SQL injection detection using REGEX classifier *

B. Kranthikumar and R. Leela Velusamy

Department of Computer Science and Engineering, National Institute of Technology, Tamilnadu India- 620015
er.krantikumar@gmail.com, leela@nitt.edu

Abstract. In digital world of Web Technology, Web applications are used to provide on-line services such as paying bills, reading news, shopping, social networking, banking, etc. These services are growing day by day. As these services are growing more in number, we are facing various sophisticated attacks that target them [4]. These attacks are the most serious threats to web applications. SQL(Structured Query Language) injection has become one of the most common attack that access the data in the web in an unauthorized manner. This paper focuses on detection of SQL injection attack using pattern based classification called REGEX [3] and the outcome is compared with machine learning classifications such as SVM(Support Vector Machine), Gradient boosting Algorithm and Naive bayes classifier . By performing classification on collected synthesized dataset with 20474 queries and it is shown that the REGEX classifier gives 97% accuracy along with 3.98 sec computation time in execution which more efficient than above machine learning techniques.

Keywords: SQL Injection · REGEX patterns · Machine learning · SVM · Gradient boosting Algorithm · Naive bayes classifier.

1 Introduction

The OWASP(Open web application security project) is a worldwide non profit charitable organization that is continuously working on application software security issues and providing information about AppSec to corporations, individuals and organizations etc. to take decisions. The top 10 list consists of the 10 most seen application vulnerabilities. Among those vulnerabilities the top most one is SQL injection followed by broken authentication. SQL injection attack is a type of security vulnerability that target database connected web applications. It generally allows an attacker to view data that they are not normally able to retrieve In this attack, the attacker inserts a malicious SQL query into the web application to manipulate data or even to gain access to the back-end databases. This vulnerability mainly occurs due to weaknesses present in source codes. The other reasons for this vulnerability may be the weakness of the programming language or improper input validation. Most of the works in Detecting SQLIA(SQL

* Supported by organization x.

injection attack) are focused on SQL injection structure at Application Level, but this method fails in detecting such attacks that use stored procedure and also the altered data that present in database system [7]. The SQL injection attack can update data, delete data, insert and execute commands on server which cause to download and install Trojans and other malicious programs. Exporting valuable data such as credit card details, email, and passwords to the attackers remote server Getting user login details etc alter or delete data stored in the back-end databases, read sensitive information from the database and perform an administrative operation on the database for example shutdown of the database management system or some times an attacker attacks to compromise the underlying back-end infrastructure or server, or also can perform a denial-of-service attack. A SQL injection attack is created by addition of a SQL inquiry with the input information from the client to the application. Many researches are working to find the vulnerabilities in web applications which are prone to the SQL injection attack and trying to provide solution to prevent them from happening. In this paper 3 machine learning techniques namely SVM, Gradient Boosting algorithm and Naive Based classifier and another REGEX classifier is implemented for detecting SQL injection attack. The performance of the techniques are tested and results compared with respect to Accuracy, Sensitivity, Specificity, Precision, and Computation time using the synthesized Dataset obtained from GITHUB.

1.1 Insecure Coding

SQL Queries are used to access database servers like MySQL, SQL server, and Oracle servers. Web programming languages provides facilities to construct and execute the SQL queries. During the process of development there is chance that the developers often misuse these methods due to lack of training and experiences [9].

1.2 Query types

Execution of database application occurs by SQL queries. These Queries are of two types
1) Original Queries and
2) Suspicious Queries.
User login_id and passwd is supplied using web application interface as shown in Fig. 1. For example a user enters query to request for his grade sheet from a static service such as *http://misnew.nitt.edu*, this web page can be retrieved from the institute web server.(shown in Fig. 1, lines 1,2,3,4). This information can be accessed by application page that stored in the application server followed by Database using credential parameters such as Username and Password which is further executed through available modules or codes of databases.If valid input with correct syntax, secure coding and guideline in design of web is not followed, malicious code could be injected in database [8]. E.g., Select * from ADMN where loginid = OR 1 = 1; (Suspicious Query). Several techniques

