Enterprise Level SQL Injection Issues and Countermeasures

Amir Gill[1], Muhammad Kashif Siddiqui[2]

[1]*MSCS SZABIST, Karachi*
amirgill06@hotmail.com

[2]*National Bank of Pakistan*

Abstract: *Today, information is a vital and essential resource of all organizations and key to success and growth. Having the right and accurate information at the right time can make the difference between success and failure.*
On the other hand, security of information assets is a vital aspect of an organization. The impact of a security breach may be worse than the expected. Loss of critical and sensitive data/information may not only directly affect cash flow and competitiveness but also damage reputation of an organization. Many organizations uses web on both public Internet and Intranets for their critical information exchange that directly come from the databases. There are numerous threats to the database management system such as SQL Injection, denial of service, excessive privilege abuse, legitimate privilege abuse etc. SQL Injection is an increasingly dangerous threats to security of information upon databases, however though web applications are the most targeted one. This paper discusses the SQL Injection with its countermeasures that help organizations to secure their information assets.

Keywords: *SQL Injection, Dynamic SQL, Information Security, Database Security, Application Security.*

1 1. INTRODUCTION

Today, information is a vital and essential resource of all organizations and key to success and growth. Having the right and accurate information at the right time can make the difference between success and failure.

On the other hand, security of information assets is a vital aspect of an organization. The impact of a security breach may be worse than the expected. Loss of critical and sensitive data/information may not only directly affect cash flow and competitiveness but also damage reputation of an organization.

Many organizations uses web on both public Internet and Intranet as their critical Information exchange. The use of database management system (DBMS) in web application & e-commerce solutions has increased the likelihood of attack on database management system indirectly that is through Internet or Intranet.

This paper provides the awareness and countermeasures for effective security in respect to SQL Injection that help organizations to increase the level of security regarding web applications. This paper focuses on SQL Injection vulnerabilities, attack methods and preventions.

1.1 Scope

Before defining the scope of this research paper, let's classify the attack methods of SQL Injection. SQL Injection's attack methods are classified into direct and indirect attack.

Indirect methods are those which are executed through web applications. Its main goal is to directly attack the RDBMS and the data it's stored. In direct attacks, an attacker directly tries to take control of the DBMS and may further aim to gain control of the host computer.

Direct method is out of the scope of this paper merely because of two reasons:

- It communicates directly with the DBMS and not aim of the attacking on the DBMS itself.

- It relates to network security because flaws like open ports can be prevented by implementing the network security countermeasures.

This paper covers the indirect methods of SQL Injection and its countermeasures.

2. INFORMATION SECURITY

Information security means protecting information, information assets and information system from unauthorized access, manipulation, disruption or modification.

Information security must support the goal or mission of the Information security must support the goal or mission of the organization. Organizations must need to protect their information assets and also decide the level of risk that can be acceptable to them when determining the cost of security controls. It’s not necessarily that the newest, best, or costliest technology is the right solution for every organization.

3. DATABASE SECURITY

One of the technology terms that most of the people are familiar or usual in hearing either at workshops, meetings, office, students at universities or while surfing the internet is the database. [6]

A database is a collection of information that is organized in such a way so that it can easily be accessed, managed, and updated. Traditional databases are organized into Records containing Fields. Each field contains specific information. Database can be thought of as an electronic filing system.

As database provides the functionality or capability for storing, searching, manipulating and accessing large amount of data with ease, they are key elements of many applications in organizations.

A telephone book is an example of a database.

