Sql Injection Attacks in Web Application

Harsh Gupta

Guide Name: Prof Jasbir Singh Saini
Department: Computer Science & Engineering
College : Guru Nanak Dev Engineering College Ludhiana(Pb)

Co-Guide: Dr. Ajav Shiv Sharma
Department: Information Technology
College: Guru Nanak Dev Engineering College Ludhiana (Pb)

Abstract : In terms of web application, Databases are the most vulnerable part of the application that harms by the attackers. In Web Application, when ID and PASSWORD are retrieved by the user, they can misuse the application at large. This paper discuss about SQL Injection (SQLIA) attack and its identification then according to that prevention and detection measures are suggested. Some additional features are added to it Web Services and Advance SQL Injection (ASQLA) which will focus on more Security of Web Application. In short improving the database and web security is the objective of my paper with reference to developing community.

Keywords: SQL Injection, Tautology, SQLIA, Blind injection, Piggybacking, PSIAW.

I.INTRODUCTION

Now a days, web applications have become one of the most important communication media between service clients and service providers. The increasing complexity of web based application and its ratio of attacks has raised awareness of web application developers and system administrators to effectively protect their web applications at all levels of the application. SQLIAs are performed by hackers that are responsible for inserting a malicious SQL query into the web application to manipulate data, or to access the back-end database. So, number of SQLIA"s reported in the last decade and showing a steadily increasing trend and graph is increasing day by day. It is, therefore, of prime importance is to prevent these types of attacks, and SQLIA prevention has become one of the most important topics of research in the industry and laboratories.

There has been significant progress in this field and a number of models have been proposed to develop and counter SQLIA"s, but no one is able to guarantee an absolute level of security in complex web applications, mainly due to the diversified nature of the application and scope of SQLIA"s. One Common practice of programming in current time is to use a SQLIA's database stored procedures rather than use of direct SQL statements that will interact with different databases in a web application.

2 Procedure for Paper Submission

2.1 Review Stage:

In review process of research paper, various attacks such as Tautologies, Piggy-backed queries, union query, stored procedures, inference, blind injection, timing attacks and various techniques for preventing the Sql injection such as generating a hashing, validation, string matching were followed .

2.2 Final Stage

In final stage of the paper, technique is proposed for preventing SQL injection attack in web application (PSIAW) that is based on the comparison of username with level at the backend.

2.3 Figures

SQL INJECTION ATTACK: It is defined as the web application security attack in which an attacker is able to enter a database by way of SQL command, which is run by the web application, that interacting the back-end database. SQL Injection attacks can happen when a web application accepted user- inputted data without proper checks & validation or different encoding as part of query string or a simple command.

SQL injection is a software threat that occurs when data is entered by the end user and sent to the SQL interpreter as a part of an SQL query

USER-id:

ABC

Password

23

Select * from login where USER-id='ABC' and *PASSWORD='*123';

USER-id

or 1=1;/*

Password

–

Select * from users login USER-id=" OR 1=1;/* and password =*/–;

Figure- SQL Injection:

In this figure attackers provide specially manipulated input data to the SQL interpreter and to execute unintended commands. So, hackers utilize this vulnerability by providing specially manipulated input

data to the SQL interpreter in such a way that the interpreter is not able to differentiate between the designed commands and the attacker's specially designed data. So, interpreter is challenged to executing malicious commands.

SQL injection uses security vulnerabilities at the layer of database. By entering the SQL injection flaws, attackers modify or delete sensitive data at all layers

Types Of Sql Injection Attack

There are different ways of attacks that depends on the goal of attacker are performed together or serially. For a successful SQLIA, the attacker should modify a correct command to the desired SQL query. So, the classification of SQLIAs is as followed .

Tautologies : In this type of attack injects SQL tokens to the conditional query statement to be evaluated which always true. This type of attack used to bypass the control of authentication and access to sensitive data by using vulnerable input field which uses the WHERE clause."SELECT * FROM reg WHERE user id = '115' and password ='abc' OR '1'='1'" So this tautology statement (1=1) added to the query statement which is always true.

