ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

Technique against SQL Injection Attack

Khushboo Choubey [1], Prof. Priyanka Jain [2]

M Tech Student, Department of Computer Engineering, GNCSGI, Jabalpur, India[1]

Associate Professor, Department of Computer Engineering, GNCSGI, Jabalpur, India[2]

ABSTRACT: There are multiple web servers in the world and different web sites are using these web servers World Wide Web to be accessed. These websites are vulnerable to attacks that are usually of input validation type. Because of these types of attacks websites can be easily hacked and confidential data can be leaked by any anonymous user. This is a real dangerous situation in the open market.

KEYWORDS: SQL Injection, Techniques, SQLIA, String limit, Privilege.

I. INTRODUTION

A study conducted by White Hat in relation to web security shows 14-15% attacks on the web application by SQL Injection [8]. The day by day increase in the attacks on web applications, has led to the conclusion that we should be aware about the existing attacks, because vulnerabilities such as phishing, social engineering attack, denial of service attacks are common nowadays. The most basic Social Engineering attacks are Phishing and Email spamming.

The most emerging phishing attack is Tab Nabbing, which is even tricky on the tech confident online user [9]. On a survey on web security conducted by us where a total of about 100 students participated, it was really surprising to acknowledge that almost 80% were not able to trace phishing attack and roughly 70% didn't knew Email spam. Hence a basic awareness on web security is a must for everyone, because confidential transactions are being performed every now and then by almost all.

II. RELATED WORK

Of the security incidents that occur on a network, the vast majority (up to 85 percent by many estimates) come from inside the network. These attacks may consist of otherwise authorized users who are disgruntled employees. The remainder comes from the outside, in the form of denial of service attacks or attempts to penetrate a network infrastructure. Intrusion detection systems remain the only proactive means of detecting and responding to threats that stem from both inside and outside a corporate network.

SQL Injection is a critical web security vulnerability. SQLIA is a type of code-injection attack. It is caused mainly due to improper validation of user input.

The main intent to use SQL injection attack include illegal access to a database, extracting information from the database, modifying the existing database, escalation of privileges of the user or to malfunction an application. Ultimately SQLIA involves unauthorized access to a database exploiting the vulnerable parameters of a web application

In the past, many works pertaining to have been proposed and implemented by researchers.

- William G.J. Halfond in 2008 proposed a secure web applications from SQL injection and cross site scripting attacks. In their thesis they focuses on dynamic analysis.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

- Yao Wen Huang et al (2003) proposed a techniques for web security assessment mechanisms in order to identify poor coding practices that render web applications vulnerable to attacks such as SQL injection and cross site scripting attacks. They uses a software testing techniques including dynamic analysis, black box testing, fault injection, and behavior monitoring for testing the security in web applications.

- Balzarotti et al in 2008 proposed a novel approach for the analysis of the sanitization process. They work in the field of static and dynamic techniques by developing a modal combining both static and dynamic analysis techniques to identify faulty sanitization procedures that can be bypassed by an attacker.

- Raymond Wu and Masayuki Hisada (2010) explained their work by using macro and micro views. The macro view was based on syntax structure and identification, while micro view oversees metadata messaging and parser automation.

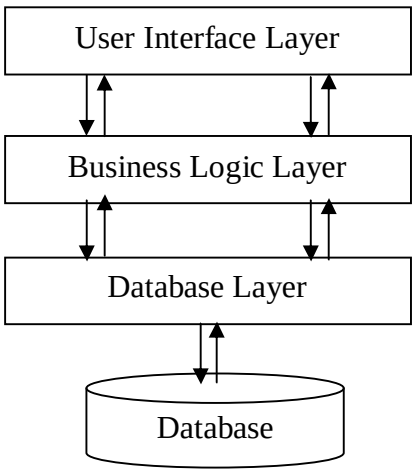
- Moreover, Inyong Lee et al in 2010 proposed a very simple and effective detection method for SQL injection attacks. Their method removes the value of an SQL query attribute of web pages when parameters are submitted and they have compared it with a predetermined one. Their method uses combined static and dynamic analysis.

