ISSN 2395-1621

Detection of SQL Injection for Security Check-up

Dr. B.L.Gunjal, Sudarshan Bade, Abhinav Tiwari, Sanket Bhandari, Shubham Tiwari

Sudarshanbade9@gmail.com,
abhinavvtiwari97@gmail.com,
sanketbhandari101@gmail.com,
shubhamtiwariz966@gmail.com,

Department Of Information Technology
Amrutvahini College Of Engineering, Sangamner, 422608.

ABSTRACT

As we all know that security is important , as the security come for the mobile data security or any another personal data security, in which unauthorized person can access the databases using SQL injection. The aim of this project to check whether SQL injection is performing on databases or not and provide report. Web sites are dynamic, static, and most of the time a combination of both. Web sites need protection in their database to assure security. An SQL injection attacks interactive web applications that provide database services. These applications take user inputs and use them to create an SQL query at run time. In an SQL injection attack, an attacker might insert a malicious SQL query as input to perform an unauthorized database operation. Using SQL injection attacks, an attacker can retrieve or modify confidential and sensitive information from the database.

ARTICLE INFO

Article History

Received: 22nd June 2020

Received in revised form : 22nd June 2020

Accepted: 24th June 2020

Published online :

24th June 2020

I. INTRODUCTION

SQL injection was an attack in which malicious code was embedded in strings that were later passed to database backend for parsing and execution. The malicious data produced database query results and acquired sensitive information, such as account credentials or internal business data . At present the standard defmition of SQL injection technique was not yet fully established. Chris Anley discussed the various ways in which SQL could be injected into the application and resolved some of the data validation and database lockdown issues. Because most web applications were associated with database backend, there were possibilities of SQL injection attacks on its Through analyzing the principle of SQL injection attacks, prevention method was proposed to solve the double defense through the browser and server ends [4]. Jang exhibited a novel scheme that automatically transformed web applications, rendering them safe against SQL injection attacks.The issue is that some current techniques cannot be applied in practice because they cannot prevent typical attack or have not been implemented yet. So they have limitations that influence their effectiveness and practicability. Some of them need to modify web application code or additional infrastructures. Typical SQL injection attack and prevention technologies are introduced in the paper. The detecting methods not only validate input values but also use type-safe SQL parameters, which is effective against SQL injection vulnerabilities.

II. PROBLEM STATEMENT

A SQL injection attack might cause significant damages to the organization that is being attacked. Some of these damages could result to the loss of an entire table of valuable information or get access to read unauthorized information since the unauthorized user can delete, modify, read, and insert unauthorized records medical records, date of birth, etc.

The main purpose of this project is to detect the sql injection is happening or not and check whether the particular website is vulnerable.

III. LITERATURE SURVEY

Detecting SQL injection attacks (SQLIAs) is becoming increasingly important in database-driven web sites. Until now, most of the studies on SQLIA detection have focused on the structured query language (SQL) structure at the application level. Unfortunately, this approach inevitably fails to detect those attacks that use already stored procedure and data within the database system. In this paper, we propose a framework to detect SQLIAs at database level by using SVM classification and various kernel functions. The key issue of SQLIA detection framework is how to represent the internal query tree collected from database log suitable for SVM classification algorithm in order to acquire good performance in detecting SQLIAs. To solve the issue, we first propose a novel method to convert the query tree into an n-dimensional feature vector by using a multi-dimensional sequence as an intermediate representation. The reason that it is difficult to directly convert the query tree into an n-dimensional feature vector is the complexity and variability of the query tree structure. Second, we propose a method to extract the syntactic features, as well as the semantic features when generating feature vector. Third, we propose a method to transform string feature values into numeric feature values, combining multiple statistical models[2].