Ilegal/LogicallyIncorrectQueries: : In this attack, a query is rejected when it is passed to the system interface and an error message is generated and access of necessary information database including necessary information. This error messages helps an attacker to find vulnerable parameters in the application and further accessing the data base of the application. In fact attacker injects malicious input or SQL tokens in query to generate syntax error, logical errors mismatching of type by its own purpose. In said example attacker did a type mismatch error by changing the following text into the *pin* input field:[3]

1) Original form of query string

URL:http://www.abc.poli.it/eventi/?id_nav=1024

2) SQL Injection attack: http://www.abc.poli.it/eventi/?id_nav=8864'

3)Error message showed:(SELECT name FROM student WHERE id =8864\') From the message error we can find out name of table and fields: name; Employee; id. By this attacker can gained information and organize more number of strict attacks.

Piggy-backed Queries: It is defined as the attack in which intruders destroy the database by the query delimiter, such as ";", and to append extra query to the original query. With this attack database receives and execute a number of different queries. Normally the original query is legitimate query, whereas below queries could be illegitimate. In this way attacker can inject different SQL command to the shell of database. In this example, attacker inject " 0; *drop table user* " into the *pincode* input field else by inputting the any logical value that changes the structure of the database. So, the application would generate the query: (SELECT info FROM users WHERE login= 'abc' AND pin=0; drop table users) Because of ";" (terminator) character, shell of database accepts both queries and executes them. The second query is malicious and can drop table having name users from the database. It should be noticeable that many databases do not need special character separation in number of different queries, so for detecting that type of attack, scanning for a special character is not good solution.

Union Query:

In this technique, attacker's uses join injected query to the original query by the word "UNION" and then can access the data about other tables from the application. Suppose for example that the query which is run by the server is at follows.

SELECT Name, Phone FROM customer WHERE identity=$id then injecting the following Id value: $id=1 UNION ALL SELECT CardNumber,1 FROM Card Table We will have the following query: SELECT Name, Phone FROM customers WHERE Id=1 UNION ALL SELECT CardNumber,1 FROM Card Table which will show the result after joining the original query with all the card users.

Stored Procedure: Stored procedure is a function used in the database methodology in which programmer might set an extra abstraction layer on the database. As stored procedure could be programmed by programmer, so, that part is highly inject able as web application uses the forms controls. So, depend on specific stored procedure programmed in the layer of database there are numerous ways to attack the application. In the following example, attacker harms parameterized stored procedure.

CREATE PROCEDURE DBO .is Authenticated @ username varchar2, @password varchar2, @pincode int AS EXEC("SELECT * FROM users WHERE loguser='" +@username+ "' and passuser='" +@password+ "' and pincode =" +@pin);

Timing Attacks:

A timing attack is defined as an attacker has to collect information from a different databases i.e bulk of databases by observing delays in the database's response time. So, this technique uses conditional clause i.e if-then statement that cause the SQL engine to run a continuously running query or used a statement that having delay response time that depends on the inputted injected logic. This attack is very similar to blind injection and attacker/hacker can easily measure the time i.e the page takes time to load to determine if the malicious statement is true which is entered by the user. So, this technique uses an conditional statement for injecting/altering queries i.e if, then . WAIT FOR is a keyword that effects the response time of the database for its a specific time Period. For example, in following query: declare@varchar(2000)select@= db_nameO if (ascii(substring(@, 1, 1)) & (power(3, 0))) > 0 wait for delay '0:0:5' Database will be pause for five seconds if the first bit of the first byte of the current database is 1. After this attacker check the status of the condition then code is injected to generate a delay in response time Also, attacker can ask a number of other questions about this character. As these examples show, vulnerable parameter is used for extracting the information from the user.

2.4 Copyright Form

For the sake of authorization, a separate form is attached

3. Proposed Technique : Our research work proposes the technique, preventing SQL injection attack in web application (PSIAW).

