A Learning-based SEPDA model for Detection and Prevention of SQLI Attacks inside the DBMS

SAYALI TELI, PRATIMA MORAJKAR, TEJAS PENDHARKAR, TANMAYEE KULKARNI
BE Students, Department of Information Technology, Marathwada Mitra Mandal's College of Engineering, Pune.

PREETI JOSHI
Assistant Professor, Department of Information Technology, Marathwada Mitra Mandal's College of Engineering, Pune.

***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. The grocery store, bank, video rental store and favourite clothing store all use databases to keep track of customer, inventory, employee and accounting information. 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 now they are deployed. In this paper, we propose SEPDA(Self-Protecting to the data inside DBMS), a mechanism for DBMS attack prevention, which can also assist on the identification of the vulnerabilities in the applications. To implement SEPDA mechanism, we develop an online shopping application that 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. 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.

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 it's 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 to the data inside DBMS (SEPDA). The project comprises of an online shop that allows users to check for the different categories of books available at the online store and can purchase books online. The project consists of list of books displayed in various categories. The user may browse through these products as per categories. If the user likes a book, 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, and 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 book 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 user credentials. The email id and password will be encrypted using AES (Advanced Encryption Standard) algorithm and then will be stored in database.

II. LITERATURE SURVEY

In [1] named "SEPTIC: Detecting Injection Attacks and Vulnerabilities Inside the DBMS" IEEE Transaction paper 2019 proposed by authors named as NunoNeves, Miguel Beatriz present SEPDA (Self protecting databases from attacks) mechanism which is used in 3 different modes such as training, detection and prevention. In training mode the application is trained by firing large number of queries without performing any malicious code. The result of this is stored in

query models. For every query a ID is generated. For SQLI, attacks are identified by comparing queries with query model that is the queries stored in query model. If mismatch is found the query is aborted before execution. SEPDA is implemented by a module inside the DBMS, allowing every query to be checked for attacks. Comparing QS with QM corresponds to first two steps of the detection process. For detection purpose queries are evaluated at the end of the dbms by sematic matching before the query is executed. In prevention mode query processing is aborted, in detection mode query is executed. SETIC also includes incremental method which is used in last two modes that is in detection and prevention mode.

In [2] named "Detecting, Data Leaks via SQL Injection Prevention on an E-Commerce" International Journal of Scientific & Engineering Research,2018 proposed by authors named as Karan Ray, Nitish Pol present 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.

In [3] named "SQL Injection Detection using Machine Learning" International conference, 2014 proposed by authors named as Anamika Joshi, Geetha V present that 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.

In [4]named "New Strategy for Mitigating of SQL Injection Attack" International Journal of Computer Applications,2016 proposed by authors named as Ammar Alazab, Ansam Khresiat present 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.

In [5] named "Preventing SQL Injection Attack Based on Machine Learning" International Conference paper, 2014 proposed by authors named Eun Hong Cheon, Zhongyue Huang from 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.

Table 1:Literature Survey

SR NO.	AUTHOR, TITLE AND JOURNAL NAME	IMPLEMENTATION
1	Nuno Neves , Miguel Beatriz. SEPTIC: Detecting Injection Attack and Vulnerabilities inside the DBMS, IEEE transaction Paper(2019).	SEPTIC works in 3 modes such as training, detection and prevention
2	Karan Ray, Nitish Pol. Detecting Data Leaks vi SQL Injection Prevantion on an E-Commerce, International Journal of Scientific and Engineering Research Volume 9 ,Issue 3,March 2018.	After choosing a product once user make the successful transaction.Using AES encryption technique user account details and transaction can be made secure.
3	Anamika Joshi,Geeta V, SQL Injection Detection using Machine Learning,International Conference(2014)	A method for detection of SQL injection attack based on Naïve Bayes Machine Learning Algorithm combined with Role Based Access control mechanism.
4	Ammar Alazab,Ansam Khresiat,New Strategy For Mitigating of SQL Injection Attack,International Journal of Computer Applications(0975-8887)Volume 154-No.11,(2016)	Evaluations have been performed using three different applications. Further they distinguish between legitimate SQL queries and malicious ones that had adopted various evasion methods.
5	Eun Hong Cheon, Zhongyue Huang, Preventing SQL Injection Attack Based on Machine Learning International Conference(2014).	Ability to stop SOLIA

III. SYSTEM OVERVIEW

We have developed a web application in which work of both user and admin is mandatory. For user, to browse the list of products and to purchase the product it is essential to create a account or register on the application. Once user registers the user credentials are encrypted and stored directly in the database. Admin has to provide the user easy access to online shopping. The overall focus is on securing user details and the system from being attacked.

