RESEARCH PAPER | Engineering | Volume : 5 | Issue : 5 | May 2015 | ISSN - 2249-555X

Proposed Preventive measures and Strategies Against SQL injection Attacks

KEYWORDS

Gaurav Parmar | Dr. Kirti Mathur

ABSTRACT *Now a days so many cyber crimes are occuring. Most of the cyber crimes involves SQLi attacks. It is a popular attack used by hackers to grab confidential information from the database system for which they are not authorized. Sometimes it results in defacement of website in return damaging company's reputation. SQLi and Advanced SQLi attacks are the recent trends in attacking. If in case a bank's website is concerned then this attack can lead to great loss to the account holder and bank as well. This paper aims to aware people about the techniques to prevent themselves from SQLi attacks. Also would be useful to Software Developers, Software Designers and Software owners. By using the methods that we have proposed in this paper, one can make his application more secure and reduce the probability of getting attacked.*

I. INTRODUCTION

Database is an essential part of software. When small software which does not hold any sensitive data, is concerned then leakage of information can be ignored but in software containing confidential information, leakage can mean a catastrophe. Mostly in case of banking software if credit/ debit card information is leaked then it would be irreversible and lead to catastrophe to customers and damage bank's reputation as well. When government websites are concerned then outflow of confidential information can be harmful. Leakage of important information from a company's website can be advantageous for its rivals. In most cases unexpected and unauthorized outflow of data from database is destructive in some or the other way. Keeping a database secure from attacks is a big challenge and when attack like SQLi is concerned then it is a big reason to worry about. SQLi (SQL injection) is one of the most commonly used attack today. SQLi attack is nothing but the injection of SQL statements with the input data by attacker and then execution of this SQL statements after getting attached to SQL query. According to Acunetix [1], SQL injection was the largest data security breach in U.S. history when 130 million credit and debit card numbers were stolen from 5 leading companies by three men. This research paper proposes some SQLi attack prevention techniques.

II. OVERVIEW OF SQLi ATTACK

SQLi or "Structured Query Language Injection" is an unexpected SQL command usually injected through input area(text box) or URL with malicious intention which gets attached with SQL query. This attachment of SQLi with SQL query results in leakage of confidential information from database, unauthorized access and benefit to attacker in some way or the other. SQLi is a malicious code[2].

In case of database-driven applications SQLi attack is million credit and debit card numbers were stolen from 5 leading companies by three men[1]. Due to rise in the use of internet, web application vulnerabilities are also increasing and SQLi attack is one of the most common and easy attack method that has been used by attackers on databases [4].

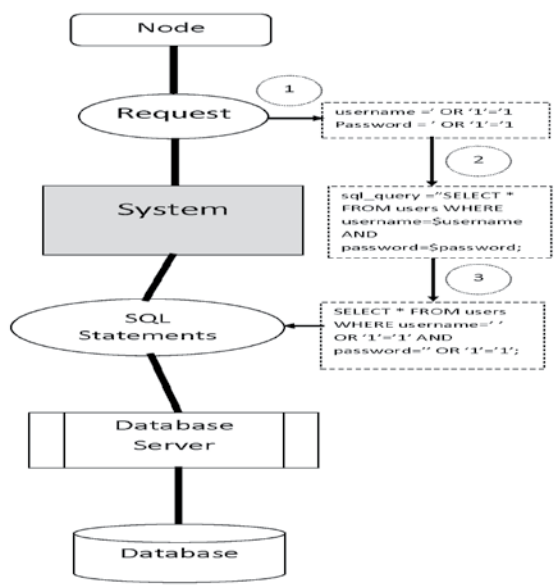

Fig 1. Steps of SQL injection

SQLi attack usually happens due to poor input validation[10]. SQLi is a type of code injection attack that involves injection of harmful SQL commands through input data[11]. Encryption and decryption of data may help preventing SQLi[12]. OWASP has placed SQLi in a list of top 10 vulnerabilities [13]. We can also use database stored procedures in place of direct SQL since it uses parameterized queries[14]. Providing multi-level application security is a better choice since no solution is completely perfect[15]. Attacker often uses input fields or hidden parameters not easily visible to user[16].

