AN INTRUSION DETECTION SYSTEM FOR SQL INJECTION ATTACK USING MACHINE LEARNING METHOD

[1]Priyanka Yadav
M.Tech. Student,
Department of Computer Science & Engg.
Shri Shankaracharya Group of Institution

[2]Dr. Abha Choubey
Associate Professor,
Department of Computer Science & Engg.
Shri Shankaracharya Group of Institution

Abstract*— Most of the administrations in the world uses the web as a medium for their day to day communications and businesses transactions. The strength of all web based applications are Relational Data base which are functioned by Structured Query Language (SQL). As the web becomes more prevalent, web attacks are increasing day by day. SQL Injection Attacks (SQLIAs)-Structured Query Language (SQL) is an interpreted language used in database driven web applications which construct SQL statements that incorporate user-supplied data or text. , if this happened in an precarious manner, then the web application may be susceptible to SQL Injection Attack i.e. If user abounding data is not appropriately authenticated then user can amend or expertise a malevolent SQL statements and can execute haphazard code on the machine or can alter the contents of database. In this paper we will use machine learning algorithm (classification) for detection of SQL injection over web application also measured the performance of proposed SQL injection classifier output with SVM classifier.

***Keywords*— IDS;SQL,XSS;SQL;ML**

I. INTRODUCTION

Internet users are increasing day by day, as stated by International Telecommunication Union (ITU) [ITU, 2015] the number of Internet users founds more than 40% of the world population

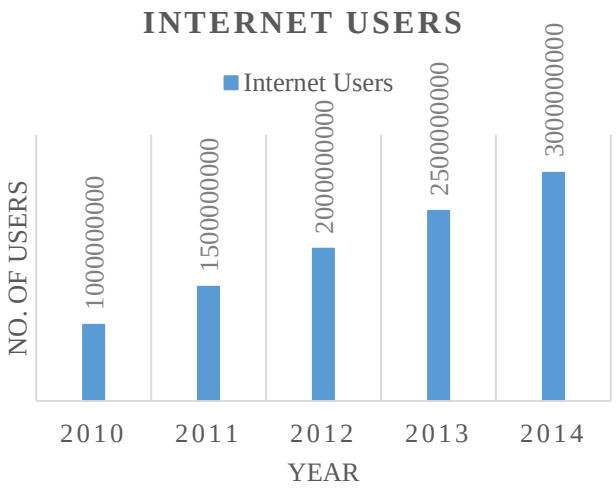


Fig.-1. No. of Inter net user in world

As we can see from fig.1 number of internet users increases which concludes that there is huge amount of data floats over the internet which cause security issue. The fact of having billions of internet users, being gain access from diverse platforms, handling huge amounts of money and handling secretive information make them appealing for cybercrimes.

SQL Injection Attacks Structured Query Language (SQL) is an interpreted language used in database driven web applications which build SQL statements that include user supplied data or text. If this happened in a hazardous manner, then the web application become susceptible to SQL Injection Attack i.e. when user supplied text is not correctly validated then user can alter malevolent SQL statements and can execute indiscriminate code on the target machine or alter the contents of database.

In order for an SQL Injection attack to take place, the vulnerable website needs to directly include user input within an SQL statement. An attacker can then insert a payload that will be included as part of the SQL query and run against the database server.

Types of SQL injection

- Tautology-based SQL Injection
- Piggy-backed Queries / Statement Injection
- Union Query
- Illegal/Logically Incorrect Queries
- Inference
- Stored Procedure Injection

In a tautology-based attack, the code is injected using the conditional OR operator such that the query always evaluates to TRUE. Tautology-based SQL injection attacks are usually bypass user authentication and extract data by inserting a tautology in the WHERE clause of a SQL query. The query transform the original condition into a tautology, causes all the rows in the database table are open to an unauthorized user. A typical SQL tautology has the form "or <comparison expression>", where the comparison expression uses one or more relational operators to compare operands and generate an always true condition.

Example URL of this type of arrack as:

```
http://www.example.com/product.php?id=10 AND 1=1
```

In Piggy-backed Queries type of attack is different than others because the hacker injects additional queries to the original query, as a result the database receives multiple SQL queries. The first query is valid and executed normally, the subsequent queries are the injected queries, which are executed in addition to the first. Due to misconfiguration, a system is vulnerable to piggy-backed queries and allows multiple statements in one query.