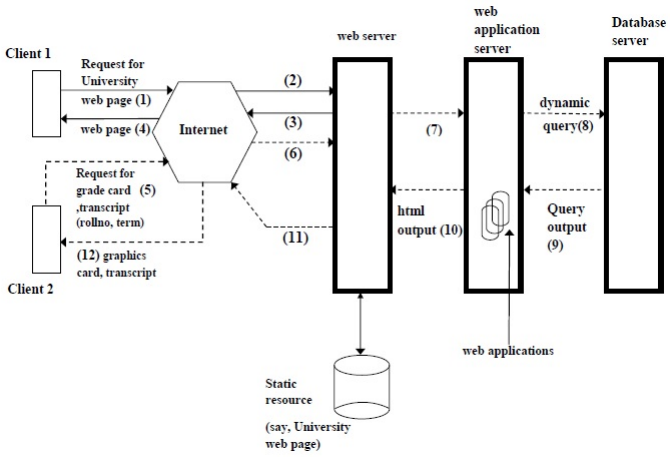

Fig. 1. Flow of SQL queries and server Response

has been already defined for protecting the system from SQL-Injection attack such as Intrusion detection System Model using Document Object Model-tree comparison of different SQL queries, tools such as Swaddler for detecting SQL-Injection attack [2]. This paper describes the detection methods, implemented them and compared the outputs observed. Synthesized dataset from GitHub was used for testing the implemented methods. This data set contains SQL injection query(suspicious) text and plain query(Original) text as shown in table 1.

Table 1. Query Types

Benign Queries	3694
Suspicious Queries	16780

1.3 Support Vector Machine Learning

Applying Machine learning(ML) technique to classify SQL injection attack is a regular practice and one of the popular ML algorithm used for classification is Support vector machine(SVM). SVM is linear classifier and every SQL query in Dataset are labeled with '1' or '0' for implementation.

1.4 Gradient Boosting Algorithm

Gradient Boosting Algorithm is a ensemble model called Boosting model which is a combination of many weak models to form a strong model. This is another

ML algorithm used for classification of SQL injection attack. This algorithm is a special type of algorithm to reduce the error sequentially.

1.5 Naive Bayes Classifier

This classifier is one of the popular method to classify the text category. In machine learning technique, this classifier comes under a family of simple 'probabilistic classifier'. This classifier applies Bayes' theorem with strong independence assumptions in-between the query features. The classification method considers whether the specified feature in the class is present or not. By considering e.g features of an apple fruit are like red color, round shape and diameter about 3 inches. All these features are dependent on each other or upon the existence of the other features. These features independently considered to the probability that given fruit is an apple and that is why it is known as Naive. This classification model is easy to implement and works effectively with very big data sets.

1.6 Regular Expression

REGEX is a classifier which uses regular expressions as filter to classify the applied query SQL injection or genuine query. The following 11 regular expressions are used in this classifier to classify the queries as SQL injections or not.

REGEX patterns are

1. ;+\"+\"","One or more ; and at least one \" or \'"
2. (–.*)$", "– at end of SQL
3. (/*).*(*/), "Found /* and */"
4. "\"{2,}+", "Two or more \""
5. "\'{2,}+", "Two or more \'"
6. "\\d=\\d", "any digit=any digit"
7. "(\\s\\s)+","two or more white spaces in a row"
8. "(#.*)$", "# at end of SQL"
9. "%{2,}+", "Two or more \\% signs"
10. "([;\'\"\\=]+.*(admin.*)—((admin.*).*[;\'\"\\=]+)", "admin (and variations like administrator) and one of [; ' \" =] before or after admin");
11. "%+[0-7]+[0-9—A-F]+","ASCII Hex"

2 Literature Survey

Most of the existing SQL injection techniques such as information flow analysis, filtering, defensive coding and penetration testing can detect and prevent only a subset of the vulnerabilities that lead to SQL injection attacks. In this section,

we list some relevant techniques for SQL injection and discuss their limitations. Debasish and Sharma et al [6]. had proposed two classifications namely Edit distance algorithm and Binary distance algorithm. The author compared the outcome of his proposal with support vector machine algorithm, Naive Bayes classifier and Parse Tree Based approach methods and justified his proposal was better. The limitation with this method was that it can handle only two types of SQL injection attacks where as the literature claims that there are 6 types of attacks [6] as listed in table 2 Ahuja et al. [1] had recommended the implementation

Table 2. Types of SQLi Attacks

Type	**Description**	**Technique**
Tautology	The condition of the statement gives always TRUE result	SELECT * FROM tableName WHERE user_login= or 1=1–
Union	Combining the results of two or more statements	SELECT * FROM tableName WHERE user_login= UNION SELECT * FROM tableName WHERE No=12345 – AND passwd = AND pin=
Logical or illegally incorrect	Inject parameters which creates syntax, type conversion, or logical error	SELECT * FROM tableName WHERE user_login= 'kranthi"' AND passwd =
Piggy Backed	Update original query by inserting additional queries to the original statement	SELECT * FROM tableName WHERE user_login=kranthi AND passwd=; drop tableName user – AND pin=221
Stored Procedure	Run built-in functions using malicious SQL codes Modify the injection statement by alternating encoding to escape from detection	SELECT * FROM tableName WHERE user_login= 'kranthi' AND passwd ='kumar'; SHUTDOWN;–;
Alternate encoding	Alter or modify SQL injection query statement to escape from detection	SELECT * FROM tableName WHERE user_login= 'kranthi';exec(char(0x59842 352646f776e)) AND passwd ='kumar' AND pin =; SHUTDOWN;–;