III. SOME COMMON TYPES OF SQLi ATTACKS

There are diffrent types of SQLi attacks but out of them some commonly used SQLi attacks include[5] -

Error based SQLi attacks :- In this type of SQLi attack an SQL query gets failed and in return we get some message containing the error and debugging information[6]. This error message can carry secret information.

Volume : 5 | Issue : 5 | May 2015 | ISSN - 2249-555X

Always true statements(tautologies):- In this type of SQLi attack, the attacker uses logically correct statements like '2'='2' which follows "OR", as in "' OR '2'='2 " and this turns out to be dangerous when there is "WHERE" clause in SQL query accompanied by no security for SQLi attack[7].

UNION based SQLi attacks :- In UNION based SQLi attacks, the attacker joins additional query statement by "UNION" clause and the result of this additional query is grouped with the result of actual query[9]. In this way attacker gets confidential information which accidently gets leaked after joining the results of both the queries.

Appending additional SQL queries using query delimiter:- In this type of SQLi attack, the attacker successfully appends his own query by taking advantage of query delimiter such as ;(semicolon). If the second query that is attached to the main query is legitimate then it will also get executed by the SQL interpreter and fulfill attacker's intent.

IV. SQLi ATTACK PROCESS

The process of SQLi attack includes-

1. Injection of SQL command(SQLi) by user through text box, text area or URL.
2. Attachment of user given SQLi in the SQL query.
3. Execution of SQL query having SQLi by SQL interpreter.
4. Leakage of confidential information, unauthorized access or benefit to attacker in some way.

V. PROPOSED PREVENTIVE MEASURES FOR SECURTIY AGAINST SQLi

Most of the SQLi attacks happen through input text provided using text boxes or URL bar. Changes in front-end and back-end can be introduced to provide security from SQLi attacks. An application component can be built in many different ways but out of them we have to opt the one which is most secured.

But there are some cases when the user input is not predefined and hence we have to provide text boxes. So in this case also we have proposed some methods for security in this paper.

1. Use minimum text boxes

Using minimum number of text boxes will reduce entrances for SQLi. It some cases text boxes can be eliminated. For example, date value can be taken by a text box, three text boxes, three combo boxes or jQuery date picker. These first two ways seems vulnerable while others do not.

2. Use radio buttons/drop down maximum possible

Radio buttons or drop down menu can be a better option in case of single user input which is one among some limited known values. For example, to get gender details we can have a text box, three radio buttons or a drop down menu mentioning genders. Here last two options seem to be more secure.

3. Use of check boxes in case of limited predefined choice(s)

Check boxes can serve better than text boxes in cases where user input is certain. For example, to get multiple subject choices from users we can have a text box, multiple text boxes or a group of check boxes labeled with subject names. The last option prevents SQLi.

4. Scrutinize user input

Check user input for SQLi syntax like- ', --, UNION, OR, =, --, #, /*…*/ etc [9].

5. Use language specific SQLi attack prevention functions

Many programming languages like - PHP provide functions like mysql_real_escape_string() to prevent SQLi attack.

6. Assign database access rights

Defining database user rights and restricting database access for normal visitor or users of some category can also aid in prevention from SQLi attacks.

7. Use encryption algorithms

Use of encryption techniques to store password and other will prevent direct injection of SQLi.

8. Watch GET parameters or inputs through URL

In cases when GET method is used for form submission then monitoring user input supplied through URL is equally important and essential because this is another entrance for SQLi attack. The parameter values should also be sanitized before being passed to any SQL query to prevent any SQLi attack.e.g. -

www.somedomain.com/test.php?userid=15and 1=convert(int,system_user)

This will result in an error revealing database name if there is no SQLi attack protection.

9. Use of POST method instead of GET method for form submission

POST eliminates the chances of SQLi through URL.