III. SCOPE

- The system frames certain rules based upon the input given by the user. It then allows traffic inwards or outwards based upon the rules.
- The system also detects certain well-known attacks and gives warnings to the user.

In order to locate the hotspots where SQLIA vulnerability occurs, we first discuss about the 3-tier logical view architecture of web applications. A. 3-tier Architecture of web application

1. User interface tier: This layer forms the front end of web application. It interacts with the other layers based on the inputs provided by the user.
2. Business logic tier: The user request and its processing are done here. It involves the server side programming logic. Forms the intermediate layer between the user interface tier and the database tier.
3. Database tier: It involves the database server. It is useful in storage and retrieval of data.



Fig(1.1) Web 3-Tier Architecture

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

IV. BASIC PRINCIPLE IN SQL INJECTION

SQL injection attack is a web security attack by using SQL statements exploiting the poorly designed input elements of a web form. This compromises the confidentiality and integrity of user's sensitive data. SQLIA takes place between the user interface layer and the business logic layer. To understand the essence of SQL injection let us see the following example.

SELECT * FROM tablename WHERE user='' and password =' ';

A sample SQL statement containing two input parameters is considered .Instead of typing the actual username and password, if a hacker tries to gain access to the application by inputting SQL statements, it is said to be a SQLIA attempt. For example if the hacker inputs, ' OR '1' = '1' --, the statement becomes,

SELECT * FROM tablename WHERE user='' OR '1' = '1' -- and password=' ';

Therefore, if the inputs by the user were not properly sanitized, it might lead to a critical web security attack. This describes the basic principle involved in SQL injection.

V. OUR OBJECTIVE

Using client side script validation such as JavaScript, a lot of SQL injection attacks can be prevented in Web application. Though this approach does not solve all the attack types it is necessary to provide the basic security to prevent illegal attacks. The sequence of steps that increase the level of security in case of vulnerabilities is depicted in the activity diagram.

The advantage of client side validation is that it reduces CPU cycles since it avoids a number of round trips to the server. Some of the steps involved in client side validation include limiting the input size, restricting the use of special characters etc. But limiting the size of the input and restricting the use of special characters cannot be imposed on users in all applications. Also the protection provided by client side scripts can be easily bypassed. The use of this approach can solve attacks using tautology or incorrect queries. It cannot solve the threat posed by blind injection techniques. Hence we prefer server side validation techniques.

VI. PROCESS

The process diagram shown below gives the information about how the implementation takes place when user input SQL statement; the size of the inputted string is checked.

For example, if the SQL statement containing two input parameters. The user is allowed only to enter the actual data. Instead of it if the hacker tries to gain access to the application by inputting special characters in the SQL statements, they got the access privilege. The checking phase considers the special characters similar as the comments and takes no action against those characters.

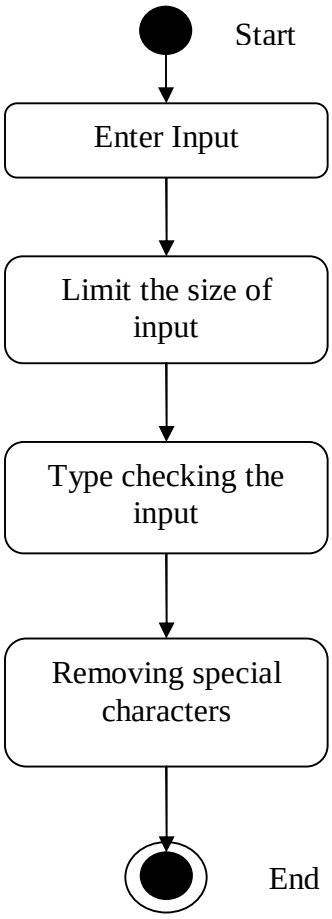
ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019



Fig(1.2) Process diagram of SQL statements

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

VII. IMPLEMENTATION SCREENS

1.1 Input screen

- This screen shows the actual login phase where the user input the necessary details for login.