Example URL of this type of arrack as:

```
http://www.mydomain.com/products/products.asp?productid=123 UNION SELECT user-name, password FROM USERS
```

```
http://www.mydomain.com/products/products.asp?productid=123; DROP TABLE Products
```

Certain characters and character sequences such as ; , --, select, insert and xp_ can be used to perform an SQL injection attack. Remove these characters and character sequences from user input which reduces the chance of an injection attack. Scan query string

for undesirable word like "insert", "update", "delete", "drop" etc. check whether it represent a statement or valid user input.

Example URL of this type of arrack as:

> http://www.example.com/product.php?id=10; INSERT INTO users (…)

II. LITERATURE SURVEY

We have gone through several literature some of them are discussed further.

Inyong Lee et. al. [Elsevier 2011] said that 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.

According to Inyong Lee et. al. [Elsevier 2011] protection methods for the SQL injection attack are:

Static analysis analyzes the SQL query sentences of web applications to detect and prevent SQL injection attacks. It also requires rewriting of web applications. The focus of the static analysis method is to validate the user input type in order to reduce the chances of SQL injection attacks rather than detect them. JDBC-Checker uses the Java String Analysis (JSA) library to validate the user input type dynamically and prevent SQL injection attacks. However, if malicious input data has the correct type or syntax, it cannot protect against the SQL injection attack. Also, the JSA library only supports the Java programming language.

Dynamic analysis analyzes the response from a web application after scanning it. A scan means to send every kind of input to the target and receive the response. Unlike static analysis, it can locate vulnerabilities from SQL injection attacks without making any modifications to web applications. Paros, which is an open source program, finds not only SQL injection attacks, but also other vulnerabilities within the web application. Paros is not effective because it uses predetermined attack codes to scan and determines the success or fail with the HTTP response.

The web framework uses a filtering method to remove special characters. Recently, some web frameworks have provided a wider variety of prevention methods than ever before. PHP provides Magic Quotes, which works when any combination of 4 special characters ', ", /, NULL exists in the data field of the POST, GET and COOKIES pages. It automatically adds a '\' in front of the special character to prevent SQL injection attacks. However, Magic quotes only works for the four special characters and therefore, other detouring attacking methods exist. Also, web applications must be rewritten in order to configure the Magic Quotes function.

Indrani Balasundaram et. al. combines static and dynamic analysis in the static analysis stage, the prevention technique represents in three level phases Malicious Text Detector, Field Constraint Validation and Static Query Length Validation. In runtime validation stage, the user input data is validated with all these stages and results the user input as safe or unsafe. This prototype tool that implements our technique for .NET based web applications; Current technique was able to stop all of the 500 attacks that we performed on the considered applications without producing any false positive for the 1500 legitimate accesses to applications. This technique was able to correctly identify all attacks as SQLIA‟s, while allowing all legitimate queries to be performed. This proposed technique is able to stop all the attacks that performed on the considered applications without producing any false positive.

Rung-Ching Chen and Kai-Fan Cheng et. al. [IEEE 2009] proposed an intrusion detection method using an SVM based system on a RST to reduce the number of features from 41 to 29. We also compared the performance of the SVM with that of a full features and Entropy. Our framework RST-SVM method result has a higher accuracy as compared to either full feature or entropy. The experiment demonstrates that RST-SVM yields a better accuracy. In the future, we will increase number of testing data for our system and to find vary of accuracy. We also hope to combine RST method and genetic algorithm to improve the accuracy of IDS.

S. No.	Author/Title/ Publication	Type of Attack	Description	Future Direction
1.	Inyong Lee et. al./A novel method for SQL injection attack detection based on removing SQL query attribute values/Elsevier 2012	SQL injection	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.	Efficiency can be increased using Machine learning Algorithm and can be work upon other attack as XSS.
2.	Indrani Balasundaram et. al./An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching/Elsevier 2012	SQL injection	This security prevents the unauthorized access to your database and also it prevents your data from being altered or deleted by users without the appropriate permissions.	This Technique can be applied over all other web based Attacks
3.	Yuji Kosuga et. al./Sania: Syntactic and Semantic Analysis for Automated Testing against SQL Injection/ IEEE 2007	SQL injection	Sania intercepts the SQL queries between a web application and a database, and automatically generates elaborate attacks according to the syntax and semantics of the potentially vulnerable spots in the SQL queries.	NA
4.	Vipin Das et. al./Network Intrusion Detection System Based	Network IDS	This intrusion detection method using an SVM based system on a	Machine Learning (SVM)