IV. SYSTEM ARCHITECTURE

Fig 1: System architecture

V. METHODOLOGY

In this section ,we propose a method for detecting and preventing SQL injection .Our approach detects the SQLIAs using a well known method known as SEPTIC(Self-Protecting database from attacks) which is combined with two algorithms Naïve Bayes and AES respectively, to increase the efficiency and security concerns. Machine Learning methods provide highly accurate results on test data. They also give leverage to larger data sets which is crucial factor in our case, because there are many different kinds of attack for which a particular pattern cannot be dug. The whole process from encryption of the user credentials to the prevention of attacks is termed as SEPDA(Self-Protecting to the data inside the DBMS).SEPDA works as following

1. Learning Method

Training mode:-

In this mode database is fired with multiple numbers of queries which are not attacks and is stored in QM. Each query has a associated QID.

Whenever a query is fired the query is parsed, validated .In normal operation, SEPDA generates a QS and a ID for incoming request. For attack detection the QS is compared with the QM that was previously learned for that ID AND then looking for dissimilarities between them. A SQLI attack is found if there is no match.

Notation:

QS:-Query Structure

QM:-Query Model

QID:-Query Identifier

2. Prevention of Attack

When a new query is validated with the incremental method and if no match is found the attack is aborted, i.e the query is dropped thus preventing the system from attack .Once the attack is detected ,the information as the type of attack and the ID of the attacker is stored in the log file created in the database.

3. Encryption of User Credentials

As soon as the user registers on the application, user credentials such as user name and password are encrypted and stored directly in the database. This algorithm is combined with SEPTIC method for security purpose.AES algorithm uses 126 bit key.

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.

KeyExpansion(byte key[16], word w[44])

```
{
word temp

for(i = 0; i< 4; i + +)

w[i] = (key[4*i], key[4*i + 1], key[4*i + 2], key[4*i + 3]);

for(i = 4; i< 44; i + +)
{
temp = w[i – 1];

if ( i mod 4 = 0)

temp = SubWord(RotWord(temp)) ⊕Rcon[i/4];

w[i] = w[i-4] ⊕ temp
}
}
```

• The key is copied into the first 4 words of the expanded key

• Each subsequent word w[i] depends upon w[i-1] and w[i-4]

• For words whose positions are NOT a multiple of 4 w[i] = w[i-4] ⊕ w[i-1]

•Otherwise w[i] = w[i-4] ⊕SubWord(RotWord(temp)) ⊕Rcon[i/4]

4. Naive 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 to and is independent of other all other incident is unrelated incidents. Naïve Bayes classifier is used to classify between malicious and non-malicious SQL queries.

Naïve Bayes(Test_Data_Dir, Training_Data_Dir)
{
For(each test file in test data directory)
For each class
Map<class, probability>ProbabilityMap;
For each word in test file
Wordprobability=Probability of occurance of that word in the class
ProbabilityMap.put(className,probability*Wordprobability)
Class "TestRecord"
Holds the Test record as an object.
* String RecordId Filename of the Test File
* String fullRecord Test record as a single string.
* ArrayList<String> words words in the test record.
Class "OccurrenceProbabilties"
Used as a cache to store the probabilities of words associated with a particular class.
* String classNameClassname
* Hashmap<String,Double> Probability of the each word
Class "MemoryFile"
Holds the training record as an object.
* String className Class name of the training file
* ArrayList<String> content Words in the class.
}
Flow of the Code:

1. Read each test file,remove stopwords, perform stemming and load in to objects.

2. Read each training file, remove stopwords, perform stemming and load in to objects.

3. For each test file, for each class name, for each word; check if the probability already exist in cache.

4. Else compute the probability of each word and multiply them to get overall probability for the test file.

5. Check which probability has maximum among the classes for the test file which gives the class value of the file.

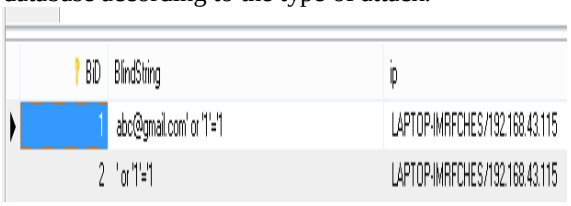
5. Storing details in Log file

A specific log file is generated in the database to keep the track of possible and newly attacks detected. This log file contains information of the type of attack and the ID of the attacker. This also increases the security concern as the information can be viewed at admin side .

VI. RESULT

Experiments are done by a personal computer with a configuration: Intel (R) Core (TM) i3-2120 CPU @ 3.30GHz, 4GB memory, Windows 7, MySQL 5.1 backend database and jdk 1.8. The application is web application used tool for design code in Eclipse and execute on Tomcat server. 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

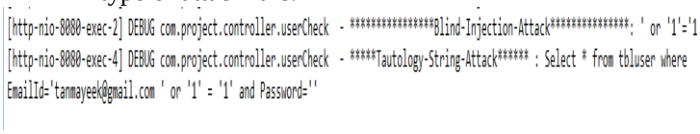
1. Database: Attacks detected and stored into the database according to the type of attack.