Most web applications have critical bugs (faults) affecting their security, which makes them vulnerable to attacks by hackers and organized crime. To prevent these security problems from occurring it is of utmost importance to understand the typical software faults. This paper contributes to this body of knowledge by presenting a field study on two of the most widely spread and critical web application vulnerabilities: SQL Injection and XSS. It analyzes the source code of security patches of widely used Web applications written in weak and strong typed languages. Results show that only a small subset of software fault types, affecting a restricted collection of statements, is related to security. To understand how these vulnerabilities are really exploited by hackers, this paper also presents an analysis of the source code of the scripts used to attack them. The outcomes of this study can be used to train software developers and code inspectors in the detection of such faults and are also the foundation for the research of realistic vulnerability and attack injectors that can be used to assess security mechanisms, such as intrusion detection systems, vulnerability scanners, and static code analyzers[5]

Internet users are increasing day by day. The web services and mobile web applications or desktop web application's demands are also increasing. The chances of a system being hacked are also increasing. All web applications maintain data at the backend database from which results are retrieved. As web applications can be accessed from anywhere all around the world which must be available to all the users of the web application. SQL injection attack is nowadays one of the topmost threats for security of web applications. By using SQL injection attackers can steal confidential information. In this paper, the SQL injection attack detection method by removing the parameter values of the SQL query is discussed and results are presented[7]

SQL injection or SQL insertion attack is a code injection technique that exploits a security vulnerability occurring in the database layer of an application and a service. This is most often found within web pages with dynamic content. This paper proposes a very simple and effective detection method for SQL injection attacks. The method removes the value of an SQL query attribute of web pages when parameters are submitted and then compares it with a predetermined one. This method uses combined static and dynamic analysis. The experiments show that the proposed method is very effective and simple than any other methods[1]

IV. PROPOSED SYSTEM

The aim of this project-based seminar is to study how the SQL Injection Attacks are performed. After studying the working of the SQL Injection Attack we will provide a website to provide security against this attack. To provide a security for the website. To detect and secure the website against the SQL Injection. This website will provide a report to the user and techniques to prevent from SQL Injection.

V. METHODOLOGY

1. Privilege Escalation Attack Detection Module

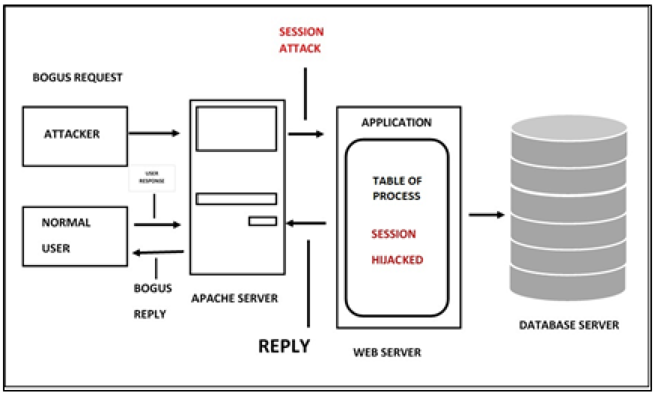
Lets assume that the website serves both regular users and administrators. For a regular user, the web request Ru will trigger the set of SQL queries Qu, for an administrator, the request Ra will trigger the set of admin level queries Qa as shown in Figure. Now suppose that an attacker logs into the web server as a normal user, upgrades his/her privileges, and triggers admin queries so as to obtain an administrator data. This attack can never be detected by either the web server IDS or the database IDS since both Ru and Qi are legitimate requests and queries. Our approach can detect this type of attack since the DB query Qa does not match the request Ru, according to our mapping model.

2.Hijack Future Session Attack Detection

This class of attacks is mainly aimed at the web server side. An Attacker usually takes over the web server and therefore hijacks all subsequent legitimate user sessions to launch attacks which is shown in Figure . For instance, by hijacking user sessions, the attacker can eavesdrop, send spoofed replies, and/or drop user requests. A session-hijacking attack can be further categorized as a Spoofing/Manni-the Middle attack, Denial-of-Service/Packet Drop attack, or a Replay attack.