	On Machine Learning Algorithms/IJCSIT 2010		RST to reduce the number of features from 41 to 29. Author also compared the performance of RST with PCA.	can improve performance
5.	Muhammad Saidu Aliero et. al./Classification of Sql Injection Detection And Prevention Measure/ IOSRJEN 2016	SQL injection	The goal of this paper is to provide programmers with common issues that need to be considered before choosing a particular technique and to raise awareness of issues related to such techniques as many of those techniques were not meant for the purpose of protection of SQLIA. In addition, author hope to provide researchers by shedding light on how to develop good SQLI (SQL Injection) protection tools as most of the SQLI protection tools were developed using combination a of two or more defensive coding techniques. Lastly we provide recommendations on to avoid such issues.	Machine Learning Approach simulates a high number of attack patterns in training data including blind SQL injection attack which is very difficult to address
6.	Mahdi Zamani and Mahnush Movahedi/Machine Learning Techniques for Intrusion Detection/arXiv 2015	Single or a network of computers for malicious activities	Characteristics of ML techniques makes it possible to design IDS that have high detection rates and low false positive rates	Noisy Training Data leads to decrease the performance of ML based IDS
		(attacks)	while the system quickly adapts itself to changing malicious behaviors. Author divided algorithms into two types of ML-based schemes: Artificial Intelligence (AI) and Computational Intelligence (CI).	
7.	Rung-Ching Chen and Kai-Fan Cheng/Using Rough Set and Support Vector Machine for Network Intrusion Detection System/IIDS 2009	Network IDS	This paper, author use RST (Rough Set Theory) and SVM (Support Vector Machine) to detect intrusions. First, RST is used to preprocess the data and reduce the dimensions. Next, the features selected by RST will be sent to SVM model to learn and test respectively. The method is effective to decrease the space density of data.	Author said in the future, we will increase number of testing data for our system and to find vary of accuracy. We also hope to combine RST method and genetic algorithm to improve the accuracy of IDS.

III. PROBLEM IDENTIFICATION

After going through several literature we have identified some problem which are as:

- Developer learning is required.
- Source code adjustment is needed.
- Earlier system uses static and dynamic analysis in which there is a chance to get false positives and false negatives in some situations.
- When the string analysis results in a SQL query model that is overly conservative and includes spurious queries (i.e. queries that could not be generated by the application) that happen to match an attack.
- When a legitimate query happens to have the same" SQL structure" of an attack. For example, if a developer adds conditions to a query from within a loop, an attacker who inserts an additional condition of the same type would generate a query that does not violate the SQL-query model.

- In some cases, as the analysis cannot distinguish a variable or a hard-coded SQL token, it raises false positives for a string model that is precise enough. In particular, if the hard-coded string is used in the application to construct a SQL token, the technique will generate an incomplete SQL-query model.

IV. PROPOSED METHDOLOGY

Inyong Lee et. al. discussed In their conclusion section that by applying machine learning algorithm we can increase the efficiency of detection system.

In this project we have used machine learning for SQL injection string classification.

Proposed Algorithm

Step-1. Input URL as string.Step-2. Check for Valid Input URL.
Step-3. URL string is broken into tokens.
Step-4. Check if the site contains any error or missing then site is vulnerable.
Step-5. Check if token contains vulnerable SQL strings e.g. union, drop, mysql_fetch etc., 1==1 etc..
Step-6. Classify the URL string as vulnerable or non-vulnerable.
Step-7. Add malicious or non-malicious URL string.