In this Login form having two fields are designed such as Username and Password and a login table in database name 'ims' having columns Username and password are created by DBA that having other columns also. Our methodology requires session variable in a coding page. In designing page, id is created in input controls having name 'username and password. When user enter the value in the login form i.e. Username and password and submitting the control, the value will be fetched with the help of id and called with the help of post method and function used for passing the value to the variable is

mysql_real_escape_string (strip slashes( ))that having ability to accept the accurate string. The desired username and password is matched in the database login table and result will be set in the variable by using function mysql_query and login identity is set in session variable that is used in throughout the application for identification purpose. In this, user can access the application by getting the appropriate level of the user that otherwise further part of the application will not be processed.

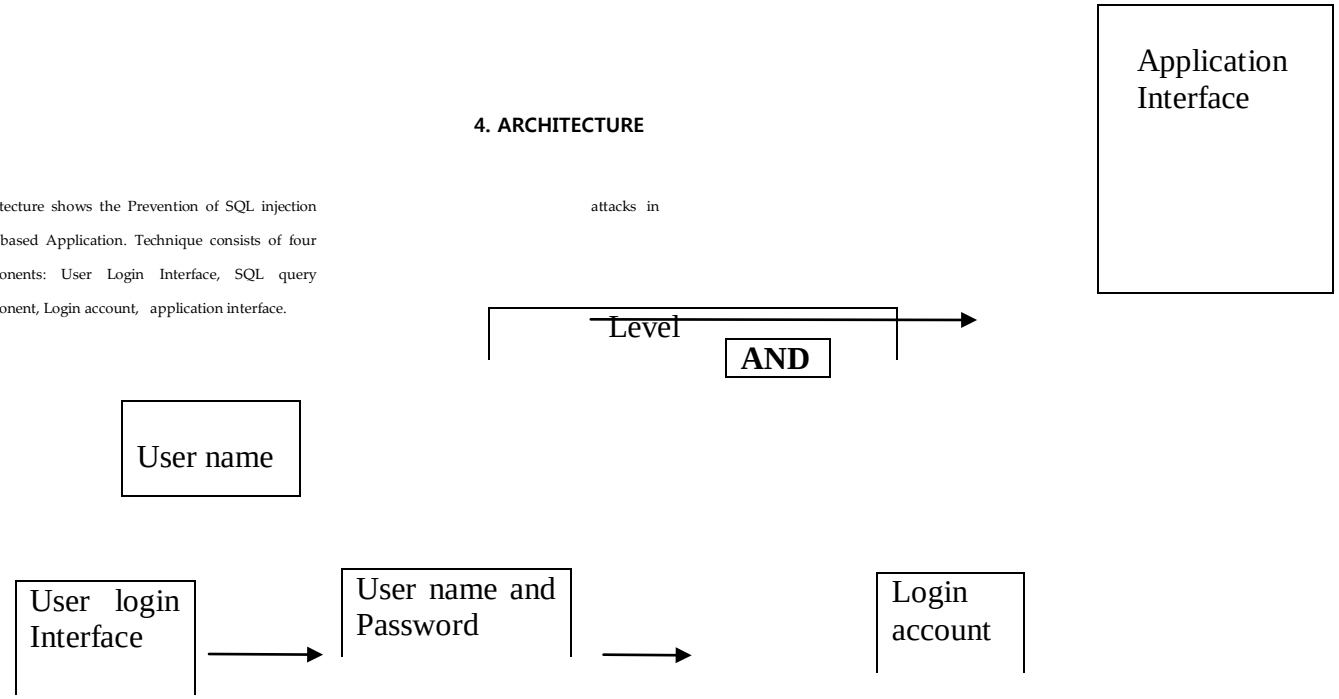
In this Interface only two boxes are provided, malicious user cannot enter the level. Hence the attacker will not access the database and web application is secured. For the Sake of securing the database, other fields also have been designed that they don't have directly access to the user. i.e User cannot submit the data directly in their fields as the controls is in the hands of admin panel and the security of the admin panel depends upon the community of the server members.