of three different approaches to prevent the SQL injection attack namely Query Rewriting Approach, Encoding-based Approach and Assertion-based approach. Rewriting based approach is the easiest way to check whether SQL query is normal or an attack. But if the Query is longer one then computational overhead problem arises. The second technique is Encoding based approach technique. In this approach only UID and password was encoded and decoded. E.g.,

```
SELECT uid
FROM loginfo
WHERE
login = '0101010101010101010100011001010001 '
AND
pass = '0101010101010101010100011001010001'.
```

The limitation in this approach is, it has the code conversion overhead and it is time consuming method. The third technique is Assertion-based Approach. In this approach a piece of code (assertion) is added in the web application in order to verify that no SQLIA happens. The drawback of this approach is practically it is very difficult to implement and in case of large database it introduces computational overhead.

John et al. [5] had proposed an algorithm to prevent the SQL injection attack at the login phase by combining both the code conversion and the parse tree validation methods. The author studied varies techniques for the SQL attacks and the prevention techniques. The parse method validates the user input for its vulnerability, and the code conversion takes place if there was a chance of vulnerability. The integrated algorithm is capable of preventing only text field SQL attack. Thus future work was to develop a better algorithm that can prevent the SQL attack through other methods.
Tang et al.[10] proposed the deep learning based method for detecting SQL injection attack. Feature values based on the user behaviour are extracted from their http traffic with the help of deep learning technique and the values are fed into the Deep Neural Network-Multi Layer Perception(MLP) and the Long Short Term Memory(LSTM) networks. The MLP network and LSTM network accomplished the accuracy of 99% and 95% respectively. The work was based on the training results only.

3 Objectives

The main objective of this methodology is

- To improve the efficiency of the system
- To improve the accuracy (True Positive and True Negative)

4 Methodology for Proposed Solution

To overcome the above limitations to detect the SQL injection attack this paper presents a simple and efficient method for SQL injection attack detection. Whenever a SQL query is generated from client side the proposed model classifies whether that query is injected one or normal query by using its pattern

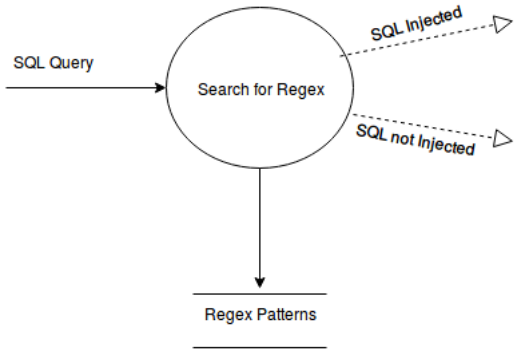

Fig. 2. REGEX working model

checking method as shown in Fig.2. The proposed model and the Machine Learning models were implemented using Python, and tested using the synthesized dataset obtained from GITHUB.

5 Result Analysis

5.1 Training results

The machine learning algorithms SVM, Naive Bayes and Gradient Boosting algorithms are trained with the training set (80%) and the effect of the machine learning algorithms are detected with the test set(20%). The observed outputs are tabulated and presented in Table 3. The formulas for detection of accuracy rate , Sensitivity, Specificity and Precision are given below.

Table 3. Performance Evaluation Matrix

	TN	FP	FN	TP	Total
Naive	761	0	4308	9680	14749
SVM	38	723	26	13962	14749
Gradient	468	293	552	13436	14749
REGEX	0	0	514	19960	20474

The REGEX is a pattern based classifier so there is no need of training with 20% data set. All 20474 queries are set for input.

6 Formulas

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

7 Result analysis Graphs

The following Figure.3 shows the comparison graphs of Computation Time, Accuracy, Precision and Sensitivity that are derived from the implementation.