10. Restricting application user scope

This involves tracking user IP and only allowing machines with authorized IP to access the system.

VI. CONCLUSION

The motive of this paper was creating awareness among users about SQLi attacks and about the security measures that can be taken to protect from this attack. This paper is beneficial to Software Developers fraternity as they can take measures to protect the system from this attack and create an unassailable system. It suggests user to use less number of text boxes or text areas and GET method making the application more secure and in case if they are must then also it is suggested that most of the attacks can be prevented by sanitizing input from text box, text area and URL bar.

VII. FUTURE WORK

In addition to the techniques proposed in this paper more techniques/tools for SQLi prevention can be explored or created.

Volume : 5 | Issue : 5 | May 2015 | ISSN - 2249-555X

REFERENCE

1] Acunetix, Report on "SQL injection used in largest data security breach in US history" http://www.acunetix.com/blog/news/sql-injection-used-in-largest-data-security-breach-in-u-s-history/ | [2] M. Martin et al (2005) "Finding Application Errors and Security Flaws Using PQL: A Program Query Language" in ACM SIGPLAN Notices vol. 40 issue 10 pp. 365-383. | [3] Atefeh Tajpour et al (2012) "Web Application Security by SQL Injection Detection Tools" in International Journal of Computer Science Issues (IJCSI) vol. 9 issue 2 p. 3. | [4] Sonam Panda et al (2013) "Protection of Web Application against SQL Injection Attacks" in International Journal of Modern Engineering Research (IJMER) vol. 3 issue 1 pp-166-168. | [5] A. Sravanthi et al (2012) "Detecting SQL Injections From Web Applications" IJESAT vol. 2 issue 3 pp. 664-671. | [6] Priyanka, Vijay Kumar Bohat (2013) "Detection of SQL Injection Attack and Various Prevention Strategies", International Journal of Engineering and Advanced Technology (IJEAT) vol. 2 issue 4 pp. 457-460. | [7] Johannes B. Ullrich Jason Lam (2008) "Defacing websites via SQL injection" in Journal of Network Security vol. 2008 issue 1 pp. 9-10. | [8] Online SQL syntax checker. | http://www.wangz.net/gsqlparser/sqlpp/sqlformat.htm | [9] V.Shanmughaneethi S.Swamynathan "Detection of SQL Injection Attack in Web Applications using Web Services" in IOSR Journal of Computer Engineering (IOSRJCE) volume 1 issue 5 pp. 13-20. | [10] Martin Bravenboer et al (2010) "Preventing injection attacks with syntax embeddings" in Science of Computer Programming volume 75 issue 7 pp. 473-495. | [11] Sayyed Mohammad et al (2013) " Study of SQL Injection Attacks and Countermeasures" in International Journal of Computer and Communication Engineering vol. 2 issue 5 pp. 539-542. | [12] Dr.Manju Kaushik Gazal Ojha (2014) "SQL Injection Attack Detection and Prevention Methods - A Critical Review" in International Journal of Innovative Research in Science, Engineering and Technology vol. 3 issue 4 pp. 11370-11377. | [13] Shaimaa Ezzat Salama et al (2012) "Web Anomaly Misuse Intrusion Detection Framework for SQL Injection Detection" in (IJACSA) International Journal of Advanced Computer Science and Applications vol. 3 issue 3 pp. 124-129. | [14] Mayank Namdev et al (2012) "A Novel Approach for SQL Injection Prevention Using Hashing & Encryption (SQL-ENCP)" International Journal of Computer Science and Information Technologies (IJCSIT) vol. 3 (5) pp. 4981-4987. | [15] Varun Tiwari et al (2012) "A Study of SQL Injection In Banking Transaction" in International Journal of Engineering Inventions vol. 1 issue 8 pp. 70-75. | [16] Y. Huang et al (2005) "A Testing Framework for Web Application Security Assessment" in Journal of Computer Networks vol. 48 issue 5 pp. 739-761. |