In this Technique, firstly inputted string has to pass via methods that is used in the filter mechanism. Query is filtered as per the designed method and extra quotes are highlighted and vanish so that it cannot change the original meaning of the coding. In this, input string is fetched via variables that I used for data storing and processing purpose. Only required variables are called and matched as per the design of the application. The access rights of the variable are in the control of admin panel of the developer's community. The variables are of many types that can be act as locally and globally and is liable for making the security. In this Technique, required input string has to pass the flow of the application as there is level wise security is defined.

In context of this technique, session variable are used, it can be started and destroyed as per the developer way of coding the application. In web application, the purpose of scripting language and how it can embed into the web page is matter of importance .For the sake of detection and prevention of injection attacks level of security mechanism will be preferred.

4. ARCHITECTURE

Architecture shows the Prevention of SQL injection attacks in Web based Application. Technique consists of four components: User Login Interface, SQL query component, Login account, application interface.



"Architecture of Preventing SQL Injection Attack in Web Application"

"Architecture of Preventing SQL Injection Attack in Web Application"

Now, user login interface is just the user entry form containing two boxes for username and password. Main component of PSIAW is SQL Query Component. SQL Query component is the component where username and password is selected. These values are then combined with using AND operator. Every time the user enters username and password, their result is set in the session. The query formed is then sent to database. Subcomponents of SQL Query component are username, level. User login table is the component where username, password, for user and passwords are stored here.

5.CONCLUSION : In many web applications a middleware technology is designed to request from a relational database in SQL context. SQL Injection is a technique that is used commonly by the hackers to inject in these web based applications. These attack alters the SQL queries and changing the behavior of the program for the benefit of the hacker.

In our research work, we have presented a technique for protecting authentication against SQL Injection. This technique presents the need for creating adding additional columns in login table. These columns store level values of that matches with username and password in application. When the user gets itself registered with a web application, it selects its username and password. In this application level is assigned by the admin panel. If SQL Injection attack i.e String is entered for logging into the database, it cannot access the database as level field is not present in the interface.

6.FutureWork: This technique used to protect application authentication, and preserve the security of database. Rest of the SQL Injection techniques can't be prevented for heavy end applications. So, in future, we will try to improve the technique by the way of protocol mechanism as application is bigger and more complex day by day. As this technique covers the level wise security mechanism, so various checks and comparison will be performed while using the application and it ensures the data integrity and flexibility of application at all levels. So in this way application will be adaptive to the user requirement that ensures high level of security. As in multi national organization, users are huge in number, so Application will be adaptive to meet the requirement of said etc. This technique can enhance the flexibility of database design as one can modify more number of fields and different operation would be performed to achieve the desired result. As user can filter number of queries and in the end different types of report will be generated by the different users having distinct access levels. So, this technique consists of distinct structured components and these components of architecture will be modified and accessibility of these will depend upon the requirement of the application and its implementation.

7. ACKNOWLEDGEMENTS

The author wishes to thanks Prof Jasbir Singh Saini Department of Computer Science & Engineering, Guru Nanak Dev Engineering College Ludhiana(Pb) & DR.Ajay Shiv Sharma Department of Information Technology, Guru Nanak Dev Engineering College Ludhiana(Pb) for their contribution during research work.

REFERENCES

[1] Gandhi Mihir, Baria Jwalant " SQL Injection Attacks in Web Application" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307 Volume-2, Issue-6, January 2013.

[2] By Mayank Namdev *, FehreenHasan, GauravShrivastav "Review of SQL Injection Attack and Proposed Method for Detection and Prevention of SQLIA" Volume 2, Issue 7, July 2012.

[3] By AtefehTajpour , "Suhaimi Ibrahim, Mohammad Sharifi "Web Application Security by SQL Injection DetectionTools". IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 3, March 2012

Author : Harsh Gupta

Email id : guptaharsh_123@yahoo.co.in