SQL-Injection
A hazard to databases security via web application

1Jagjit Kaur, 2Amrita, 3Mankiran
1,2,3 Assistant Professor
Department of Computer Science and Engineering, Chandigarh Group of Colleges, Landran, Mohali

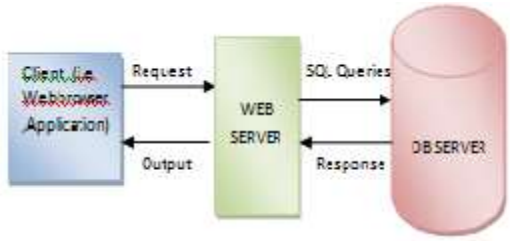
Abstract: **Databases are the first target of the hackers. Weak authentication is one of biggest reason that for which we lose our confidential data. We usually share our sensitive information such as Debit/ credit card numbers, SSN etc without caring of the fact that when our system will interact in the internet through the WWW environment, huge amount of data is being created which may have user's personal information. It may cause a break in the user's privacy, if this data is fetched by any unauthorized party. Hence, the big question is that how can we ensure the database security against cyber-attacks. In recent times, SQL Injection attacks have emerged as a major threat to database security. SQL Injection Attack has been around for over a decade. It allows attackers to obtain unauthorized access at the back-end database to change the intended application-generated SQL queries. Such attacks target to databases through web frontend layer take advantage of flaws happens at user end. For the purpose of security, we have surveyed various SQL injection techniques which cause threat to database security. We will discuss various technical methods, terminologies used to handle these attacks.**

Keywords: **SQL, WWW, DB.**

1. Introduction

SQL injection is a type of attack which the attacker adds SQL code to a text input fields to obtain access or make desirable Changes to data [1].SQL means structured query language. With the help of SQL language we can perform various operations such as insertion, deletion, updation as well as can create indexes, procedures & can create databases. SQL injection has become one of security flaw allows an attacker to indulge queries with database and destroy functionality .With changing life styles & advancement of technology, undoubtedly we are getting more dependent on the web applications for the fulfillment of our daily routine needs like online shopping, online banking, share trading, ticket reservation booking, online transactions etc has increased. Because of this, our confidential data is present in the databases of various applications on Web [5]. Basically:-

1) SQL injection is a type of attack where user supplied data passed to a web application with improper validation.
2) How basic structure works, we can look out in following diagram:-



[1.1 Request-Response Architecture]

In this figure we have shown that how simply Request/Response architecture works. Firstly client side program send a request to web server which is an intermediate between client and database server. Then web server convert the request to SQL query request and send to database server. Then DB server responds to that query. Then web server gives the desired result to client side program. For e.g.:- Web application send an input form to user, when user submit input form, Meanwhile attacker attach some malicious code with data[4].

- Attacker submits form with SQL exploit data.
- Application builds string with exploited data.
- That web application passed SQL queries to database server.
- Database execute query including exploit, sends data back to application.
- Application returned data/resultant output to user

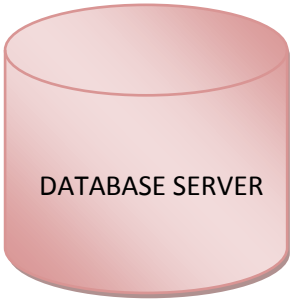
SQL injection attack works in role when an unauthorized user through specifically crafted input, causes a web application to generate and send a query that functions differently than the programmer intended. SQL injection can threat authority, integrity and confidentiality, availability of databases but they are not sufficient [7].

2. EXAMPLE APPLICATION:-

A SQL query look like this:-

Select * from LOGIN; //selects everything

LOGIC: 'a'='a'

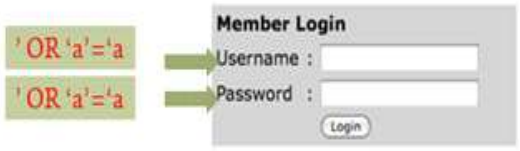


Example: SELECT * FROM `table` WHERE 'a'='a'

SELECT * FROM `login` WHERE `user`='jaggs' AND `pass`='cse'

[2.1 User Log-In form]

Now what attacker does:-

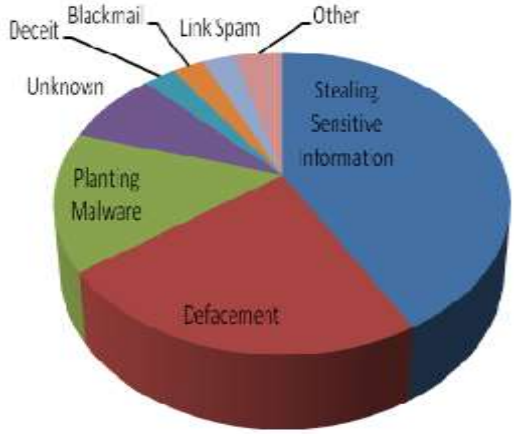


[2.2 User's SQL-Injected Log-In form]

SELECT * FROM login WHERE `user`='' OR 'a'='a' AND `pass`='' OR 'a'='a'

3. IMPACT OF SQL INJECTION:-

1. Leakage of sensitive/confidential information.
2. Modification/updation of sensitive information. such as:-
- Add new data to the database.
- Perform an INSERT in the injected SQL.
- Modify data currently in the database.
- Perform an UPDATE in the injected SQL.
- Often can gain access to other user's system capabilities by gain access on their password.
3. Loss of control over database server.
4. Loss of user Data.
5. Denial of service.



