International Engineering Journal For Research & Development

Vol.4
Special Issue

E-ISSN NO:-2349-0721

Impact factor : 6.03

SURVEY ON DETECTING SQL INJECTION ATTACK INSIDE THE DBMS USING SEPTIC

. Pooja Gorakh Bagal
Mr. Swapnil Mahadev Machale
Miss. Nisha Prakash Taware
Prof. Nilesh Genaba Gunaware
Department of Computer Engineering
H.S.B.P.V.T'S GOI PARIKRAMA COLLEGE OF ENGINEERING KASHTI, AHMEDNAGAR

ABSTRACT-

Database applications are used to search, sort, calculate, report and share information. Databases can also contain code to perform mathematical and statistical calculations on the data to support queries submitted by users. Databases allow for data to be stored quickly and easily and are used in many aspects of your daily life. SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). The most common cause of database vulnerabilities is a lack of due care at the moment they are deployed. In this paper, we propose SEPTIC(Self-ProtecTIng databases from attaCks), a mechanism for DBMS attack prevention, which can also assist on the identification of the vulnerabilities in the applications. To implement SEPTIC mechanism, we develop an online shopping application. The user may browse through these products as per categories. If the user likes a product, he/she can add it to his/her shopping cart. Once user wishes to checkout, he must register on the site first. Once the user makes a successful transaction admin will get report of his bought products. To develop a secure path for transaction done by the user AES algorithm is an (Advanced Encryption Standard) encryption technique, the transaction and user account details can be made secured.

KEYWORDS- SQL injection, Attack Detection, Attack Prevention, DBMS, Machine Learning.

I. INTRODUCTION

SQL Injection is "a code injection technique that exploits a security vulnerability occurring in the database layer of an application". In other words, its SQL code injected in as user input inside a query. SQL Injections can manipulate data (delete, update, add etc.) and corrupt or delete tables of the database. It is used to attack data-driven application. Lack of input validation is a major vulnerability behind dangerous web application attacks. By taking advantage of this, attacker scan injects their code into applications to perform malicious tasks. In which malicious SQL statements are into an entry field for execution. This is a method to attack web applications that have a data repository. The attacker would send a specially crafted SQL statement that is designed to cause some malicious action. Incorrectly validated or non-validated string literals are concatenated into a dynamic SOL statement and interpreted as code by the SQL engine. We propose modifying – "hacking" – DBMSs to detect and block attacks in runtime without programmer intervention. SElf-ProtecTIng databases from attaCks (SEPTIC). An online shop that allows users to check for different cloths for women's available at the online store and can purchase cloths online. The project consists of list of cloths displayed in various materials and designs. The user may browse through these products as per categories. If the user likes a product, he/she can add it to his/her shopping cart. Once user wishes to checkout, he must register on the site first. He can then login using same id password next time. Now user may pay through a Card. Once the user makes a successful transaction admin will get report of his bought products. Here we use HTML, CSS, JSP, JavaScript to make the entire frontend. The middle tier or code behind model is designed in JAVA and SQL Serves as a backend to store product data thus the online shopping project brings an entire clothing shop online and makes it easy for both buyer and seller to make deals. Admin can add data about their subscribers and it will be viewed by user. The highlighted part here is encryption of card data using AES (Advanced Encryption Standard) technique. The Online Shop secures the card payment and won't let the card data to get hacked. While user doing a card payment, all the card data is encrypted and then stored into database

II. REVIEW OF LITERATURE

1.This paper presents a Some of the most dangerous web attacks, such as Cross-Site Scripting and SQL injection, exploit vulnerabilities in web applications that may accept and process data of uncertain origin without proper validation or filtering, allowing the injection and execution of dynamic or domain-specific language code. Propose a model that highlights the key weaknesses enabling these attacks, and that provides a common perspective for studying the available defences. Availability of source code enhances basic scientific tasks like verification and reproducibility. Secure methods can form the basis for developing methods with more extensive coverage.