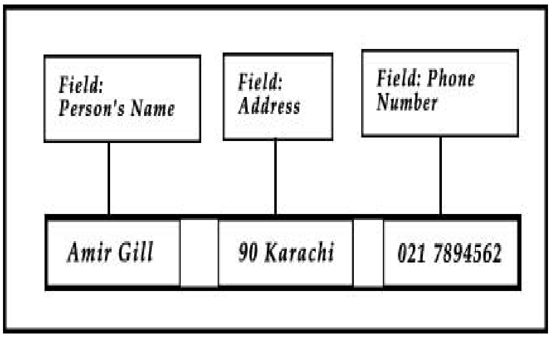

Figure 1: Database Example - Telephone Directory Record

Many organizations uses Web on both public Internet and Intranets as their vital information exchange. As it is very easy to distribute data across networks but it is equally important to protect and ensure that the valid information is only accessible and exchange to authorized users. The use of database management system (DBMS) in web application & e-commerce solutions has increased the likelihood of attack on database management system indirectly that is through Internet or Intranet.

Due to the most valuable and important information the databases stores such as financial or personal details, they are often a target of attack.

There are numerous threats to organization’s database infrastructure such as excessive privilege abuse, legitimate privilege abuse, privilege elevation, SQL Injection.

3.1 SQL INJECTION

In general SQL Injection is a technique used to exploit or attack database through injecting SQL queries / commands to an RDBMS. SQL Injection can be applied to any applications where it has the tendency of input and direct link to database, though it has vast impact or attack on web applications as often web applications may use user-supplied inputs and inject SQL query to the database. One of the examples is user login/password field.

3.2 Objectives

The objective of SQL Injection attacks can be defined by the security services: [2]

- **Access Control:** Access control enables the users to access and manipulate the data according to their privileges defined.
- **Availability:** Services offered by web server must be available when they are requested.
- **Authenticity:** Ensuring that the users who log to the system are the one who they claim to be.
- **Confidentiality:** Ensuring that the Information / data must be kept secret.
- **Integrity:** Information consistency must be maintained at all times.

3.3 Basics

A web application can be viewed from the attacker’s perspective as consisting of the following layers: [2]

- **Desktop layer:** It enables computers using web browser to access a system.
- **Transport layer:** It represents the web.
- **Access layer:** It represents the entry point into the organization system through web.
- **Network layer:** It represents the organization internal network infrastructure.
- **Application layer:** It involves web server, application servers, application’s logic and data storage infrastructure.

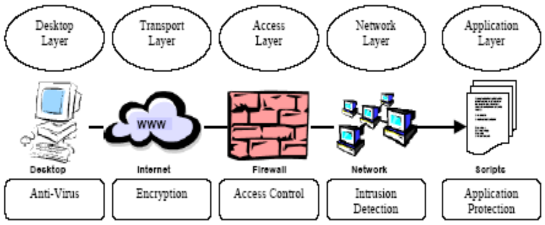

Figure 2: Source; Security layers in web applications [2]

3.4 Vulnerabilities

Web applications might present vulnerabilities that can be exploited by SQL Injection attacks: [2]

- **Invalidated Input:** Invalidated inputs or uncheck parameters to build dynamically SQL queries can be used for SQL Injection attack.
- **Error Message Feedback:** Error messages displayed on client web browser generated by

RDBMS or server-side program can be useful to obtain database information in order to construct their attack.

- **Uncontrolled Variable Size:** Variables that allow storage of large data than expected can be helpful to attacker to enter modified statements.
- **Variable Morphism:** Variables that allow automatic conversions or allow storing other data than expected.
- **Kind Privileges:** Every Web application connects to database using a specific account for accessing. The more privileges given to account the more number of available attack methods and object associated with database are affected.
- **Dynamic SQL:** Creating a query dynamically helps the attackers to his inject attack statements making the query totally different.
- **Stored Procedures:** As stored procedures are well known and easy to execute, they can be helpful to attacker for casing damage to the RDBMS as well as the host computer or other networks.
- **Client-Side-Only Control:** Usually input validations are implemented in client-side script only for performance purpose but this can help attacker to override using cross-site scripting.
- **INTO OUTFILE Support:** RDBMS that supports INTO OUTFILE clause can help the attacker to operate the SQL queries to produce text file that he can access later.
- **Sub-Select:** RDBMS that supports sub-select statements may help attacker to inject additional SELECT clause in WHERE clause.
- **JOIN / UNION:** RDBMS that supports JOIN or UNION statements may help attacker with more variations of attack methods.
- **Multiple Statements:** RDBMS that supports multiple statements may help attacker with more variations of attack methods. For instance attacker can add unwanted SQL statement.