[3.1 Effecs of SQL-Injection]

4. Types of Sql Injection Attacks:-

4.1 Tautologies:-

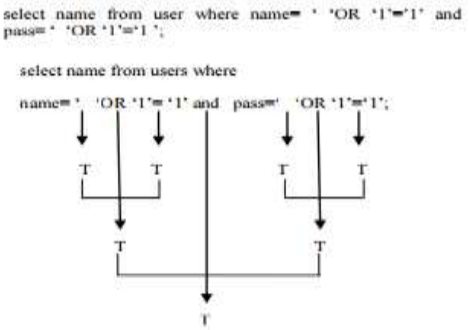
This is one of SQL injection attack based on Boolean algebra theory. In this case attacker adds such code with user's data with some conditional statement which always evaluated as true. If query become successful then it will display all the records available in the database. Such as attaching PASSWORD='OR 1=1—'.

Boolean theory evaluated it as true and hence permitted access to unauthorized user.sql query look like this before code attached.

Select * from table where user_name=username AND password=password ;

SQL query look like this after code attached.

Select * from table where user_name=username AND password=password OR =1=1;



[4.1 Tautology Method]

The code injected in conditional phrase transforms the where clause to tautology which every time turns out to be true and helps to extract the data[8].

4.2 Union Query SQL INJECTION ATTACK:-

The union operator is basically used to join a query, In this form of attack, Attacker try to harm database by injecting statement to form by using UNION SELECT<injected query>cause attacker has completely control the another injected query they can use that query to extract information from a specific table[10]. This will give the results of original first query as well result for injected another associated query. If attacker inject code in this way as

'UNION SELECT * from table WHERE User_Name=username'

The query wil be look like this:-
if the attacker inject a code like

'UNION SELECT * FROM tbl_user WHERE user_name = username'.

The query would be like this

SELECT * FROM tbl_user WHERE user_name = username AND password = '' UNION SELECT * FROM tbl_user WHERE user_name = username;

If no password exist for user,then Null value will be returned for 1st query but the injected cod will return all information of that username.Again the attacker can merge the resultant values and will gain the access to the application without having password of user.

4.2.1 Second Case:-

Suppose the query executed from the server is the following:

Let Name,phone,address are the column fields in creditcard(tablename)

SELECT Name, Phone, Address FROM Users WHERE Id=$id;

$id=1 UNION ALL SELECT creditCardNumber,1,1 FROM CreditCardTable;

We will have the following query:

SELECT Name, Phone, Address FROM Users WHERE Id=1 UNION ALL SELECT creditCardNumber,1,1 FROM CreditCardTable

Which will join the result of the original query with all the credit card numbers in the CreditCardTable table?

4.3 Blind injection:-

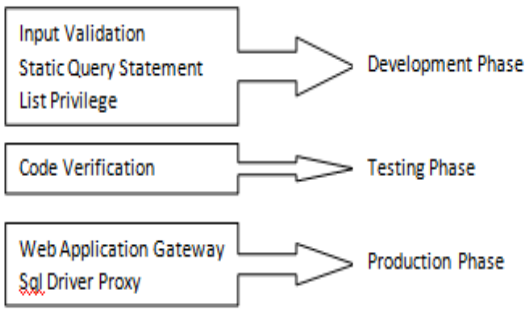
What basically attacker does:- they use the error messages from the database to check out ther areas prone to be effected from injection to determine whether or not that database is vulnerable to a SQL injection attack.? This attack is often used when the web application is configured to show generic error messages.Blind sQL technique determine the answer based on application response[12].
suppose that we have site – Say itcompany.com – that has different profiles for people having all the information of the people[16].Each user on the site itcompany.com has a unique ID number assigned to them that identifies them & query string is used to extract each individual's profile .If the user ID of 111 belongs to a user named "xyz". Let's assume that the user ID would be used to retrieve the user's profile details (like links to pictures, his/her personal information, DOB etc) from a database. So, if a user requests the URL 'http://www.itcompany.com?id=111 ', then that query string would be used to run some SQL on the servers of itcompany.com. That SQL could look like this:

SELECT * FROM tbl_user WHERE user_id = 111;

5.1 Prevention architecture for sql injection:-

There is some prevention measure for ensuring database security. If these rules are enforced there may be huge degradation in percentage of losing user data information[2]

- Prevention by checking syntax of input fields for validity because
- Many classes of input have fixed languages which can Verify that the input is a valid string in the language
- Excluding quotes and semicolons may works.