3.SQL Injection Attack Module

such as SQL injection do not require compromising the web server. Attackers can use existing vulnerabilities in the web server logic to inject the data or string content that contains the exploits and then use the web server to relay these exploits to attack the back-end database which is shown in Figure . Since our approach provides a two-tier detection, even if the exploits are accepted by the web server, the relayed contents to the DB server would not be able to take on the expected structure for the given web server request.



4.Direct DB Attack Module

It is possible for an attacker to bypass the web server or firewalls and connect directly to the database . An attacker could also have already taken over the web server and be submitting such queries from the web server without sending web requests. Without matched web requests for such queries, a web server IDS could detect neither. Furthermore, if these DB queries were within the set of allowed queries, then the database IDS it would not detect it either. However, this type of attack can be caught with our approach since we cannot match any web requests with these queries

VI. REQUIREMENTS

Software Requirements

i. Windows 7 or higher
ii. MySQL
iii. NetBeans

Hardware Requirements

i. Processor – i3
ii. Hard Disk – 5 GB
iii. Memory – 2 GB RAM

VII.PROPOSED OUTCOME

Here we have proposed a more feasible model to secure the website using sql injection techniques. If any case problem such as attack by third party or cyber-attack we can easily recover data by using sql injection techniques

Some Benefits:-

i. It will be cost efficient.
ii. Protection Against SQL Injection Attack
iii. Accessible from anywhere
iv. Runtime Protection
v. User will be able to know about the security of its website
vi. Data Security
vii. Database Security Scanning

VIII. CONCLUSION

In this paper we have proposed Pattern matching technique to detect and prevent SQL Injection attacks on the websites. It successfully detects five types‟ attacks on websites and provides security to the websites. It is a useful application for data driven web applications, hence give security to websites. In future work it can be extended to cover all types‟ threats on websites. Future work should work on efficient methods for detecting the SQL injection attack and methods to prevent it. A less time must be consumed to detect the SQL injections, for this more new and robust methods are to be developed. The impact on the businesses must be understood to reduce the risk of SQL injection attacks. More research is needed for accurately detection of SQL injection attacks.

REFERENCES

[1] Inyong Lee, Soonki Jeong, Sangsoo Yeo, Jongsub Moon,” A novel method for SQL injection attack detection based on removing SQL query attribute values”, Mathematical and Computer Modelling (Elsevier), Volume: 55, Issue: 1-2, PP. 58-68, January 2012.

[2] Mi-Yeon Kim, Dong Hoon Lee,” Data-mining based SQL injection attack detection using internal query Trees,” Expert Systems withApplications (Elsevier), Vol. 41, Issue 11, PP. 5416–5430, September 2014.

[3] Lwin Khin Shar, Hee Beng Kuan Tan,” Defeating SQL Injection,” Computer: the flagship publication of the IEEE Computer Society (IEEE), Volume: 46, Issue: 3, PP. 69 - 77, March 2013.

[4] David Henderson, Michael Lapke and Christopher Garcia,” SQL Injection: A Demonstration and Implications for Accounting Students,” AIS Educator Journal, Vol. 11, Issue 1, PP. 1-8, 2016.

[5] Jose Fonseca, Nuno Seixas, Marco Vieira, and Henrique Madeira,” Analysis of Field Data on Web Security Vulnerabilities,” IEEETransactions on Dependable and Secure Computing, Vol. 11, Issue 2, PP. 89-100, March/April 2014.

[6] Anjali Sardana and et. al., “ Protecting Web Applications from SQL Injection Attacks by using Framework and Database Firewall”, ACM, 2012.

[7] Ashwin Ramesh et. al., “An Authentication Mechanism to Prevent SQL Injection by Syntactic Analysis”, IEEE, 2015.

[8] Indrani Balasundarama and et. al., “An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching”, International Conference on Communication Technology and System Design, 2011.

