Cybersecurity Proposal

Ester Aguilera, Jingru Dou, Kelsey Knowlson, Victoria Lien,
Bryce Palmer, Emely Seheon, Lasair Servilla

Due: February 14, 2024

Table of Contents

1 Introduction

SQL, an acronym for Structured Query Language, has emerged as the one of the most commonly used database management languages since its development in 1973.[1] It is used across a broad spectrum of industries, spanning from highly technical sectors to those requiring minimal education levels. Its ubiquitous nature across industries renders mastering the language invaluable not only for its use in myriad of applications, but also for the prevention of potential malicious attacks and exploitation of vulnerabilities.

As SQL databases are frequently used for organizing and quickly accessing a plethora of proprietary and confidential information-such as client or employee names, payment details, healthcare data- unauthorized access or misuse could lead to significant damage to all parties involved. For this reason, our project proposal will focus on evaluating the implementation and potential vulnerabilities of SQL databases, with the primary aim of creating a system more robust against threats.

2 Problem Statement

The specific problem at hand revolves around identifying vulnerabilities within prevalent structures like SQL databases in the back-end architecture of web services and websites. This problem is characterized by the need to enhance security measures due to potential weaknesses in these widely used systems.

The lack of robust safeguards leaves these platforms susceptible to exploitation, thus creating a need for a solution to mitigate these risks effectively. If we succeed in pinpointing vulnerabilities in these widely used structures, we would then attempt to find these solutions.

3 Motivation

As of now, SQL injection is still a reliable and strong attack route that can compromise sensitive data by taking advantage of vulnerabilities in database-driven systems. This concern is raised as organizations increasingly rely on

databases to store sensitive information; the exploitation of SQL injection vulnerabilities poses a significant risk to data integrity and system security. Because so many organizations rely on database systems to handle vital data, a successful SQL injection attack might have disastrous repercussions, from system failures to data leaks. Through the implementation of strong input validation procedures and a thorough understanding of SQL injection vulnerabilities, this project aims to further the more general objective of improving cybersecurity safeguards in database-driven environments.

4 Objectives

The primary objective of this project is to assess and enhance the security of SQL databases by identifying and mitigating vulnerabilities, particularly safeguarding against SQL injection attacks. Our first objective is to identify vulnerabilities. We want to conduct a comprehensive analysis of SQL structures to prevent potential security weaknesses. Our second objective is to develop secure practices. We want to establish standard practices that include things like input validation, parameterized queries, and more. We also want to test these security measures by strategically attempting yo break the system to assess its effectiveness. Finally, we want to implement the solutions based on our results to address the identified vulnerabilities and enhance the security of the system.

5 Methodology

In order to test for vulnerabilities in a legal manner we will create a simplified version of a web interface to a SQL database following the standard practices for safe guarding such systems. An example standard being the sanitizing of user input by treating it as a parameter rather than a string. Once we have a functioning service we will use our knowledge of how the system was set up to strategically attempt to break it. Breaking the system would mean gaining information from the database that was private or in a unintended manner.

6 Expected Outcomes

The successful implementation of this project is expected to yield several outcomes that will contribute to enhancing the security of SQL databases, focused specifically on mitigating risks associated with SQL injection attacks.

Through the implementation of a secure SQL database with encryption techniques, the project strives to protect against unauthorized access by users, ensuring that even in the case of an attack, any exposed data remains secure. To mitigate the risk against data manipulation, the project will focus on incorporating measures that prevent alterations to sensitive data, such as input validation and parameterized queries. By addressing these vulnerabilities, the expected outcome is a more secure environment that safeguards against exploitation and protects the integrity of data.

7 Conclusion

In summary, this proposal presents an in-depth overview of the importance of recognizing and addressing vulnerabilities associated with SQL databases. Given the critical role that SQL databases play in managing and storing sensitive data, the objective of this project is to focus on the security of SQL databases and mitigating risks related to SQL injection attacks. Furthermore, by fortifying the database with encryption techniques, the project aims to add an additional layer of protection to prevent against unauthorized user access and data manipulation. By setting up an environment that replicates a web service linked to a SQL database in accordance with established secure practice principles, this environment will allow for the assessment of potential vulnerabilities and the testing of potential security measures. Upon successful completion, this project is expected to determine a database structure that defends against potential attacks and vulnerabilities. By addressing risks associated with SQL injection and enhancing data integrity through the implementation of encryption techniques, this project will establish measures for creating a more secure database environment.

References

[1] LearnSQL.com. "The history of sql: How it all started." (2022), [Online]. Available: `https://learnsql.com/blog/history-of-sql/` (visited on 02/12/2024).