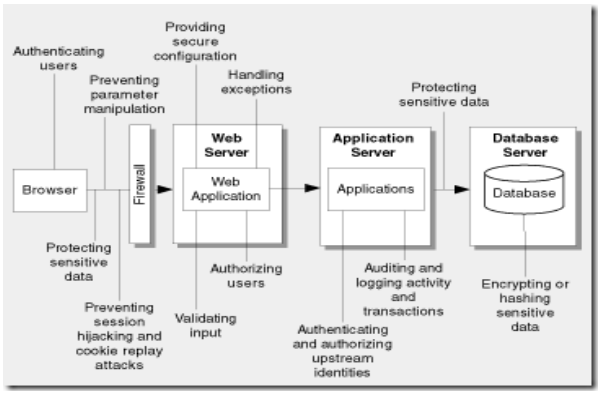


[Prevention technique against SQL injection]

[5.1 Prevention Technique]

- Logic to allow only numbers / letters in username and password.

- Enforce constraint on client side as well as server side.
  - Many SQL injection attacks depend on entering long strings
- Scan query string for undesirable combinations of SQL statements
  - INSERT, DROP, SELECT, UNION, ORDERBY etc.
- Limit database permissions and segregate users



[5.2 Modern Request/Response architecture]

6.1 Sql injection conclusion:-

In this paper we survey and analyze different SQL Injection attacks. SQL injection is technique for exploiting applications that use relational databases as their back end. SQL injection use the fact that many of these applications concatenate the fixed part of SQL statement with user-supplied data that forms WHERE predicates or additional sub-queries. The technique is based on malicious user-supplied data & transforms the innocent SQL calls to a malicious call which can cause unauthorized access, deletion of data, or theft of information. The vulnerability is in the application layer outside of the database, and the moment that the application has a connection into the database. It has been shown that how these attacks reshape the SQL queries, and then alter the behavior of the program for their own intended purpose. In our research work, we have presented some preventive measures for protecting authentication against SQL Injection. . Regular users can safely perform operations on database by following simple preventive measures which stop attacker to not access the database.

References

1. Atefeh Tajpour et al, "Evaluation of SQL Injection Detection and Prevention Techniques", *Centre for Advanced Software Engineering (CASE), University Technology Malaysia, Kuala Lumpur, Malaysia.*
2. Prithvi Bisht et al,"CANDID : Preventing SQL Injection Attacks Using Dynamic Candidate Evaluations",Madhusudan,and V. N. Venkatakrishnan, University of Illinois, Chicago.
3. FehreenHasan et al,"Review of SQL Injection Attack and Proposed Method for Detection and Prevention of SQLIA"Volume 2, Issue 7, July 2012.
4. S. W. Boyd and A. D. Keromytis. SQLrand: Preventing SQL Injection Attacks. In *Proceedings of the 2nd Applied Cryptography and Network Security (ACNS) Conference*, pages 292–302, June 2004.
5. Susanta Nanda, Lap Chung Lam,"Web Application Attack Prevention for Tiered Internet Service" Fourth International Conference IEEE 2008.
6. William J. Halfond and Alessandro Orso,"AMNESIA: Analysis and Monitoring for NEutralizing SQL Injection Attacks, College of Computing Georgia Institute of Technology.
7. Prasant Singh Yadav, and Dr pankaj Yadav,"A Modern Mechanism to Avoid SQL Injection Attacks in Web Applications",Ijrrest- International Journal of Research Review in Engineering Science and Technology ,Volume-1 Issue-1, June 2012.
8. Sruthy Manmadhan et al, "A Method of Detecting SQL Injection Attack to secure web Applications" nternational Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.6, November 2012.
9. Ravindra Kumar Purwar et al,"SQL INJECTIONS – A Hazard to Web Applications" International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 6, June 2012.
10. X. Zhang, C.J. Lin et al"TransSQL: A Translation and Validation Based Solution for SQL-injection Attacks,"in Proceedings of First International Conference on Robot, Vision and Signal Processing, 2011, pp. 248–251.
11. Kasra Amirtahmasebi, Seyed Reza Jalalinia," A Survey of SQL Evaluation of Popular Copy-Move Forgery Detection Approaches", *IEEE transactions on information forensics and security*, 2012.
12. Dr.Manju Kaushik et al,"SQL Injection Attack Detection and Prevention Methods: A Critical Review" International Journal of Innovative Research in Science, Engineering and Technology ISO Vol. 3, Issue 4, April 2014.
13. Sadeghian, A.; zamani, M.; Abdullah, S.M., "A Taxonomy of SQL Injection Attacks," Informatics and Creative Multimedia (ICICM), vol., no., pp.269,273, 4-6 Sept. 2013.
14. Shaukat Ali, Azhar Rauf, Huma Javed "SQLIPA:An authentication mechanism Against SQL Injection".
15. K. Amirtahmasebi, S. R. Jalalinia, S. Khadem, "A survey of SQLinjection defense mechanisms," Proc. Of ICITST 2009, vol., no., pp.1-8, 9-12 Nov. 2009.
16. Shubham srivastava,"A Survey On: Attacks due to SQL injection and their prevention method for web application" (IJCSIT) Vol. 3 (1), 2012, 3225-3228

