth technical assistance for SQL injection testing as well as a solid assurance for WEB information system SQL injection protection.

In 2014, Navdeep Kaur, Master's Degree, M. Tech. in Software Systems, Guru Nanak Dev University, Amritsar and Parminder Kaur, Assistant Professor, Department of Computer Science & Engineering, Guru Nanak Dev University, Amritsar published an article on "SQL Injection – Anatomy and Risk Mitigation ". [6] The paper goes through SQL Injection, SQL Injection Anatomy, SQL Injection Mitigation, and GreenSQL in depth.

3 The Proposed Approach for

4 Experimental Results and Analysis

5 Conclusion

References

1. Antunes N, Vieira M. "Comparing the effectiveness of penetration testing and static code analysis on the detection of sql injection vulnerabilities in web services," in 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing. 2009;301-306.
2. Halfond WG, Viegas J, Orso A. "A classification of SQL-injection attacks and countermeasures," in Proceedings of the IEEE international symposium on secure software engineering. 2006;13-15.
3. Patel N, Mohammed F, Soni S. SQL injection attacks: techniques and protection mechanisms. International Journal on Computer Science and Engineering. 2011;3:199-203.

4. Ntagwabira L, Kang SL. "Use of query tokenization to detect and prevent SQL injection attacks," in 2010 3rd International Conference on Computer Science and Information Technology. 2010;438-440.
5. Zhang H, Zhang X. "SQL injection attack principles and preventive techniques for PHP site," in Proceedings of the 2nd International Conference on Computer Science and Application Engineering. 2018;1-9.
6. Kaur N, Kaur P. SQL injection–anatomy and risk mitigation. Cover Story What, Why and How of Software Security 7 Cover Story Developing Secure Software. 2014;9:27.