3.5 Attack Methods

There are number of ways to accomplish SQL Injection attack and it depends upon what security services to cause danger to and what are the vulnerabilities a web application. They are group into two main categories: [2]

Data Manipulation: In data manipulation method attacker can bypass the authentication process to retrieve, change, fabricate or delete data in the database. This method is also known as indirect method.

Command Execution: In Command execution method, attacker executes the SQL statements / commands through the RDBMS and may further take control of other host computers in the network. This method is also known as direct method.

4. COUNTERMEASURES

The countermeasures that are presented in this research paper are fully technical that helps in preventing against SQL Injection attacks. [2]

- **Different Accounts:** Default accounts that come with the RDBMS such as with administrative privileges should never be used for web applications. Instead different accounts should be created according to the profiles.
- **Limited Privileges:** Web application that use database account should be given privileges that are only required. Also SELECT is used when user login to the system, not any other privileges such as DELETE or INSERT.
- **Static SQL:** Use of Static SQL can prevent attack because it cannot be altered by inserted SQL keywords.
- **Error Handling:** Error messages generated by the RDBMS or web servers should never be passed back to the cline side.
- **Input Validation:** Input validation implemented at client side is for performance purpose, however every parameter or value sent from the client side should be properly examined and validated by the server side.
- **Character Escaping:** Characters such as quotation marks or semicolons must be allowed, and replaced by ASCII code or using bind variables.
- **Stored Procedure Limitation:** Limit the use of the store procedures and remove all other which are not in use for web application’s database account.
- **Variable Size:** Control the length and size of the variable. Proper check should be made to check length such as in java, check the length of the string variable by length function.
- **Strong Typing:** Make sure that variables are explicitly typed, however If need of weak type variable then it properly be checked.

5. FUTURE WORK

In this research paper I have focused on the attack techniques of SQL Injection and prevention from it. I believed that this area is in need of further investigation mainly because as SQL Injection evolves and new vulnerabilities will be found then there should be new countermeasures to deal with them. Secondly I have listed some prevention techniques according to the factors I considered, to my concern there should be considered more factors in broad range so that proper countermeasures should be discussed.

6. CONCLUSIONS

One of the goals of this research paper was to increase the level of security awareness among organizations regarding web applications, especially towards SQL injection threats. I hope that further research in this area and in related web application will help in implementing the security standards and countermeasures during application development. Ultimately organizations use a proactive approach towards application layer security.

REFERENCES

[1] Ross J. Anderson. Security Engineering. John Wiley & Sons, Inc., 2001.

[2] Uzi Ben-Artzi Landsmann and Donald Str¨omberg, Web Application Security: A Survey of Prevention Techniques against SQL Injection, Department of Computer and System Sciences. Stockholm University / Royal Institute of Technology, 2003

[3] Wei, K. Muthuprasanna, M. Suraj Kothari, Preventing SQL injection attacks in stored procedures, Proceedings to the 2006 Software Engineering Conference, Australia April 2006.

[4] Xiang Fu Xin Lu Peltsverger, B. Shijun Chen Kai Qian Lixin Tao, A Static Analysis Framework For Detecting SQL Injection Vulnerabilities, Proceedings to the 2007 Computer Software and Application Conference, July 2007.

[5] Bertino, Elisa Kamra, Ashish Early, James P. , Profiling Database Application to Detect SQL Injection Attacks, Proceeding in IEEE International Conference on Performance, Computing, and Communications, 2007

[6] Tech FAQ website; Source: http://www.tech-faq.com/database.shtml