1.2 Login Menu

- This screen shows how the user can login either for detection or prevention of SQL injection.

1.3 Injecting special character

- In this screen the user input the special character in the username and tries to login as an unauthorized one.

1.4 Attacking with wrong details

- Here the user tries to login by entering the email details with having a special character in the password field.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

1.5 Login Without Prevention

- This window is showing that how an unauthorized user can login without any prevention.

1.7 Searching through stored details

- Here the user tries to fetch the data by searching the relevant details.

1.6 Entering details regarding email

- This screen is showing that when the user is trying to fetch data by entering details of email without any SQL string.

1.8 Flashing comment as an unauthorized one

- Here the user tries to fetch the data without inputting the password details but the admin is peeping the comment to enter the necessary details for login.

- This screen is showing all the details regarding the student which are stored in database .
- The welcome window by which any user grants access to login, search or other function provided.

The above screens were showing that how an authorized user can access the data by entering the necessary details without any special characters. The user is allowed only to input the essential details without any extra SQL string .If he/she either an authorized or unauthorized tries to access the data by inputting wrong details or inputting special characters for example '1'='1'--,the user are not allowed to fetch the data.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

VIII. RESULT

By extended the report we got the resultant that we are able to detect and prevent the attacks done through SQL injection by detecting the special characters edit on the original SQL string and don't giving the access to the unauthorized user.

IX. CONCLUSION

Working on such a kind of tool that can detect SQL Injections is an interesting work. The implementation shows dealing attacks of only one type, the future work shall be based on extension of the same tool so that other types of attacks can also be taken care of.

REFERENCES

[1]. OWASPD-Open Web Application Security Project. "Top ten most critical Web OWASPD-Open Web Application Security Project. "Top ten most critical Web Application Security Risks", https://www.owasp.org/index.php/Top_10_2010-Main,2010.

[2]. W. G. Halfond, J. Viegas, and A. Orso. A Classification of SQLInjection Attacks and Countermeasures. In Proc. of the Intl. Symposium on Secure Software Engineering, Mar. 2006.

[3]. Atefeh Tajpour, Suthaimi, Maslin Masrom. SQL Injection Detection and Prevention Techniques .In Proc. International Journal of Advancements in Computing Technology Volume 3, Number 7, August 2011.

[4]. Ke Wei, M. Muthuprasanna, Suraj Kothari , "Preventing SQL Injection Attacks in Stored Procedures" Proceedings Australian Software Engineering Conference (ASWEC'06 IEEE),2006

[5]. Z. Su and G. Wassermann "The essence of command injection attacks in web applications". In ACM Symposium on Principles of Programming Languages (POPL'2006), January 2006.

[6]. S. W. Boyd and A. D. Keromytis. SQLrand: Preventing SQL Injection Attacks. In Proceedings of the 2nd Applied Cryptography and Network Security Conference, pages 292–302, June 2004.

[7]. William G.J. Halfond and Alessandro Orso," Preventing SQL Injection Attacks Using AMNESIA" ICSE'06, Shanghai, China ACM 06/0005 May 20–28, 2006.

[8]. G. Buehrer, B.W. Weide, P.A.G. Sivilotti, Using Parse Tree Validation to Prevent SQL Injection Attacks, in: 5th International Workshop on Software Engineering and Middleware, Lisbon,Portugal, pp. 106–113,2005.

[9]. R.A. McClure, and I.H. Kruger, "SQL DOM: compile time checking of dynamic SQL statements," Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on, pp. 88- 96, 15-21 May 2005.

[10]. P. Bisht, P. Madhusudan, and V. N. Venkatakrishnan. CANDID: Dynamic Candidate Evaluations for Automatic Prevention of SQL Injection Attacks. ACM Trans. Inf. Syst. Secur., 13(2):1–39, 2010.

[11]. Shaukat Ali, Azhar Rauf, Huma Javed. SQLIPA: An Authentication Mechanism Against SQL Injection. In Proc. European Journal of Scientific Research ISSN 1450-216X Vol.38 No.4 , pp 604-611,2009.