2. This paper an attempt has been made to develop an online shop that allows users to check for different cloths for women's available at the online store and can purchase cloths online. The user may browse through these products as per categories. If the user likes a product, he/she can add it to his/her shopping cart. Once user wishes to checkout, he must register on the site first. Once the user makes a successful transaction admin will get report of his bought products. The objective of this project is to develop a secure path for transaction done by the user. Using AES (Advanced Encryption Standard) encryption technique, the transaction and user account details can be made secured. AES encryption is also used to encrypt the user's card and password information while transaction.

3. The web is the firmest and most common medium of communication and business interchange. The attackers make use of these loop holes to gain unauthorized access by performing various illegal activities This may result in theft, leak of personal data or loss of property. The proposed classifier uses combination of Naïve Bayes machine learning algorithm and Role Based Access Control mechanism for detection Our approach detects malicious queries with the help of classifier. The addition of another parameter for RBAC has increased the accuracy of detection.

4. In this paper, we studied the scenario of the different types of attacks with descriptions and examples of how attacks of that type could be performed and their detection & prevention schemes. It also contains strengths and weaknesses of various SQL injection attacks. A proposed a new approach that is completely based on the hash method of using the SQL queries in the web-based environment, which is much secure and provide the prevention from the attackers SQL

5. In this paper, a successful SQLIAs can have serious consequences to the victimized organization that include financial lose, reputation lose, compliance and regulatory breach. To this end, we propose an approach based on negative tainting along with SQL keyword analysis for detecting and preventing SQLIA. We were able to successfully distinguish between legitimate SQL queries and malicious ones that had adopted various evasion methods such as encoding, comments and white space evasion methods as well as logical expressions and string techniques that were not captured by commercially available detection engines.

6. This paper a system is proposed to detect the SQL Injection attacks by using SVM classification and Fisher Score. A lot of research is done to detect and prevent the SQL Injection attacks. To reveal data from database users with valid login id can also enter the malicious queries. Proposed System can also classify the users in to normal users or attackers according to the query submitted by them

7. In this paper, every SQL Query that allows the inputs from the attacker sides can defect our real web application. Uses which are immune to SQL injection attacks validate and sanitize all user input, never use dynamic SQL injection, perform using an account with few privileges, hash or encrypt their secrets, and present error messages that reveal little if no useful information to the hacker. For information on testing your application for injection vulnerabilities, see the sidebar "Injection Testing".

III. SYSTEM OVERVIEW

As we are developing an online application in which the work of both user and admin is mandatory and the essential factor is that as the user select the product it is put into the cart for further process but before that we need to have an account on that web online shopping based application so that in future we can access that account again . Here the work of the user is to shop or can say buy a product and admin work is to display the product in terms of many categories. The purpose to developed is the prevention in two aspects. While accessing the account that is in running time online prevention and second one is when we store our personal details into the database.

IV. SYSTEM ARCHITECTURE

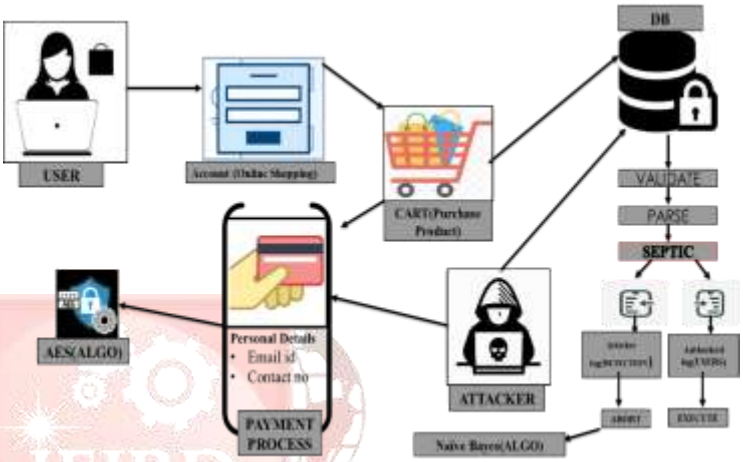


Fig. 01 System architecture

V. ALGORITHMS:

- **Naïve Bayes:**

Detection of SQL Injection attacks using a machine learning algorithm called Naïve Bayes Naïve Bayes is a classification machine learning algorithm that assumes that a particular incident is unrelated to and is independent of other all other incidents. Naïve Bayes classifier is used to classify between malicious and non-malicious SQL queries.