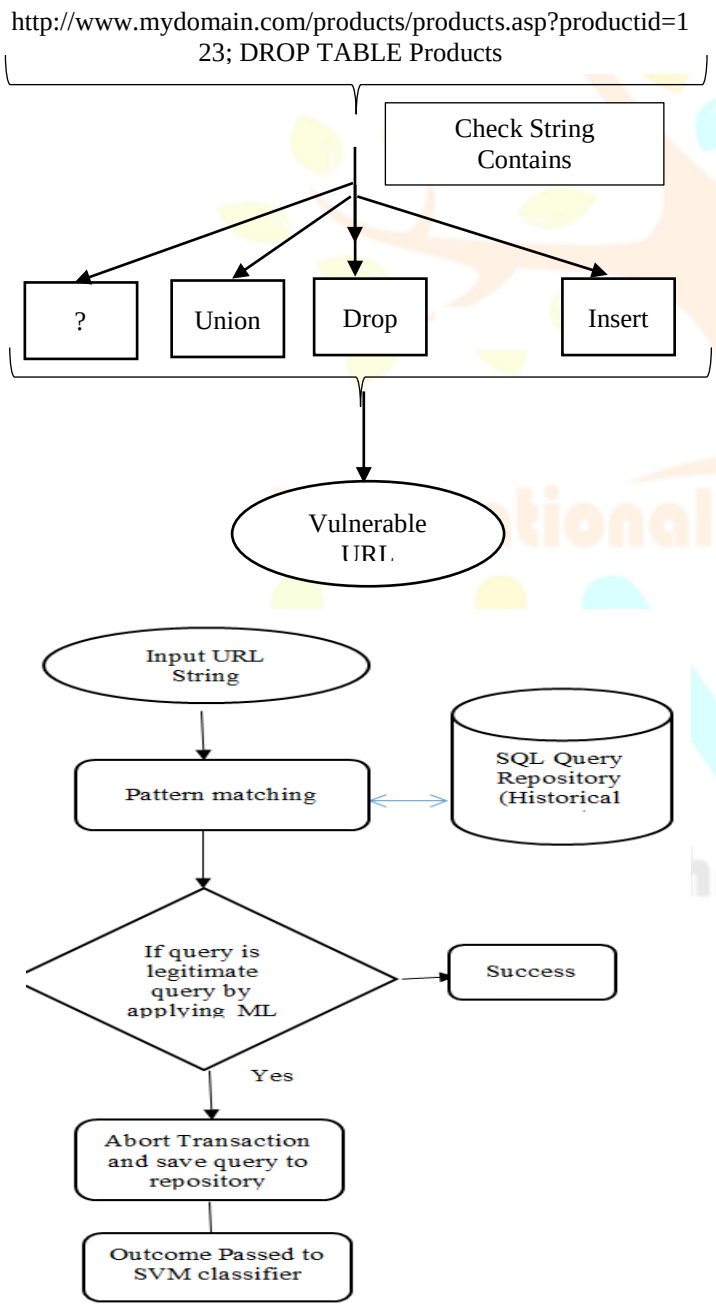


Fig. 2. Proposed System Flow

V. RESULT AND DISCUSSION

For implementation of proposed method we have used jdk1.8.0 64 bit. We have taken SQL injection URL examples of different categories from https://www.owasp.org. For performance evaluation outcome of proposed method passed through SVM (Support vector machine), this part implemented on Matlab 2016a 64 bit. Following are the some snippet of implementation:

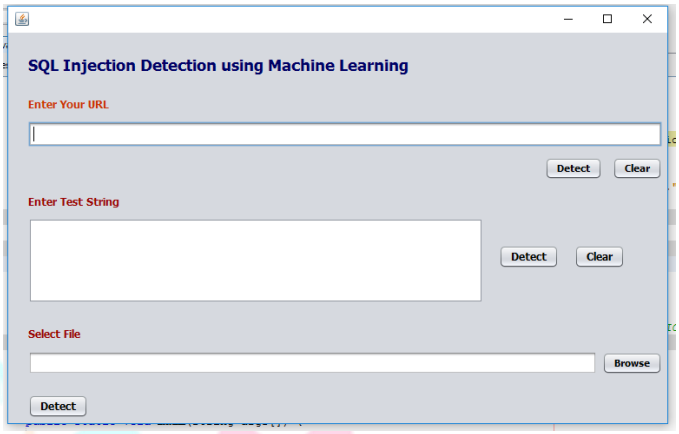
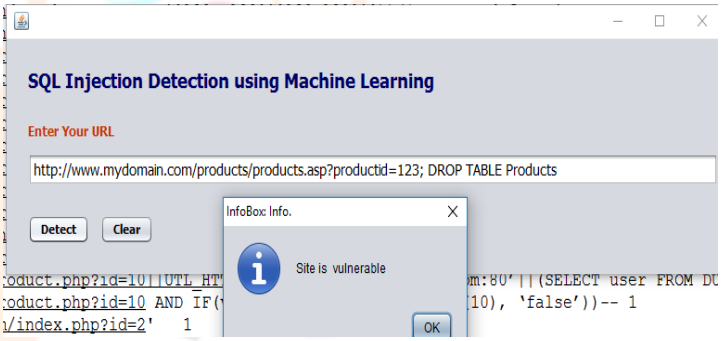


Fig.-3 Main GUI of Proposed System

Fig.-3 shows that we can supply input URL string.