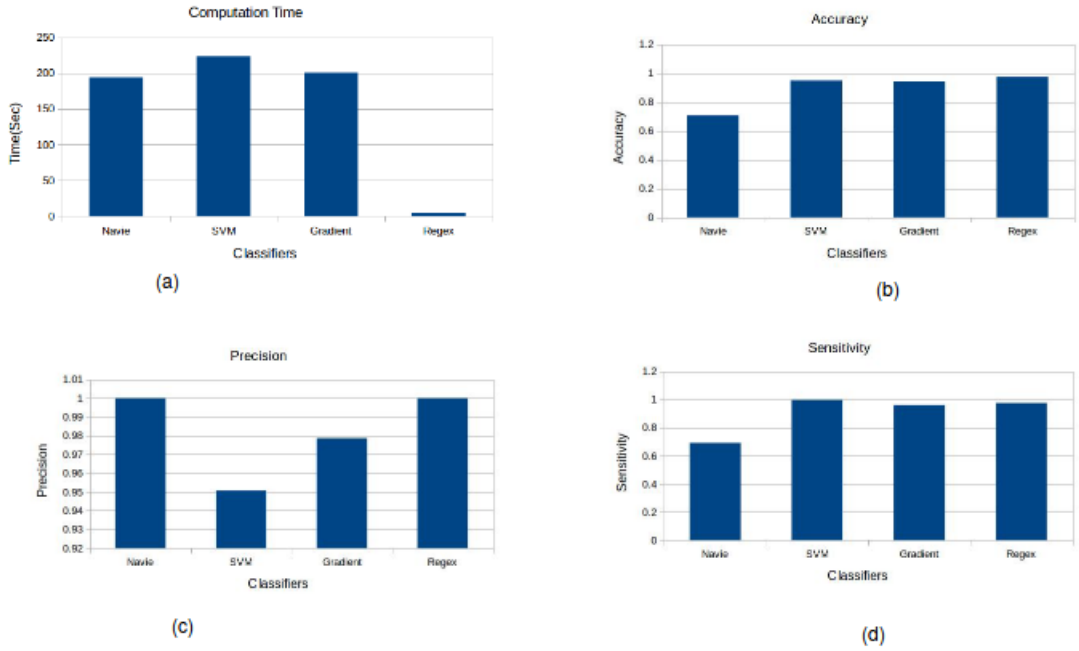

Fig. 3. a) Computation time, b) Accuracy, c) Precision, d) Sensitivity

From the above graph, we observed that REGEX performance is better than other classifiers with respect to comparison metrics as following-
a) Computation time is very less, hence it is faster.
b) Accuracy is more so it is less erroneous
c) The higher precision rate indicates positive prediction rate
d) Sensitivity is high, so the ability to detect injected queries is more.

8 Conclusion

This paper is a comparative study of SQL injection detection. Three machine learning techniques namely SVM, Naive Bayes, Gradient boosting Algorithm and another REGEX classifier was implemented and tested. The techniques were tested using synthesized dataset, which includes 20474 SQL injection queries. Experiments showed that REGEX method took 3.98 sec computational time to detect SQL injection attack using pattern scanning method. REGEX gives a very good performance with 97% accuracy and also 100% precision.

References

1. Bharat Kumar Ahuja, Angshuman Jana, Ankit Swarnkar, and Raju Halder. On preventing sql injection attacks. In *Advanced Computing and Systems for Security*,

pages 49–64. Springer, 2016.
2. Marco Cova, Davide Balzarotti, Viktoria Felmetsger, and Giovanni Vigna. Swaddler: An approach for the anomaly-based detection of state violations in web applications. In *International Workshop on Recent Advances in Intrusion Detection*, pages 63–86. Springer, 2007.
3. Dipesh Gautam, Zachari Swiecki, David W Shaffer, Arthur C Graesser, and Vasile Rus. Modeling classifiers for virtual internships without participant data. In *Proceedings of the 10th International Conference on Educational Data Mining*, 2017.
4. William GJ Halfond and Alessandro Orso. Amnesia: analysis and monitoring for neutralizing sql-injection attacks. In *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering*, pages 174–183, 2005.
5. George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.
6. Da-Yu Kao, Chung-Jui Lai, and Ching-Wei Su. A framework for sql injection investigations: Detection, investigation, and forensics. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2838–2843. IEEE, 2018.
7. Mi-Yeon Kim and Dong Hoon Lee. Data-mining based sql injection attack detection using internal query trees. *Expert Systems with Applications*, 41(11):5416–5430, 2014.
8. Romil Rawat and Shailendra Kumar Shrivastav. Sql injection attack detection using svm. *International Journal of Computer Applications*, 42(13):1–4, 2012.
9. Lwin Khin Shar and Hee Beng Kuan Tan. Defeating sql injection. *Computer*, 46(3):69–77, 2012.
10. Peng Tang, Weidong Qiu, Zheng Huang, Huijuan Lian, and Guozhen Liu. Sql injection behavior mining based deep learning. In *International Conference on Advanced Data Mining and Applications*, pages 445–454. Springer, 2018.