To estimate Navie Bayes the formula is:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \qquad (1)$$

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \qquad (2)$$

- **AES algorithm:**

This algorithm is use for security purpose i.e., to enter the data into encrypted format into a database.

Input:

1. Generate an Initialization Vector (IV)
2. Generating or Loading a Secret Key.

3. Creating the Cipher.
4. Encrypting a String.
5. Decrypting Back to a String.

Output: Inserting the data into the database into an encrypted format

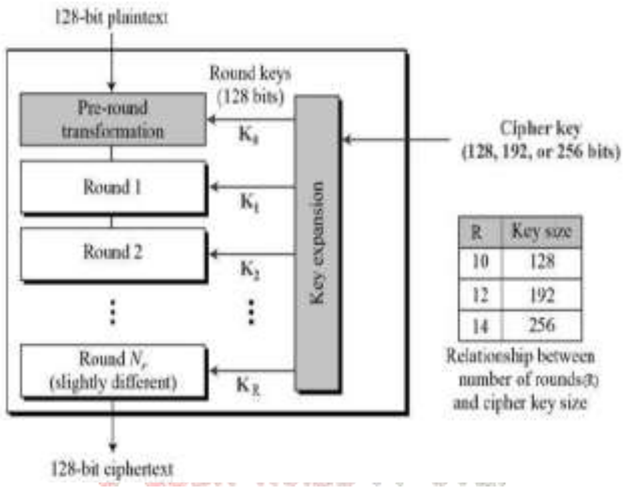
- **Operation of AES:**

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The schematic of AES structure is given in the following illustration −



- **AES Analysis**:

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches.

However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

- **SEPTIC:**

In this we are using SEPTIC methods for prevention of database system from different type of attacks from attacker by following three modes:

- Training mode
- Detection mode
- Prevention mode

VI. CONCLUSION:

According to technology vendor application security the top threat related to databases are SQL injection can be prevented by the new form of mechanism of SEPTIC it also gives an idea of catching attacks inside the DBMS and identifying the vulnerabilities in an application code, when attacks were detected. In future we can use the mechanism to prevent business related confidential data so that no one can try to gain credentials of others and exploit the victim

VII. REFERENCES:

1) Dimitris Mitropoulos, Panos Louridas,† Michalis Polychronakis,‡ and Angelos D. Keromytis, “Defending Against Web Application Attacks: Approaches, Challenges and Implications”, IEEE Transactions on Dependable and Secure Computing Volume: 16 , Issue: 2 , 2019.

2) Karan Ray, Nitish Pol, Suraj Singh Guided by Prof. SUVARNA ARANJO , “Detecting Data Leaks via SQL Injection Prevention on an E-Commerce”, International Journal of Scientific & Engineering Research Volume 9, Issue 3, March-2018.

3) Anamika Joshi, Geetha V “SQL Injection detection using machine learning”, 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT).

4) Mayank Namdev , Fehreen Hasan, Gaurav Shrivastav, “A Novel Approach for SQL Injection Prevention Using Hashing & Encryption (SQL-ENCP)”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3, 2012.

5) Ammar Alazab Al-Balqa, Ansam Khresiat, “New Strategy for Mitigating of SQL Injection Attack”, International Journal of Computer Applications Volume 11, November 2016.

6) Aniruddh Ladole , Mrs. D. A. Phalke,“SQL Injection Attack and User Behavior Detection by Using Query Tree, Fisher Score and SVM Classification”, International Research Journal of Engineering and Technology Volume: 03 June-2016.

7) Sanchit Narang, Shivam Sharma, Rajendra Prasad Mahapatra,“Prevention of SQL injection in E - Commerce”, International Journal of Computational Engineering Research September – 2015
Beach, California, USA