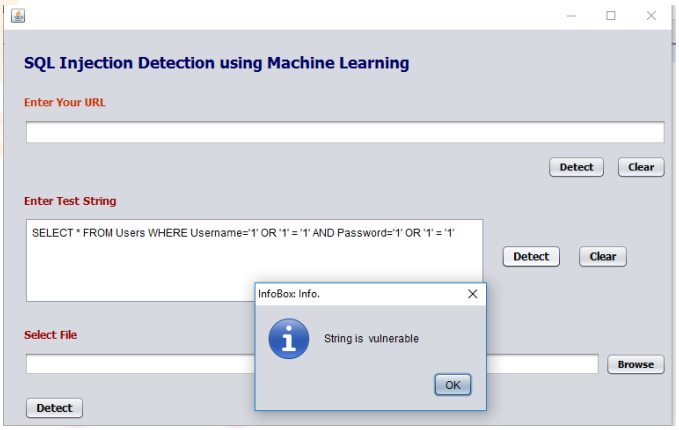


Fig.-4 Final Outcome URL Vulnerable or Not

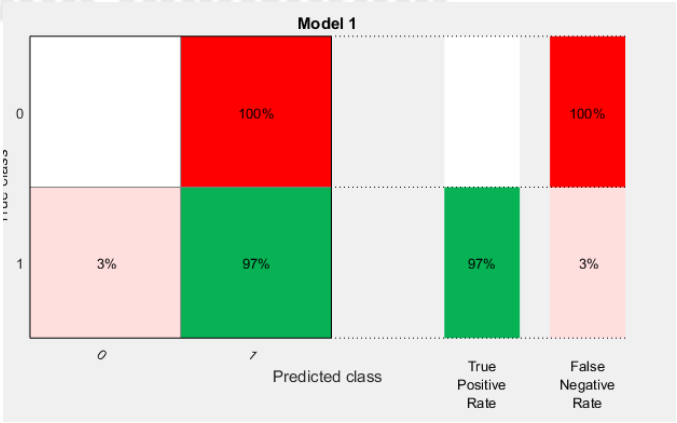


Fig.-5 SVM Classifier Output

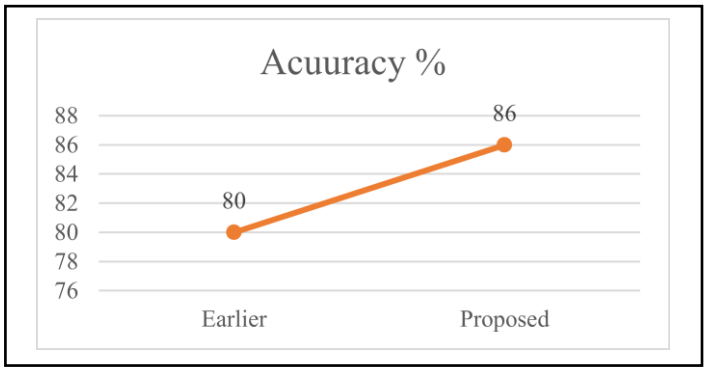


Fig.-6 Performance Comparison

REFERENCES

[1] Inyong Lee et. al./A novel method for SQL injection attack detection based on removing SQL query attribute values/Elsevier 2012.

[2] Indrani Balasundaram et. al./An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching/Elseevier 2012.

[3] Yuji Kosuga et. al./Sania: Syntactic and Semantic Analysis for Automated Testing against SQL Injection/ IEEE 2007

[4] Vipin Das et. al./Network Intrusion Detection System Based On Machine Learning Algorithms/IJCSIT 2010.

[5] Muhammad Saidu Aliero et. al./Classification of Sql Injection Detection And Prevention Measure/ IOSRJEN 2016.

[6] Mahdi Zamani and Mahnush Movahedi/Machine Learning Techniques for Intrusion Detection/arXiv 2015

[7] Rung-Ching Chen and Kai-Fan Cheng/Using Rough Set and Support Vector Machine for Network Intrusion Detection System/IIDS 2009.

[8] V.B. Livshits, M.S. Lam, Finding security errors in Java programs with static analysis, in: Proceedings of the 14th Usenix Security Symposium, 2005, pp. 271–286.

[9] S. Thomas, L. Williams, Using automated fix generation ot secure SQL statements, in: Proceeding of the 29th International Conference on Software Engineering Workshops, ICSEW, IEEE Computer Society, 2007, p. 54.

[10] G. Wassermann, Z. Su, An analysis framework for security in web applications, in: Proceedings of the FSE Workshop on Specification and Verification of Component-Based Systems, SAVCBS, 2004, pp. 70–78.

[11] Y. Kosuga, K. Kernel, M. Hanaoka, M. Hishiyama, Y. Takahama, Sania: syntactic and semantic analysis for automated testing against SQL injection, in: Proceedings of the Computer Security Applications Conference 2007, 2007, pp. 107–117.