BID	BlindString	ip
1	abc@gmail.com' or '1'='1	LAPTOP-IMRFCHES/192.168.43.115
2	' or '1'='1	LAPTOP-IMRFCHES/192.168.43.115

Fig 2:Attacks detected and type of attack

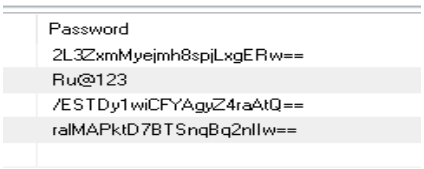
2. Data is also stored into the log file and tells which type of attack it is.



```
[http-nio-8080-exec-2] DEBUG com.project.controller.userCheck - ***************Blind-Injection-Attack***************: ' or '1'='1
[http-nio-8080-exec-4] DEBUG com.project.controller.userCheck - ******Tautology-String-Attack****** : Select * from tbluser where EmailId='tanmayeek@gmail.com ' or '1' = '1' and Password=''
```

Fig 3: Attacks detected and Data is also Stored into the log file

3. Database store users password in encrypted format using AES Algorithm.



Password
2L3ZxmMyejmh8spjLxgERw==
Ru@123
/ESTDy1wiCFYAgyZ4raAtQ==
ralMAPktD7BTSnqBq2nIIw==

Fig 4: Users password in encrypted format is been stored in database for security

	Algorithm	Accuracy
Existing Algorithm	Previous Semantic Analysis	69%

Proposed Algorithm	SEPDA Method	93%

Fig 5: Accuracy associated with algorithm for encrypting data

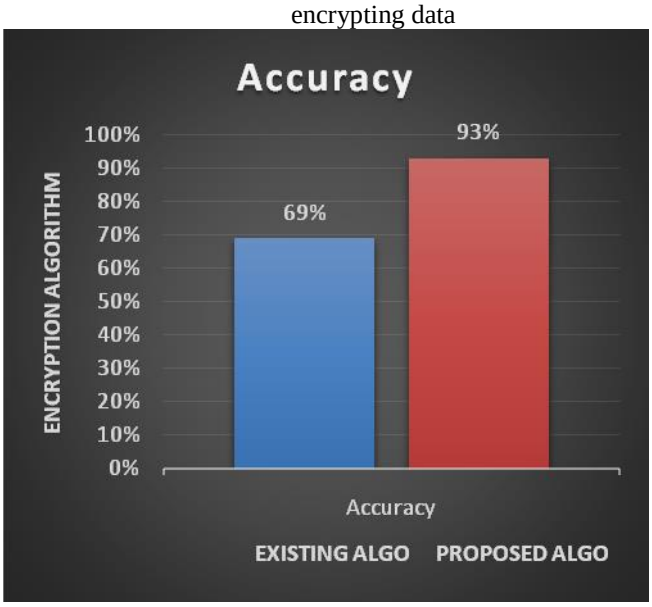


Fig 6:Existing and Proposed Algorithm

VII. CONCLUSION

In this paper ,we have proposed a method (SEPDA) for detection and prevention of SQL Injection attacks inside the DBMS based on a well known SEPTIC method combined with AES and Naive Bayes for extra security concerns. The approach of using machine learning algorithm has increased accuracy of detection. This method also provided a entry in the log file whenever it detects malicious activity(attack). The mechanism was experimented by adding vulnerabilities on purpose in the application code. Experimenting this shows increase in accuracy.

VIII. REFERENCES

[1] NunoNeves, Miguel Beatriz." SEPTIC: Detecting Injection Attack and Vulnerabilities inside the DBMS.
IEEE transaction paper(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] AmmarAlazab Al-Balqa, AnsamKhresiat, "New Strategy for Mitigating of SQL Injection Attack", International Journal of Computer Applications Volume 11, November 2016.

[5] Eun Hong Cheon, Zhongyue Huang, Preventing SQL Injection Attack Based on Machine Learning, International conference(2014)

[6] MayankNamdev , FehreenHasan, 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.

[7] Dimitris Mitropoulos, PanosLouridas,† 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.

[8]S.W.Boyd and AD.Keromytis, "SQLrand: Preventing SQL Injection Attacks," Proc. the 2nd Applied Cryptography and Network Security (ACNS)
Conference, Jun 2004.

[9] M.Howard and D.LeBlanc, "Writing Secure Code for Windows Vidta,1sted, Microsoft Press Redmond,WA,USA,2007.

[10] W.G.Halfond and Aorso, "AMNESIA Analysis and Monitoring for Neutralizing SQL Injection Attacks," Proc. IEEE and ACM International Conference on Automatic Software Engineering (ASE 2005), Long Beach, CA, USA, Nov 2005.